



Hungarian standard glossary of terms used in Software Testing

Szoftvertesztelés egységesített kifejezéseinek gyűjteménye

**HTB-Hungarian Testing Board
Magyar Szoftvertesztelői Tanács Egyesület**

Neumann János u.1/E
H-1117 Budapest, Hungary
Tel: +36 1 382 7297
Fax: +36 1 382 7298
<http://www.hstqb.com>

Version:

3.12

Last updated:

21/02/2012

Revision History

Date	Version	Description	Author
21/05/2008	0.3	Document structure	Árpád Beszédes
21/09/2008	0.4	Draft including A-K, T-Z	Gábor Kapros
18/10/2008	0.6		Gábor Kapros
02/11/2008	0.7	Suggestions for translation	Árpád Beszédes
21/03/2009	0.8	Incorporated common voting results	Gábor Kapros
26/03/2009	0.9	Incorporated common voting results	Gábor Kapros
26/03/2009	1.0	Released version	Gábor Kapros
10/10/09	2.0	Released based on review remarks	Gábor Kapros
26/02/2010	2.1	Small correction	Gábor Kapros
15/03/2010	3.0	Released version based on review assessments of foundation members of Hungarian Testing Board. Released parallel with <i>Official ISTQB CTFL Syllabus – Hungarian 2.0</i>	Gábor Kapros
22/06/2010	3.01	Small corrections	Gábor Kapros
27/06/2010	3.09	Adapted to ISTQB Glossary 2.1	Gábor Kapros
04/05/2011	3.1	HTB review	Gábor Kapros
04/05/2011	3.1	HTB review	Gábor Kapros
04/05/2011	3.11	Small corrections	Gábor Kapros
28/02/2012	3.12	rendszer teszt -> rendszerteszt hibaarány -> meghibásodási ráta kritikus tesztelési folyamat -> kritikus tesztfolyamat páros tesztelés -> páros teszt tesztelési alapelv (test policy) -> tesztelési irányelvek tesztelési becslése -> tesztbecslés tesztelési stratégia -> tesztstratégia tesztelési feltétel -> tesztfeltétel tesztelési folyamat -> tesztfolyamat tesztelési cél -> tesztcél tesztelési technika -> teszttechnika tesztelési szint -> tesztszint hibák csoportosulása -> hibafürtök megjelenése teljesítmény teszt -> teljesítményteszt hibajelenlét kimutatása -> hibák látszólagos hiánya a hibátlan rendszer téveszménye -> a hibamentes rendszer téveszméje	Gábor Kapros

Table of Contents / Tartalomjegyzék

<u>1. Foreword / Előszó</u>	1
<u>2. Document structure / Dokumentum szerkezete</u>	1
<u>3. Hungarian definitions / Magyar meghatározások</u>	2
<u>4. English-Hungarian dictionary / Angol-magyar szótár</u>	43

1. Foreword / Előszó

The main goal of this documentation is to have a unified Hungarian terminology regarding basic terms and expressions of software testing. It is based on ISTQB Glossary of Testing Terms version 2.1.

This glossary serves as a base for Official ISTQB CTFL Syllabus – Hungarian.

This documentation has been created under authority of Hungarian Testing Board.

Copyright © 2009 Hungarian Testing Board. All rights reserved. You may not mirror, modify or otherwise alter this file on the website of Hungarian Testing Board without written permission from Hungarian Testing Board.

Contact, suggestions / Kapcsolat, javaslatok

glossary@hstqb.org

Edited by / Szerkeszti

Kapros Gábor

Contributors / Közreműködők

Beszédes Árpád, Dévényi Gábor, Dusha Kiss Gábor, Kovács Attila, Balla Katalin, Csonka Béla György

2. Document structure / Dokumentum szerkezete

The documentation contains two main chapters:

- 2 a chapter containing the Hungarian terms and definitions in Hungarian alphabetical order. Every definition ends with a “→” sign followed by the corresponding English term.
- 3 a chapter that serves as an English-Hungarian dictionary in English alphabetical order

Remark:

If a Hungarian term does not contain any definition but only a reference to another

Hungarian term (started with “Lásd”), it means that both Hungarian terms (expression) are acceptable and their definitions are the same.

3. Hungarian definitions / Magyar meghatározások

absztrakt teszteset: lásd magas szintű teszteset → *abstract test case*

adaptálhatóság: a szoftver azon tulajdonsága, hogy különböző környezetben mennyire képes különböző külső eszközök nélkül helyesen működni. Lásd még: hordozhatóság → *adaptability*

adatbázis-integritás teszt: azon metódusok és folyamatok tesztje, amelyek az adat(bázis)ok elérését és kezelését végzik. Azt vizsgálja, hogy a hozzáférési metódusok, folyamatok és adat-szabályok az elvárásoknak megfelelően működnek-e, illetve hogy az adatbázis-műveletek során az adatok nem károsodnak-e, illetve nem történik-e nem szándékolt módosulás, törlés, vagy létrehozás → *database integrity testing*

adatfolyam: az adatobjektumok állapotsorrendjének, valamint lehetséges változásainak absztrakt megjelenítése, ahol az objektum állapota létrehozás, használat, illetve megszüntetés lehet → *data flow*

adatfolyam elemzés: a változók definícióján és használatán alapuló statikus elemzés → *data flow analysis*

adatfolyam lefedettség: a tesztkészlet által végrehajtott definíció-használati párok aránya → *data flow coverage*

adatfolyam teszt: fehérdoboz teszt módszer, amely során a változók definíció-használati párijainak segítségével tesztelünk → *data flow testing*

adat-integritás teszt: lásd: adatbázis-integritás teszt → *data integrity testing*

adatvezérelt teszt: olyan szkript módszer, amely egy táblázatban tárolja a bemeneteket és az elvárt értékeket oly módon, hogy egy vezérlő szkript minden tesztet végre tud hajtani belőle. Az adatvezérelt tesztet gyakran használják pl. felvevő/lejátszó tesztelő eszközök használatának támogatására. Lásd még: akciószó alapú teszt → *data driven testing*

ad hoc felülvizsgálat: lásd informális felülvizsgálat → *ad hoc review*

ad hoc teszt: olyan informális teszt, ahol nincsenek formális teszt-előkészületek, nincs felismerhető teszttervezési technika, nincsenek elvárt eredmények, a teszt végrehajtása tetszés szerint történik → *ad hoc testing*

agilis tesztelés: az agilis módszertannal (mint pl. az extrém programozás, XP) fejlesztett szoftverek tesztelési gyakorlata. Az agilis tesztelés egyik sarokpontja, hogy a műszaki teszttervezés megelőzi a fejlesztést. Lásd még: tesztvezérelt fejlesztés → *agile testing*

agilis kiáltvány: az agilis szoftverfejlesztés alapvető értékeit tartalmazó nyilatkozat. A fő értékek:

- az egyének és kölcsönhatásaik fontosabbak, mint az folyamatok és az eszközök

- a működő szoftver fontosabb, mint a minden részletre kiterjedő dokumentáció
- az ügyféllel való együttműködés fontosabb, mint a szerződéses tárgyalások
- a változásokra való reagálás fontosabb, mint a terv követése → *agile manifesto*.

agilis szoftverfejlesztés: a szoftverfejlesztési módszerek csoportja, amely az iteratív fejlesztési modellen alapul, mind a követelmények, mind a megoldások az önszerveződő, széleskörű feladatokkal rendelkező csapatok együttműködése által jönnek létre. → *agile software development*

akciószó alapú teszt: egy olyan, szkripteken alapuló tesztmódszer, amelynél a teszt-szkriptekben nemcsak tesztadatok és az elvárt eredmény található, hanem az alkalmazással kapcsolatos vezérlő kulcsszavak is. → *action word driven testing, keyword driven testing*

aktuális eredmény: a teszt során tapasztalt viselkedés → *actual outcome, actual result*

alacsony szintű teszteset: teszteset konkrét (kód szintű) bemenő és elvárt kimenő értékekkel. Ebben a teszt típusban a magas szintű tesztesetekben levő logikai műveleteket konkrét, de a logikai műveleteknek továbbra is megfelelő értékekkel helyettesítjük. Lásd még magas szintű teszteset → *low level test case, concrete test case*

alap tesztkészlet: olyan teszteset készlet, amely a komponens belső struktúrájából, vagy a specifikációból lett leszármaztatva annak érdekében, hogy 100%-osan biztosítsuk a megadott lefedettség kritérium teljesítését → *basis test set*

alapvető teszt: lásd smoke teszt. → *sanity test*

alfa teszt: szimulált, vagy tényleges tesztelés, amelyet potenciális felhasználók, vagy egy független tesztcsoport végez a fejlesztés helyszínén, de a fejlesztői szervezettől függetlenül. Gyakran használják dobozos szoftverek belső átvételi tesztjeihez → *alpha testing*

állapotdiagram: egy komponens vagy rendszer állapotait ábrázoló diagram, ami az állapotváltozást okozó eseményeket vagy körülményeket is. [IEEE 610] → *state diagram*

állapot-tábla: egy táblázat, ami minden egyes állapotra és lehetséges eseményre mutatja az állapotátmeneteket, megjelenítve az érvényes és érvénytelen átmeneteket is. → *state table*

állapotátmenet: átmenet egy komponens vagy rendszer két állapota között. → *state transition*

állapotátmenet teszt: olyan feketedoboz teszttervezési technika, amiben úgy tervezzük meg a teszteseteket, hogy érvényes és érvénytelen állapotátmeneteket generáljanak. Lásd még N-lépéses teszt → *state transition testing*

állapotdiagram: egy komponens vagy rendszer állapotait ábrázoló diagram, ami mutatja az állapotváltozást okozó eseményeket vagy körülményeket is. [IEEE 610] → *state diagram*

állapotrögzítés: a konfiguráció-menedzsment egy eleme, ami az információk rögzítéséből és jelentéséből áll, a konfiguráció hatékony menedzselése érdekében. Ezek az információk tartalmazzák a jóváhagyott konfiguráció azonosítóját, az előterjesztett konfiguráció változásainak állapotait és a jóváhagyott változások megvalósítási állapotait. [IEEE 610] → *status accounting*

állapottábla: egy táblázat, ami minden egyes állapotra és lehetséges eseményre mutatja az állapotátmeneteket, megjelenítve az érvényes és érvénytelen átmeneteket is. → *state table*

általános célú szoftvertermék: lásd dobozos szoftver → *standard software*

alulról-felfelé haladó teszt: az alsóbb szintű komponensek tesztelésével kezdődő tesztechnika. Lásd még: integrációs teszt → *bottom-up testing*

al-útvonal: egy komponensen belüli végrehajtható utasítások sorozata. → *subpath*

átadandó: bármely olyan termék vagy munka, amelyet a készítőjétől eltérő személynek (szervezetnek) át kell adni → *deliverable*

áttekintőtábla: egy adott szervezet, illetve tevékenység hatékonyságának megjelenítését szolgálja. A metrikákat olyan metaforákkal helyettesítik, mint például „számlap”, vagy „mutató”, vagy más kijelző, amelynek segítségével az autóiparra lehet asszociálni, és így a tevékenység, vagy et pesemény hatása egyszerűen megérthető és hozzáköthető a működési célokhoz. Lásd még egyesített áttekintőtábla, értékelőlap → *dashboard*

átvétel: lásd átvételi teszt → *acceptance*

átvételi teszt: a felhasználó, vagy a megrendelő által a végterméken végzett feketedoboz teszt, amely azt hivatott eldönteni, hogy megfelel-e a termék a megfogalmazott (üzleti) elvárásoknak, illetve folyamatoknak → *acceptance testing*

átvételi kritérium: az elvárt eredmény, amely alapján eldöntjük, hogy a teszteset sikeres, vagy elbukott → *acceptance criteria*

átvizsgálás: egy dokumentum szerzője által végzett lépésenkénti bemutató abból a célból, hogy információt gyűjtsön vagy közös álláspontot alakítson ki. [Freedman és Weinberg, IEEE 1028]. Lásd még egyenrangú felülvizsgálat → *walkthrough*.

audit: a szoftvertermék, illetve a folyamat független értékelése annak érdekében, hogy kiderítsük, megfelel-e a szabványoknak, a specifikációknak, az irányelveknek, és/vagy azon dokumentumokban rögzített objektív kritériumoknak, amelyek meghatározzák

- 1 a termék formáját, illetve tartalmát,
- 2 a folyamatot, ahogy a terméket elő kell állítani.
- 3 hogy hogyan mérhető a standard-oknak, illetve az irányelveknek történő megfelelés → *audit*

audit nyomvonal: olyan útvonal, amelyen a folyamat eredeti bemenete (pl. valamilyen adatok) visszakövethető a folyamaton keresztül, a folyamat kimenetét kiindulópontnak

tekintve. Megkönnyíti a hibaelemzést, és lehetővé teszi folyamat ellenőrzés végrehajtását → *audit trail*

automatizált teszter: automatizált tesztnél használt teszteszköz, mint pl. egy szkriptkészlet → *automated testware*

automatizált tesztvégrehajtás: olyan szoftver használata (pl. felvevő/lejátszó eszközök), amelyekkel a tesztvégrehajtást lehet irányítani, a várt eredményeket össze lehet hasonlítani az aktuális eredményekkel, teszt előfeltételeket lehet meghatározni és egyéb tesztirányítási és jelentési funkciókkal is rendelkezik → *test execution automation*

baseline: verziókövetéssel fejlesztett szoftver, vagy dokumentáció, amely egy adott, formálisan felülvizsgált és elfogadott állapot azonosítására szolgál. A további fejlesztés csupán a formális változásirányítás szabályai szerint történhet → *baseline*

bejárhatatlan útvonal: olyan útvonal, amely semmilyen lehetséges bemeneti érték esetén nem hajródik végre → *infeasible path*

belépési feltétel: általános és speciális feltételek halmaza, amely engedélyezi egy adott feladat végrehajtását. Az a cél, hogy ne indítsunk olyan feladatokat, amelyek több elvesztetgetett ráfordítást jelentenének, mint az elbukó belépési feltételek kijavítása [Gilb és Graham] → *entry criteria*

belépési pont: a komponensen belüli első végrehajtható utasítás → *entry point*

bemenet: a komponens által beolvasott - akár a komponensen belül, vagy kívül tárolt - változó → *input*

bemeneti érték: a bemenet egy fajtája. Lásd még: bemenet → *input value*

bemeneti értéktartomány: azon értéktartomány, amelyből érvényes bemeneti adatok választhatók. Lásd még: értéktartomány → *input domain*

benchmark teszt: összehasonlító szabvány teszt, amelynek alapján a szoftvereket, illetve komponenseket egymással összehasonlíthatjuk, illetve mérhetjük → *benchmark test*

bevált gyakorlat: a tesztelés, vagy a fejlesztés során alkalmazott leghatékonyabb innovatív eljárások, módszerek, amelyeket általában a többi szervezeti egység is átvehet → *best practice*

béta teszt: a szoftvernek egy szűkebb felhasználói körben való külső tesztje a végső kiadás előtt annak érdekében, hogy meghatározzuk, a szoftver megfelel-e a felhasználók piaci igényeinek. Gyakran megfelel a dobozos (COTS) szoftverek külső átvételi tesztjének → *beta testing, field testing*

bizonyossági teszt: lásd smoke teszt → *confidence test*

biztonságosság: a szoftvertermék azon képessége, hogy elfogadható szintű kockázatot biztosítson az emberekkel, üzlettel, tulajdonnal vagy a környezettel, a megadott használati környezetben. [ISO 9126] → *safety*

biztonság-kritikus rendszer: olyan rendszer, aminek a meghibásodása, vagy hibás

működése emberek súlyos sérüléséhez, halálához, eszközök komoly sérüléséhez, elvesztéséhez vagy környezeti károsodáshoz vezethet. → *safety critical system*

biztonság: a szoftvertermékek azon tulajdonságai/attribútumai, amik a programokhoz és adatokhoz való jogosulatlan hozzáférést előzik meg. A hozzáférés lehet szándékos vagy akár véletlen is. Lásd még funkcionalitás. [ISO 9126] → *security*

biztonsági teszt: olyan teszt, amivel a szoftvertermék biztonságát határozzuk meg. Lásd még funkcionalitási teszt → *security testing, safety testing*

biztonsági teszteszköz: olyan eszköz, ami a biztonsági karakterisztikák és sebezhetőségek tesztjében nyújt támogatást. → *security testing tool*

biztonsági eszköz: olyan eszköz, ami a működési biztonságot támogatja. → *security tool*

blokkolt teszteset: olyan teszteset, amely nem hajtható végre, mivel a végrehajtás előfeltétele nem teljesíthető → *blocked test case*

bukás: akkor mondjuk, hogy egy teszt elbukik, ha a tényleges eredménye nem egyezik meg az elvárt eredménnyel → *fail*

Cél Kérdés Metrika: 3 szintű modell a szoftver minőségének mérésére: koncepcionális szint (cél), működési szint (kérdés) és mennyiségi szint (metrika). → *Goal Question Metric (GQM)*

Chow lefedettségi metrika: a tesztkészlet által meghívott N+1 átmenetek aránya → *Chow's coverage metrics, N-switch coverage*

ciklomatikus komplexitás: a független útvonalak maximális száma a programban. A ciklomatikus komplexitás definíciója: $L - N + 2P$, ahol

- L = az élek/kapcsolatok száma a gráfban
- N = a csomópontok száma a gráfban
- P = a nem kapcsolódó részek a gráfban (pl. egy meghívott gráf, vagy szubrutin)
- [McCabe szerint]

→ *cyclomatic complexity*

ciklomatikus szám: lásd ciklomatikus komplexitás → *cyclomatic number*

cselekvés (IDEAL): az IDEAL model egyik fázisa, amely során a javításokat kifejlesztik, átültetik a gyakorlatba és az egész szervezetben életbe léptetik. A cselekvési fázis a következő tevékenységeket tartalmazza: a megoldás létrehozása, tesztje/pilotálása, finomítása és megvalósítása. Lásd még IDEAL → *acting (IDEAL)*

csimpánz-teszt: véletlenszerű bemenettel, tetszőleges billentyű leütésével végrehajtott teszt, függetlenül attól, hogy a terméket helyesen használjuk-e → *monkey testing*

csonk: egy szoftver komponens speciális célú vagy részleges megvalósítása. A csonkot arra használjuk, hogy támogassuk a komponens(ek) fejlesztését vagy tesztelését. Helyettesíti a meghívott komponenst. [IEEE 610] → *stub*

definíció-használat pár: a változó definíciójának összekapcsolása a változó használatával. A változó lehet számításokat végző változó (mondjuk szorzás egyik operandusa), vagy a végrehajtási útvonalat befolyásoló változó → *definition-use pair*

Deming ciklus: iteratív, négylépéses, problémamegoldási folyamat (tervezés – végrehajtás, ellenőrzés, analízis). Jellemzően a folyamatjavítás során használják (Deming szerint) → *Deming cycle*

diagnózis (IDEAL): az IDEAL modellen belül az egyik fázis, amelynél meghatározzuk a kívánt állapothoz képest fennálló helyzetet, A diagnózis fázis : az aktuális és a kívánt állapot leírását, továbbá fejlesztési javaslatokat tartalmaz. Lásd még: IDEAL → *diagnose(IDEAL)*

dinamikus elemzés: a rendszer memória teljesítményének, CPU használatának kiértékelési folyamata → *dynamic analysis*

dinamikus elemző eszköz: a szoftverkód állapotáról futási időben információt szolgáltató eszköz. Leggyakrabban érvénytelen mutatók, memóriafoglalások – és törlések, illetve memóriaszivárgások felderítésére használják → *dynamic analysis tool*

dinamikus teszt: olyan teszt, amely magában foglalja egy komponens vagy rendszer szoftverének futtatását → *dynamic testing*

dinamikus összehasonlítás: az elvárt és a tényleges eredmények összehasonlítása a szoftver végrehajtása közben – pl. egy tesztvégrehajtó eszköz alkalmazásával → *dynamic comparison*

dobozos szoftver: egy, az általános piaci igényeknek megfelelő szoftvertermék, pl.: nagy számú vevő számára, amely minden vásárlónak hasonló, megjelenéssel érhető el. → *off-the-shelf software, COTS*

dokumentáció teszt: a dokumentáció (pl. a felhasználói, vagy installációs kézikönyv) minőségének tesztje → *documentation testing*

döntés: olyan progampont, ahol a vezérlési folyamtnak két, vagy több alternatív útvonala van. Egy csomópont, amelyből két, vagy több elágazás indul → *decision*

döntési eredmény: a döntés eredménye, amely meghatározza, hogy melyik elágazást kell bejárni → *decision outcome*

döntési feltétel lefedettség: a tesztkészlet végrehajtása során az összes feltétel eredmény és döntési eredmény meghívásának százalékos aránya. 100% döntésifeltétel-lefedettség 100%-os feltétel lefedettséget és 100%-os döntési lefedettséget jelent → *decision condition coverage*

döntési feltétel teszt: fehérdoboz teszttervezési módszer, amely során a tesztesetek célja a feltétel eredmények és a döntési eredmények végrehajtása → *decision condition testing*

döntési lefedettség: a tesztkészlet végrehajtása során a döntési eredmények meghívásának százalékos végrehajtási aránya. 100%-os döntési lefedettség 100%-os elágazási lefedettséget és 100%-os utasítás lefedettséget jelent → *decision coverage*

döntési tábla teszt: feketedoboz tesztervezési módszer, amely során olyan teszteseteket tervezünk, amelyek a döntési táblában szereplő különböző okok és bemenetek kombinációit igyekeznek tesztelni. Lásd még döntési tábla → *decision table testing*

döntési teszt: fehérdoboz tesztervezési módszer, amelynek célja, hogy a tesztesetek döntési eredményeket hajtsanak végre → *decision testing*

döntésközi útvonal: olyan végrehajtási útvonal, amely nem tartalmaz feltételes csomópontot. Jellemzően két döntés közti útvonal. A végrehajtási útvonalat megjeleníthetjük például egy folyamatábrával. → *dd-path*

EFQM kiválósági model: a European Foundation for Quality Management tulajdonában levő és általa definiált keretrendszer egy adott szervezet minőségügyi rendszeréhez. Nem előírásokat tartalmaz, hanem 5 "Elősegítés" (Enabling - amit a szervezet csinál) és 4 "Eredmény" (Result, amit a szervezet elér) kritériumon alapul. → *EFQM (European Foundation for Quality Management) excellence model*

egyedi szoftver: Nem tömeggyártási céllal, hanem a megrendelő egyedi igényeinknek megfelelően készített, illetve testreszabott szoftver. A dobozos szoftver (OTS-szoftver) ellentéte. → *bespoke software, custom software*

egyenrangú felülvizsgálat: egy szoftver-munkatermék felülvizsgálata a termék előállítójának munkatársai által, amelynek célja, hogy hibákat fedezzen fel illetve javító javaslatokat hozzon. Példák: inspekció, technikai felülvizsgálat, átvizsgálás → *peer review*

egyesített áttekintőtábla: a közösségi teljesítmény áttekintését szolgálja → *corporate dashboard*

egység: lásd komponens → *unit*

egységteszt: lásd komponens teszt → *unit testing*

egységteszt keretrendszer: olyan környezetet biztosító keretrendszer, amelyben egy komponens egyaránt tesztelhető különállóan valamint a megfelelő segédprogramokkal. Hibakeresési funkciójával támogatja a szoftverfejlesztő munkáját is. [Graham] → *unit test framework*

együttélés: a szoftver azon képessége hogy egy közös környezetben képes együtt élni egy másik, független programmal a közös erőforrások megosztása tekintetében → *co-existence*

együtműködő képesség: a szoftver azon jellemzője, hogy egy, vagy több adott komponenssel, illetve rendszerrel milyen az egymásra hatásuk [ISO 9126 szerint]. Lásd még: funkcionalitás → *interoperability*

együtműködő képességi teszt: egy adott szoftver együtműködő-képességének vizsgálata. Lásd még funkcionalitási teszt → *interoperability testing*

ekvivalenciaosztály: lásd ekvivalencia partíció → *equivalence class*

ekvivalencia-partíció: olyan bemeneti, vagy kimeneti értéktartomány, amelyre a specifikáció alapján a rendszernek ugyanúgy kell viselkednie → *equivalence partition*

ekvivalencia-partíció lefedettség: a tesztkészlet által végrehajtott ekvivalencia partíciók aránya → *equivalence partition coverage*

ekvivalencia partícionálás: feketedoboz teszt módszer, amely során olyan teszteseteket készítünk, amelyek az ekvivalencia partíciók egyes reprezentánsait tesztelik. Jellemzően minden egyes ekvivalencia partíciót érdemes legalább egyszer lefedni → *equivalence partitioning*

elágazás: a program egy logikai feltételtől (pl. ugrás, goto, if then else) függő útvonala → *branch*

elágazási feltétel: lásd: feltétel → *branch condition, condition*

elágazás lefedettség: a tesztkészlet által meghívott elágazások százalékos aránya. 100% elágazás lefedettség 100% döntési lefedettséget és 100% utasítás lefedettséget jelent. → *branch coverage*

elágazási feltétel lefedettség: a teszt végrehajtása által meghívott feltétel eredmények aránya. A 100%-os elágazási feltétel lefedettség azt jelenti, hogy minden döntési utasításban minden egyes feltétel igaz, illetve Hamis ága tesztelve van → *branch condition coverage, condition coverage*

elágazási teszt: olyan fehérdoboz teszttervezési technika, amely során elágazásokat tesztelünk [TMap] → *branch testing, arc testing, algorithm test [TMap]*

elemezhetőség: a szoftver azon tulajdonsága, hogy mennyire lehetséges a benne levő meghibásodások, illetve a módosítandó részek megtalálása → *analyzability*

elemi blokk: a szoftver elágazást nem tartalmazó utasításainak sorozata. Egy belépési, egy kilépési ponttal rendelkezik, és nincs benne ugrási utasítás → *basic block, segment*

elemi összehasonlító teszt: feketedoboz teszt módszer, amely során olyan teszteseteket készítünk, amelyek a feltétel meghatározási lefedettség koncepcióján alapuló bemeneti kombinációkat tartalmaznak → *elementary comparison testing*

elemző: lásd statikus elemző → *analyzer*

elérhetetlen kód: olyan kódrészlet, amely nem érhető el, így nem lehet végrehajtani → *unreachable code*

eljárás teszt: egy komponens vagy rendszer tesztje, amely arra irányul, hogy a komponens vagy rendszer együtt tud-e működni az új vagy meglévő felhasználók üzleti eljárásaival, működési folyamataival. → *procedure testing*

ellenőrzés papíron: a szoftver, vagy a specifikáció tesztje a végrehajtás kézi szimulálása által. Lásd még: statikus elemzés → *desk checking*

ellenőrző lista alapú teszt: tapasztalat-alapú teszttervezési technika, amely során a tapasztalt tesztelő magas szintű tesztelemeket jegyez fel, ellenőriz, vagy szabályokat és kritériumokat használ a program verifikáláshoz. Lásd még tapasztalat-alapú teszt → *checklist-based testing*

ellenőrző teszt: lásd újratestelés → *confirmation testing*

előfeltétel: környezeti vagy állapotbeli feltételek, amelyeket teljesíteni kell, mielőtt egy komponensen vagy rendszeren tesztet vagy tesztfolyamatokat kezdenénk. → *precondition*

előre jelzett kimenet: lásd elvárt eredmény. → *predicted outcome*

előzetes teszt: lásd befogadási teszt. → *pretest*

elszigetelt teszt: az egyes komponensek tesztje az őket körülvevő komponensektől elszigetelve, illetve azokat csonkokkal, valamint meghajtókkal helyettesítve → *isolation testing*

eltérés: lásd incidens → *deviation*

eltérési jelentés: lásd incidensjelentés → *deviation report*

elvárt eredmény: a szoftver adott feltételek mellett, a specifikáció, vagy más forrás alapján előrejelzett viselkedése → *expected outcome, expected result*

emberi eredetű hiba: emberi tevékenység, amely során helytelen eredmény jön létre → *error, mistake*

emulátor: olyan eszköz, vagy program, amely ugyanazon bemeneteket fogadja, és ugyanazon kimeneteket generálja, mint az adott rendszer → *emulator*

eredmény: a tesztvégrehajtás következménye/kimenete, amely lehet: képernyő, adatváltozás, riport, kiküldött kommunikációs üzenet, stb. Lásd még aktuális eredmény, elvárt eredmény. → *outcome, result*

életciklus modell: a projekt, vagy a termék életének fázisokra bontása [CMMI]. Lásd még: szoftver életciklus → *lifecycle model*

értékkadás: végrehajtható utasítás, amelyben egy változóhoz hozzá van rendelve egy érték → *data definition*

értékelő jelentés: az értékelés eredményét összefoglaló dokumentum, amely tartalmazhatja a következtetéseket, ajánlásokat és észrevételeket. Lásd még: folyamatértékelés. → *assessment report*

értékelő: az értékelést végrehajtó személy: az értékelő csapat bármely tagja. → *assessor*

érthetőség: a szoftver termék azon tulajdonsága, hogy a felhasználó mennyire könnyen képes eldönteni, hogy a szoftver számára megfelelő-e, továbbá hogy azt hogyan és milyen feltételek mellett használhatják bizonyos feladatokra. [ISO 9126] Lásd felhasználhatóság → *understandability*

érvénytelen mutató: olyan mutató, amely a saját hatókörén kívüli vagy nem létező pontra hivatkozik. Lásd még mutató (pointer) → *wild pointer*

érettség:

(1) egy szervezet azon képessége, hogy milyen eredményességgel és hatékonysággal használja elméleti- és gyakorlati munkafolyamatait. Lásd még képességi-érettségi model (CMM), teszt-érettségi modell (TMM).

(2) egy szoftvertermék azon képessége amelynek köszönhetően elkerülhető, hogy egy esetlegesen fellépő program hiba működési hibát okozzon. [ISO 9126] Lásd még megbízhatóság → *maturity*

erőforrás kihasználtság: annak a mérőszáma, hogy a szoftvertermék adott feltételek mellett megfelelő (indokolt) mennyiségű és típusú erőforrást használ. Például a fő és a másodlagos memóriahasználat vagy a tárolási területfoglalás megfelelő szinten marad, amikor a program utasításokat hajt végre. [ISO 9126] Lásd még hatékonyság. → *resource utilization*

erőforráshasználati teszt: a szoftvertermék erőforrás- használatának meghatározását – például memóriafelhasználást – célzó teszt. Lásd még hatékonysági teszt → *resource utilization testing*

érettségi modell: a szervezeti érettség bizonyos szempontjait leíró elemek strukturált gyűjteménye, amely a szervezeti folyamatok meghatározását és megértését segíti. Az érettségi modell gyakran közös nyelvet, jövőképet és a fejlesztő tevékenységek prioritizálását elősegítő keretrendszert nyújt. → *maturity model*

érettségi szint: a folyamatfejlesztés során, az előre meghatározott folyamatterületeken elért szint, amely szint összes célját megvalósítottuk → *maturity level*

értékelőlap: az összesített teljesítmény értékelés megjelenítésére szolgál, a hosszútávú célok felé történő előrehaladást mutatja. Az értékelőlap egy adott időszak egészén, vagy végén meglévő teljesítmény statikus mérésére alkalmazható. Lásd még: viszonyított értékelőlap, áttekintőtábla → *scorecard*

érték-alapú minőség: a minőség egy formája, ahol a minőséget az ár határozza meg. A termék, vagy szolgáltatás minőségét az határozza meg, hogy a kívánt teljesítményt elfogadható költségekért tudja-e nyújtani. A minőséget egy döntési folyamat határozza meg, amelyet az érintettekkel együtt végzünk és amelyben a határidő, a ráfordítás és a költségek szempontjait figyelembe véve között optimalizálunk. → *value-based quality*

értéktartomány: azon készlet, amelyből az érvényes bemenetek és kimenetek választhatók → *domain*

érvénytelen teszt: azon bementi értékekkel való teszt, amelyet a szoftvernek nem szabad elfogadnia. Lásd még hibátűrés → *invalid testing*

érzelmi intelligencia: egy személy képessége, tehetsége, hogy saját maga, más személy, vagy egy csoport érzelmeit felismerje, értékelje és kezelje. → *emotional intelligence*

extrém programozás: szoftverfejlesztési módszertan, amelyet az agilis szoftverfejlesztés során használnak. A legfontosabb jellemzői a páros programozás, az alapos kód felülvizsgálat, minden kódhoz unit teszt készítése, illetve a kód átláthatósága. Lásd még: agilis szoftverfejlesztés. → *extreme programming*

fagyasztott tesztbázis: olyan tesztbázis dokumentum, amelyet csupán formális változás-kezelési folyamaton keresztül lehet megváltoztatni. Lásd még: baseline → *frozen test basis*

feature: a szoftvernek a követelmény-dokumentációban meghatározott attribútuma (pl. a megbízhatóságra, a használhatóság, illetve tervezési jellemzőire)

fehérdoboz teszt: a szoftver belső struktúrájának elemzésén alapuló teszt → *white-box testing, glass box testing, clear-box testing*

fehérdoboz teszttervezési technika: olyan eljárás, amely a teszteseteit egy komponens vagy rendszer belső struktúrájának elemzéséből származtatja → *white-box test design technique*

fejlesztői teszt: a szoftver fejlesztése közben, általában a fejlesztők által végrehajtott formális, vagy informális teszt, jellemzően fejlesztői környezetben → *development testing*

feketedoboz teszt: a program belső szerkezetére történő hivatkozás nélküli funkcionális, vagy nem-funkcionális teszt → *black-box testing*

feketedoboz teszttervezési technika: olyan módszer, amelynél a szoftver specifikáció alapján, a program belső szerkezetének ismerete nélkül tervezünk teszteseteket → *black-box test design technique, black-box technique*

felderítő teszt: informális teszttervezési módszer, amely során a tesztelő aktívan felügyeli a teszteset tervezését, a futtatás során szerzett információkat összegyűjti és hasznosítja új és jobb teszteset tervezése érdekében [Bach szerint] → *exploratory testing*

felfüggesztési feltételek: azon feltételek, amik alapján a tesztesetemen (ideiglenesen) leállítjuk az összes teszt tevékenységet. [IEEE 829] → *suspension criteria*

felhasználó-alapú minőség: a minőség egy formája, amelyet azon kapacitás függvényében határozzunk meg, amely a felhasználó(k) igényeit hivatott kielégíteni. Azon termék, vagy szolgáltatás, amely nem felhasználói igényeket old meg, aligha talál magának felhasználót. Ez egy nehezen meghatározható, a környezettől nagyban függő megközelítési mód, mivel a különböző karakterisztikájú üzleti elvárások különböző termékminőséget tesznek szükségessé. → *user-based quality*

felhasználói forgatókönyv teszt: lásd használati eset teszt → *user scenario testing*

felhasználói átvételi teszt: lásd átvételi teszt → *user acceptance testing*

felhasználói teszt: olyan teszt, amelyben valós felhasználók vesznek részt a rendszer vagy rendszerkomponens használhatóságának kiértékelésében → *user test*

feltétel: olyan logikai kifejezés, amely Igaz, vagy Hamis értékeket vehet fel (pl. $A > B$). Lásd még: tesztfeltétel → *condition, branch condition*

feltétel eredménye: valamely feltétel Igaz, vagy Hamis értékre történő kiértékelése → *condition outcome*

feltétel meghatározási lefedettség: annak mérőszáma, hogy a teszt végrehajtása során milyen arányban kerülnek meghívásra a döntési eredményeket függetlenül befolyásoló egyes feltétel eredmények. 100% feltétel meghatározási lefedettség 100% döntési feltétel lefedettséget jelent → *condition determination coverage*

feltétel meghatározási teszt: olyan fehérdoboz teszt módszer, amely során azon elemi feltételekhez tartoznak tesztek, amelyek függetlenül befolyásolják a döntési eredményt → *condition determination testing*

feltétel teszt: olyan fehérdoboz teszt módszer, amely során a tesztesetek feltételekhez tartoznak → *condition testing*

felügyeleti eszköz: egy szoftver, vagy hardver eszköz, amely párhuzamosan fut a tesztelés alatt levő komponenssel vagy rendszerrel, ez utóbbiakat felügyeli, elemzi és/vagy elemzi azok viselkedését. [After IEEE 610] → *monitor, monitoring tool*

felülvizsgálat: egy termék vagy projekt státuszának értékelése. Célja, hogy feltérképezze az eltéréseket a tervezett eredményekhez képest, valamint ajánlást tegyen a továbblépéshez. Több típusa van, például: menedzsment felülvizsgálat, informális felülvizsgálat, technikai felülvizsgálat, inspekció, átvizsgálás. [After IEEE 1028] → *review*

felülvizsgáló: a felülvizsgálaton résztvevő személy, aki azonosítja és leírja a termékben vagy projektben azonosított eltéréseket. A felülvizsgálókat célszerű úgy kiválasztani, hogy többféle nézőpontot, szerepkört képviseljenek. → *reviewer*

felülvizsgáló eszköz: olyan eszköz, ami a felülvizsgálat folyamatát támogatja. Jellemzően a felülvizsgálatok tervezését, a változáskövetést, a felülvizsgálók közötti kommunikációt, közös felülvizsgálat végzését támogatja. Ezek mellett a mérőszámok egyfajta gyűjtőhelyeként, valamint az azokat tartalmazó jelentések alapjául is szolgál. → *review tool*

felülről-lefelé haladó teszt: az integrációs teszt lépcsőzetes megközelítése, amelyben először a hierarchia felső szintjén található komponens kerül tesztelésre az alsóbb rétegek csonkokkal történő szimulációja által, majd az alsóbb szinteken található komponensek tesztje következik. A folyamat addig ismétlődik, amíg a legalsó szintű komponensek is tesztelésre kerülnek. Lásd még integrációs teszt → *top-down testing*

felülvizsgáló: a felülvizsgálatba bevont személy, aki meghatározza és leírja a felülvizsgálat tárgyát képező termék, vagy projekt rendellenességeit. A felülvizsgálókat úgy szokták választani, hogy a felülvizsgálati folyamat különböző aspektusait és szerepköreit képviseljék. → *checker, reviewer*

felvevő/lejátszó eszköz: olyan végrehajtó eszköz, amely felveszi a kézi teszt lépéseit annak érdekében, hogy ezekből később végrehajtható automatikus teszt szkripteket generáljon (pl. megismétli őket). Ezeket az eszközöket gyakran használják az automatikus regressziós tesztek támogatására → *capture/playback tool, record/playback tool*

folyamat: egymáshoz kapcsolódó tevékenységek halmaza, amely a bemenő értékekből kimenő értékeket generál. [ISO 12207] → *process*

folyamat ciklus teszt: olyan feketedoboz teszttervezési technika, amelyben a teszteseteket üzleti folyamatok és eljárások futtatására tervezték. [TMap] Lásd még eljárás teszt. → *process cycle test*

folyamatértékelés: egy szervezet szoftverfolyamatainak egy referencia modell alapján

történő, szabályozott kiértékelése [ISO 15504]. → *process assessment*

folyamatjavítás: szervezeti folyamatok teljesítményének és érettségének javítására szolgáló tevékenységek és ezek eredményinek összessége. [CMMI] → *process improvement*

folyamatlefutási modell: a strukturált elemző és tervező módszerek részeként használatos. Az információ rendszerben történő áramlását mutatja. Minden egyes folyamat bemenetet (bemeneteket) kimenetté (kimenetekké) transzformálja. Lásd még adatfolyam elemzés → *process flow model*

folyamatmodell: olyan keretrendszer, amelyben az ugyanolyan jellegű folyamatokat egy általános modellhez sorolják, pl. a tesztfelkészítési modellhez. → *process model*

folytonos reprezentáció: Képességi-Érettségi modell struktúra, amelyben a képességi szintek meghatározzák a folyamat-területeken belül a folyamatjavítás formáját → *continuous representation*

fordítóprogram: egy olyan szoftver eszköz, amely a magas szintű programnyelvi kifejezéseket a gépi kódú megfelelőjére fordítja → *compiler*

forgatókönyv alapú teszt: lásd használati eset teszt. → *scenario testing*

formális felülvizsgálat: dokumentált eljárásokkal és követelményekkel (pl. inspekcióval) jellemzett felülvizsgálat → *formal review*

forráskód utasítás: lásd utasítás. → *source statement*

fő tesztterv (master test plan): olyan tesztterv, amely általában többszintű tervezést foglal magába. Lásd még tesztterv (test plan) → *master test plan*

funkcionalitás: a szoftvertermék azon képessége, hogy – bizonyos feltételek fennállása esetén - képes a meghatározott igények szerinti működésre → *functionality*

funkcionalitási teszt: a rendszer funkcionalitását vizsgáló teszt → *functionality testing*

funkcionális integráció: olyan integrációs eljárás, amely összekapcsolja az egyes komponenseket, illetve rendszereket, hogy az alapvető funkcionalitások minél hamarabb működhessenek. Lásd még: integrációs teszt → *functional integration*

funkcionális követelmény: olyan követelmény, amely a szoftverrel szemben támasztott funkcionális elvárást írja le → *functional requirement*

funkcionális teszttervezési technika: a rendszer funkcionális specifikációjának elemzésén alapuló teszttervezési módszer, amely nem tartalmaz semmilyen utalást a rendszer belső struktúrájára → *functional test design technique*

funkcionális teszt: a rendszer funkcionális specifikációján alapuló teszt. Lásd még: feketedoboz teszt → *functional testing*

funkciópont elemzés: egy informatikai rendszer funkcionalitásának méretét meghatározó módszer. A mérés független a technológiától. A produktivitás, a szükséges erőforrások, és a projektirányítás mérésének alapjául szolgálhat → *Function Point Analysis (FPA)*

független tesztelés: a felelősségi körök elválasztása, amely az objektív tesztelést segíti
→ *independence of testing*

függőleges nyomonkövethetőség: a követelmények nyomonkövetése a több rétegű fejlesztési dokumentációkon keresztül a különböző komponensekhez → *vertical traceability*

gondolattérkép: egy központi ötlet, vagy szó körül más szavak, ötletek, feladatok, vagy egyéb egységek elrendezését, illetve szerkesztését szolgáló ábra. Gondolattérképeket ötletek létrehozására, megjelenítésére, rendezésére és osztályozására használnak, illetve segíthetnek tanulmányok, szervezetek, problémamegoldások és döntések meghozatalában és leírásában. → *mind-map*

gyártás-alapú minőség: a minőség egy formája, amely során a minőséget úgy határozzuk meg, hogy a termék, vagy a szolgáltatás mennyire felel meg a terveknek, illetve a követelményeknek. A minőség az alkalmazott folyamat(ok)on alapul. Lásd még: termék-alapú minőség, transzcendencia-alapú minőség, felhasználó-alapú minőség, érték-alapú minőség → *manufacturing-based quality*

halott kód: lásd elérhetetlen kód → *dead code*

halszálka diagram: lásd ok-okozat diagram. → *fishbone diagram*

használati eset: mérhető eredménnyel rendelkező tranzakciók sorozata egy dialóguson a felhasználó és a rendszer között. → *use case*

használati eset teszt: olyan feketedoboz teszttervezési technika, amelyekben a műszaki tesztterveket (test design) különböző használati eset forgatókönyvek futtatására készítették → *use case testing*

használhatóság: a szoftver képességeinek megértése, megtanulása, használata a felhasználó számára, meghatározott feltételek mentén. [ISO 9126] → *usability*

használhatósági teszt: annak meghatározása, hogy a szoftvertermék mennyire érthető meg, mennyire könnyű megtanulni használatát és milyen könnyen kezelhető, ezáltal - meghatározott feltételek mentén - mennyire felhasználóbarát. [ISO 9126 alapján] → *usability testing*

határérték: egy olyan bemeneti, vagy kimeneti érték, amely egy ekvivalencia partíció szélén, vagy attól a legkisebb távolságra helyezkedik, például egy tartomány minimum, vagy maximum értékei → *boundary value*

határérték elemzés: a program változóinak, illetve paramétereinek szélsőérték-elemzésén alapuló feketedoboz teszttervezési technika. Lásd még: határérték → *boundary value analysis*

határérték lefedettség: a program változóinak, illetve paramétereinek szélsőértékeinek leteszteltsége az adott tesztkészlet által → *boundary value coverage*

határérték teszt: lásd: határérték elemzés → *boundary value testing*

hatáselemzés: a szoftverváltozás kihatásának elemzése a fejlesztési dokumentáció, a teszt dokumentáció, illetve a komponensek tekintetében, a követelmények változásainak megvalósítása érdekében → *impact analysis*

hatékonyság: a szoftvertermék azon képessége, hogy megfelelő teljesítménnyel működik, figyelembe véve az adott feltételek mellett felhasználható erőforrásokat → *efficiency*

hatékonysági teszt: a szoftver hatékonyságának megállapítását célzó teszt → *efficiency testing*

helyettesíthetőség: egy szoftvertermék azon képessége, hogy egy másik szoftvertermék helyett ugyanarra a célra ugyanazon a környezeten lehessen használni. [ISO 9126] Lásd még hordozhatóság. → *replaceability*

Helyreállási Idő: A rendszer bármely meghibásodása utáni helyreállási átlagidő (számítási közép). Jellemzően a hiba kijavításának tesztelését is. → *Mean Time To Repair (MTTR)*

helyszíni átvételi teszt: a felhasználói/ügyfél oldalon elvégzett teszt annak a meghatározására, hogy egy komponens vagy rendszer - általában beleértve a hardvert és a szoftvert is - kielégíti-e a felhasználó/ügyfél igényeit az üzleti folyamaton belül. → *site acceptance testing*

heurisztikán alapuló kiértékelés: statikus használhatósági teszt, amelynek segítségével meghatározhatjuk a felhasználói felület megfelelését az elfogadott használhatósági jellemzők (az ún. "heurisztikák") szerint → *heuristic evaluation*

hiba alapú technika: lásd hiba alapú teszttervezési technika → *defect based technique*

hiba alapú teszttervezési technika: olyan eljárás, amelynek során teszteseteket származtatunk, illetve választunk ki egy, vagy több hiba kategóriából annak alapján, amit tudunk az adott hibakategóriáról → *defect based test design technique*

hiba-elhelyezés: olyan folyamat, amelynek során ismert hibákat helyezünk el a programban, és azt vizsgáljuk, hogy a teszt során ezek közül mennyit találunk meg. Ebből következtethetünk a meg nem talált hibák számára → *bebugging, fault seeding, error seeding*

hibaelhelyezés: a programban szándékosan elhelyezett hibáknak a teszt során megtalált arányáról következtetni lehet a nem ismert hibák megtalálási arányára → *bebugging*

hibaelhelyező eszköz: olyan eszköz, amely – pl. szándékos beszúrással – hibákat helyez el a szoftverben → *error seeding tool, fault seeding tool*

hiba észlelési arány: egy tesztfázisban talált hibák száma osztva az adott teszt fázisban és később talált kódhibák számával → *Defect Detection Percentage (DDP), Fault Detection Percentage (FDP)*

hibafa elemzés: a hibák elemzésére szolgáló módszer. Vizuálisan modellezi, hogy a meghibásodások, az emberi tévedések és külső hatások hogyan kombinálódhatnak, hogy speciális hibákat hozzanak elő → *Fault Tree Analysis (FTA)*

hibajelentés: olyan dokumentum, amely leírja a szoftver azon hibáit, amelyek a program elégtelen működéséhez vezethetnek → *bug report, defect report*

hibakeresés: a szoftver meghibásodás okainak megtalálási, analízis és eltávolítási

folymata → *debugging*

hibakereső: lásd hibakereső eszköz → *debugger*

hibakereső eszköz: a programozó által használt eszköz a külső hibák reprodukálására, elemzésére és a hibák okainak megtalálására. A hibakereső eszközök lehetővé teszik a program lépésenkénti végrehajtását, illetve bármely utasításánál történő megállítást, valamint a programváltozók vizsgálatát és beállítását → *debugging tool*

hibakövető eszköz: a hibák és változások feljegyzésének és nyomon követésének megkönnyítését célzó eszköz. Általában végigkíséri a hibákat a bejegyzéstől kezdve a javításon át az újratestelésig, illetve különböző jelentési lehetőségeket kínál. Lásd még incidens menedzsment eszköz → *bug tracking tool, defect tracking tool*

hibamenedzsment: azon folyamat, amely során felismerjük, analizáljuk a kódhibákat, osztályozzuk őket, meghatározzuk hatásaikat → *defect management*

hibamaszkolás: olyan állapot, amikor az egyik hiba megakadályozza a másik hiba megtalálását → *defect masking, fault masking*

hibamenedzsment eszköz: olyan eszköz, amely megkönnyíti a hibák és változások feljegyzését és állapotaik nyomon követését. Lehetőséget nyújt a hibák allokálására, kijavítására és újratestelésére, valamint támogatja a jelentéskészítést. Lásd még: incidens menedzsment eszköz → *defect management tool*

hibaosztályozás: (Hierarchikus) kategóriák olyan rendszere, amely támogatást nyújthat a hibák osztályozásában → *bug taxonomy, defect taxonomy*

hibasejtés: olyan műszaki tesztervezési módszer, amely során a tesztelők tapasztalata alapján próbáljuk megsejteni a tesztelendő szoftverben levő hibákat, illetve ez alapján próbálunk megfelelő tesztek tervezni → *error guessing*

hibasűrűség: a szoftverben azonosított hibák aránya a szoftver méretéhez viszonyítva (ez utóbbi szabványos mértékegységben kifejezve, mint pl. a kódsorok, vagy az osztályok, illetve funkciópontok száma) → *defect density, fault density*

hibatámadás: célzott próbálkozás a teszt tárgyának minőségének, különösképpen a megbízhatóságának meghatározására azáltal, hogy speciális meghibásodásokat próbálunk meg szándékosan előidézni. Lásd még: negatív teszt → *attack, fault attack*

hibatűrés: a szoftver azon tulajdonsága, hogy mennyire képes folytatni a normális működését meghatározott teljesítményen a hibás bemenetek, emberi hibák, valamint a hibák és belső rendellenességek ellenére → *error tolerance, fault tolerance*

hiperlink: egy web-oldalon található, egy másik web-oldalra vezető mutató → *hyperlink*

hiperlink ellenőrző eszköz: olyan eszköz, amely azt ellenőrzi, hogy nincsenek-e hibás hiperlinkek a web-oldalon → *hyperlink tool*

hiperlink tesztszeköz: olyan eszköz, amely azt ellenőrzi, hogy a web-oldalon nincs-e érvénytelen hiperlink. → *hyperlink test tool*

hívási gráf: a programban levő szubrutinok közötti hívási kapcsolatok absztrakt megjelenítése → *call graph*

hordozhatóság: egy szoftvertermék átmozgathatóságának mértéke (hardverről egy másikra, vagy egyik szoftver környezetből egy másikba). [ISO 9126] → *portability*

hordozhatósági teszt: egy szoftvertermék hordozhatóságra vonatkozó tesztfolyamat. → *portability testing*

hozzáférhetőségi teszt: annak ellenőrzése, hogy a valamilyen fogyatékkal élő emberek számára hozzáférhető-e, illetve mennyire könnyen használható a komponens, illetve a rendszer → *accessibility testing*

IDEAL: egy szervezeti fejlesztési modell, amely a fejlesztési tevékenységek indításának, tervezésének és bevezetésének terveként szolgál. Az IDEAL modellt 5 fázisa írja le: indítás, diagnózis, kiépítés, cselekvés és tanulás. → *IDEAL*

időbeli viselkedés: lásd teljesítmény → *time behavior*

incidens: bármely olyan történés, amely vizsgálandót tesz szükségessé [IEEE1008] → *incident*

incidensjelentés: olyan dokumentum, amely minden, pl. a teszt alatt bekövetkezett incidenst tartalmaz, amely vizsgálatot tesz szükségessé [Az IEEE 829 szerint] → *incident report*

incidens naplózása: bármely bekövetkezett incidens feljegyzése, pl. a teszt alatt → *incident logging*

incidens menedzsment: az incidensek felismerésének, vizsgálatának, a különböző intézkedések és rendelkezések szervezésének folyamata. Magába foglalja az incidens loggolását, osztályozását és kihatásának vizsgálatát. [Az IEEE 1044 szerint] → *incident management*

incidens menedzsment eszköz: olyan eszköz, amely megkönnyíti az incidensek felismerését és állapotainak nyomonkövetését. Gyakran munkafolyamat-központú annak érdekében, hogy az incidens allokációját, javítását és újratestelését támogassa, illetve hogy támogassa a jelentéskészítést. Lásd még: hiba menedzsment eszköz → *incident management tool*

indítás (IDEAL): az IDEAL modell egyik fázisa, amely során megalapozzuk a sikeres fejlesztési törekvéseket. A következő tevékenységeket tartalmazza: az összefüggések meghatározása, a támogatás kiépítése, az infrastruktúra biztosítása. → *initiating (IDEAL)*

informális felülvizsgálat: olyan felülvizsgálat, amely nem formális (dokumentált) eljárás alapul → *informal review*

inkrementális fejlesztési modell: fejlesztési életciklus, ahol a projekt részét alkotó kis lépések mindegyike egy kis részt tesz hozzá az általános projektkövetelményekhez. A követelményeket priorizálják és a megfelelő inkrementális egységben a prioritási sorrend szerint szállítják. Ezen életciklusmodell néhány (de nem minden) verziójában minden alprojekt egy „mini V-modell” szerint működik, saját tervezési, megvalósítási és tesztfázisokkal. → *incremental development model*

inkrementális teszt: olyan teszt, amikor a komponenseket egyesével, vagy legfeljebb

néhányukat együtt lépésenként integrálják és tesztelik egészen addig, amíg az összes komponenst össze nem integrálják, illetve együtt nem tesztelik → *incremental testing*

inspekció: az egyenrangú felülvizsgálat egy típusa, amely a dokumentum vizuális vizsgálatán alapul, hogy megtaláljuk a hibákat, vagy pl. a szabványokhoz képest meglevő különbségeket, illetve a magasabb szintű dokumentációktól való eltéréseket. A leginkább formális felülvizsgálati módszer, amely emiatt mindig dokumentált eljárás alapján. [IEEE 610, IEEE 1028 szerint] Lásd még: egyenrangú felülvizsgálat → *inspection*

inspekció vezető: lásd moderátor → *inspection leader*

inspektor: lásd felülvizsgáló → *inspector*

instrumentálás: kiegészítő kódok beszúrása a programba annak érdekében, hogy információkat kapjunk a program végrehajtása közbeni viselkedéséről, pl. kód lefedettség mérésekor → *instrumentation*

instrumentáló: az instrumentálást végrehajtó szoftver eszköz → *instrumenter*

integráció: a komponensek, vagy rendszerek nagyobb egységbe történő összeolvasztásának folyamata → *integration*

integrációs teszt: olyan teszt, amelynek célja az integrált egységek közötti interfészekben, illetve köcsönhatásokban lévő hibák megtalálása. Lásd még: komponens integrációs teszt, rendszer integrációs teszt → *integration testing*

Integrált Képességi-Érettségi Modell (CMMI): a hatékony termékfejlesztés és karbantartás alapelemeit leíró keretrendszer. A legjobb termékfejlesztési és karbantartási, tervezési és szervezési gyakorlatokat foglalja magában. A CMM utódja → *Capability Maturity Model Integration (CMMI)*

Integrált Teszt-Érettségi Modell (TMMi): :Az Integrált Képességi-Érettségi Modell (CMMI)-hez kapcsolódó, ötszintes keretrendszer a tesztfolyamatok javítására, amely a hatékony tesztfolyamat kulcselemeit írja le. → *Test Maturity Model Integrated (TMMi)*

interfész teszt: olyan integrációs teszt, amely az interfészekre koncentrálna → *interface testing*

intézkedés előrejelző: olyan intézkedés, amelyet egymásik intézkedés előrejelzésére, vagy becslésére használunk [ISO 14598] → *indicator*

Ishikawa diagram: lásd ok-okozat diagram. → *Ishikawa diagram*

iteratív fejlesztési modell: olyan fejlesztési életciklus, amely során a projektet nagyszámú iterációk sorozatára bontják. Egy iteráció egy olyan teljes fejlesztési ciklus, amely egy végrehajtható termék (belső, vagy külső) kiadását eredményezi, a végső termék egy részhalmazát, amely iterációról iterációra növekedve válik a végső terméké → *iterative development model*

jegyzőkönyv vezető: az a személy, aki naplózott formában rögzíti a felülvizsgálat során megemlített összes hibát és azokat a javaslatokat, amik a folyamat javítására

irányulnak. A jegyzőkönyv vezetőnek biztosítani kell, hogy a naplózás olvasható és érthető legyen. → *scribe, recorder*

karbantartás: az alkalmazás átadása után végzett módosítások: hiba javítások, teljesítmény vagy más jellemzők feljavítása illetve megváltozott környezetre/követelményekre való alkalmazása. [IEEE 1219] → *maintenance*

karbantartási teszt: módosítások vagy megváltozott környezet miatt a működő rendszeren végrehajtott teszt → *maintenance testing*

karbantarthatóság: egy alkalmazás azon tulajdonsága, hogy milyen egyszerűen lehet hibákat javítani benne, új vagy megváltozott követelmények illetve környezet miatt szükséges módosításokat eszközölni. [ISO 9126] → *maintainability*

karbantarthatósági teszt: teszteljárás egy alkalmazás karbantarthatóságának meghatározására → *maintainability testing*

kereskedelmi dobozos szoftver: lásd dobozos szoftver → *commercial off-the-shelf software*

Képesség-Érettség Modell (CMM): a hatékony szoftverfejlesztés kulcs lépéseit 5 szinten leíró keretrendszer. Magába foglalja a tervezés, illetve a szoftverfejlesztés bevált gyakorlatait. Lásd még: integrált Képességi-Érettségi Modell → *Capability Maturity Model (CMM)*

képzeletbeli futtatás: a program elméleti futtatása, amikor a szoftverfejlesztő lépésről lépésre megvizsgálja a forráskódot és megállapítja, hogy az a futáskor hogyan fog viselkedni → *dry run*

kiadási jelentés: lásd kiadási megjegyzés → *item transmittal report*

kiadási megjegyzés: tesztelemekeket, azok konfigurációját, aktuális státuszát és egyéb információkat azonosító dokumentum vagy feljegyzés. Ezt a teszt végrehajtási fázis kezdetekor a fejlesztők szállítják a tesztelőknek, esetleg a megrendelőnek.. [IEEE 829] → *release note*

kiépítés (IDEAL): az IDEAL modell azon fázisa, amely során meghatározzák, hogyan érje el a szervezet a célját. A kiépítés fázis a következő tevékenységeket foglalja magában: prioritások felállítása, fejlesztési megközelítés, tevékenységek tervezése → *establishment (IDEAL)*

kilépési feltétel: általános és speciális feltételek halmaza, amelyet minden érintettel egyeztetve egy folyamat hivatalos befejezési feltételének tekintünk. A célja, hogy megakadályozzuk az olyan feladatok befejezettnek tekintését, amelyeknek még vannak függőben levő, be nem fejezett részei. A kilépési feltételeket a tesztelés leállításának tervezéséhez és jelentéséhez használjuk. → *exit criteria*

kilépési pont: egy végrehajtható utasítás, vagy folyamatlépés, amely egy adott folyamat utolsó tervezett lépését határozza meg. → *exit point*

kimenet: egy (komponensen kívül vagy belül tárolt) változó, amelyet egy komponens írhat → *output*

kimeneti érték: a kimenet egy példánya. Lásd még kimenet → *output value*

kimeneti értéktartomány: a helyes kimeneti értékeket tartalmazó halmaz. Lásd még domain → *output domain*

kimerítő teszt: olyan teszt, amely során a tesztkészletünk magába foglalja a bemeneti értékek és előfeltételek összes kombinációját → *exhaustive testing*

kiváltó ok: a hiba forrása, melyet eltávolítva a hibajelenség is csökken vagy megszűnik. [CMMI] → *root cause*

kiváltó ok elemzés: ezen elemzési technikának az a célja, hogy a hibát kiváltó okokat azonosítsa. Amennyiben a hibát kiváltó okokat meg tudjuk határozni, és ezeket még valamilyen módon mérni is tudjuk, a hibák előfordulásának valószínűségét minimalizálni lehet. → *root cause analysis*

kivételkezelés: a szoftver azon viselkedése, hogy hogyan reagál a téves bemenetekre, amelyek akár téves felhasználásból, akár egy másik szoftvertől, akár belső meghibásodásból származik → *exception handling*

kockázat: az a tényező, amely a jövőben negatív következményeket okozhat. Általában, mint hatás és valószínűség jelenik meg. → *risk*

kockázat alapú teszt: a tesztelés egy olyan megközelítése, mely csökkenti a termék kockázati szintjét és tájékoztatja az érintetteket a kockázat mértékéről már a projekt kezdeti fázisában. Magában foglalhatja mind a termék kockázatainak azonosítását, mind a tesztfolyamat kockázat szerinti vezetését. → *risk-based testing*

kockázat azonosítás: folyamat a kockázatok azonosítására valamilyen technika / módszer használatával. Ilyen technika például a "brainstorming", az ötletlista, vagy a hibatörténet elemzés. → *risk identification*

kockázatelemzés: kockázati elemek elemzésének technikája. Az azonosított kockázatokhoz hozzárendeli a bekövetkezésük hatását és valószínűségét. A kockázatelemzés eredménye meghatározhatja a rendszer fejlesztésének, illetve teszttechnikáját → *hazard analysis, risk analysis*

kockázatcsökkentés: egy olyan folyamat, melyben megelőző méréseket végzünk, és döntéseket hozunk, hogy a megadott szintre csökkentsük, vagy azon a szinten tartjuk a kockázatokat. → *risk control, risk mitigation*

kockázati szint: egy kockázat nagysága megadható a hatásának és a valószínűségének karakterisztikájával. A kockázat nagysága, azaz a kockázati szint meghatározza a teszt intenzitását is. A kockázati szint kvalitatív (nagy, közepes, kicsi) vagy kvantitatív (30%) módon egyaránt megadható. → *risk level*

kockázat kategória: lásd: kockázattípus → *risk category*

kockázat kezelés: szisztematikus alkalmazása azon folyamatoknak és gyakorlatoknak, melyek a kockázatok azonosításával, elemzésével, priorizálásával és ellenőrzésével kapcsolatosak. → *risk management*

kockázattípus: a kockázatok valamilyen szempont szerinti csoportja, mint pl. minőségi attribútum, ok, a kockázat helye, vagy a lehetséges kihatás. A kockázat típusa összefügg a teszt típusával, amely csökkentheti (kezelheti) az adott kockázattípust.

Például a felhasználói interakciókkal kapcsolatos félreértések csökkenthetők a használhatósági teszttel. → *risk type*

kodependens viselkedés (kapcsolati függőség): egy másik személytől való túlzott érzelmi, vagy pszichológiai függőség, amely során látszólag megpróbáljuk megváltoztatni a másik személy jelenlegi (nem kívánatos) viselkedését, mialatt tudat alatt partnerek vagyunk a nemkívánatos viselkedés fenntartásában: Példa lehet erre a szoftvertesztelés során, amikor panaszkodunk a fejlesztők késlekedése miatt, de tudat alatt élvezzük a „hősi” szerepet, amelyet a túlórakkal vívunk ki magunknak, és ezáltal tulajdonképpen megerősítjük a nem kívánt viselkedést. → *codependent behaviour*

kompatibilitási teszt: lásd: együttműködőképességi teszt → *compatibility testing*

komplexitás: a szoftver tervének, illetve illetve belső struktúrájának összetettsége megértés, karbantartás, valamint verifikálás szempontjából. Lásd még: ciklomatus komplexitás → *complexity*

komponens: a legkisebb önállóan tesztelhető szoftver egység → *component*

komponens integrációs teszt: az integrált komponensek közötti kapcsolatok hibáinak felderítését szolgáló teszt → *component integration testing, integration testing in the small*

komponens specifikáció: a komponens viselkedésének leírása abban az értelemben, hogy a megadott bemeneti értékre bizonyos feltételek mellett milyen kimeneti érték várható el, valamint a szükséges nem-funkcionális működés meghatározása (pl. az erőforrások kezelése) → *component specification*

komponens teszt: az egyes szoftverkomponensek tesztje → *component testing*

konfiguráció: a rendszer, illetve a szoftver összeállítása az alkotóelemeinek száma, jellege és kapcsolatai alapján → *configuration*

konfiguráció azonosítás: a konfiguráció menedzsment része, amely a konfigurációs elemek kiválasztásából, illetve ezek funkcionális és fizikai karakterisztikáinak a műszaki dokumentációba történő feljegyzéséből áll → *configuration identification*

konfigurációs elem: azon hardver, illetve szoftver egységek összessége, amelyek a konfiguráció menedzsment egységeként kezelendők → *configuration item*

konfiguráció ellenőrzés: annak vizsgálata, hogy a konfigurációs elemek tartalma megfelel-e pl. a szabványoknak → *configuration auditing*

konfiguráció irányítás: a konfiguráció menedzsment része, amely a konfigurációs elemeknek a konfigurációs állapot formális rögzítése utáni változásainak kiértékelését, koordinálást, jóváhagyását, vagy éppen elutasítását, továbbá a megvalósítását foglalja magába → *configuration control*

konfiguráció irányító testület: lásd változás irányító testület → *configuration control board (CCB)*

konfiguráció menedzsment: a következő tevékenységek technikai és adminisztratív irányítása: a konfigurációs elemek funkcionális és fizikai karakterisztikáinak

meghatározása és dokumentálása, az ezen karakterisztikákhoz képest történő változás irányítása, a változáskezelési és megvalósítási állapot nyomon követése és jelentése, illetve a különböző követelményeknek történő megfelelés → *configuration management*

konfiguráció menedzsment eszköz: olyan eszköz, amely támogatja a konfigurációs elemek meghatározását és irányítását, ezek változásainak, illetve verzióinak, státuszainak nyomon követését, illetve felügyeli az ezeket tartalmazó alapverziók kiadását → *configuration management tool*

konfiguráció teszt: lásd hordozhatósági teszt → *configuration testing*

konverziós teszt: annak tesztje, hogy egy meglévő rendszerből hogyan lehet adatokat átkonvertálni a helyettesítő rendszerbe → *conversion testing*

kód: programnyelven, vagy a fordító, illetve az assembler kimeneteként kifejezett számítógépes utasítás és adat definíció → *code*

kód alapú teszt: lásd fehérdoboz teszt → *code-based testing*

kód elemző: statikus kód elemzést végző eszköz. Az eszköz megvizsgálja, hogy a forráskód mennyire felel meg a kódolási irányelveknek, a minőségi mutatóknak, illetve vizsgálja az adatfolyam rendellenességeit. → *code analyzer, static code analyzer*

kód lefedettség: elemző módszer, amely meghatározza, hogy a szoftver mely részei lettek végrehajtva (lefedve) a teszt végrehajtása során, és melyek nem. Például: utasítás lefedettség, döntési lefedettség, feltétel lefedettség → *code coverage*

komponens kapcsolati teszt: lásd komponens integrációs teszt → *link testing*

konfiguráció irányítás: a konfigurációkezelés eleme, amely magába foglalja a konfigurációs elemek változásainak a formális rögzítést követő kiértékelését, jóváhagyását és elutasítását, valamint a megvalósítását → *configuration control*

konkurencia teszt: annak vizsgálata, hogy a szoftver megfelelően kezeli-e a két, vagy több tevékenység egyidejű futtatását, amely párhuzamos futtatásból, illetve a különböző tevékenységek összefésüléséből származhat → *concurrency testing*

konzisztencia: a szoftver, illetve a dokumentáció egységességének, szabványosságának, illetve az ellentmondásoktól mentességének foka → *consistency*

követelmény: olyan feltétel vagy képesség, amely a felhasználó számára azért szükséges, hogy megoldjon egy problémát, vagy elérjen egy célt. Ezen feltételnek vagy képességnek a rendszer vagy rendszer komponens által is megvalósíthatónak kell lennie úgy, hogy közben a szerződés, szabvány, specifikáció és egyéb formális dokumentumban támasztott követelményeknek is megfeleljen. [IEEE 610] → *requirement*

követelmény alapú teszt: a tesztelés egy olyan megközelítése, melyben a tesztesetek tervezése a követelményekből származtatott tesztcélok és feltételek alapján történik. Például speciális funkciók tesztjei vagy olyan nem-funkcionális jellemzők tesztjei, mint

a megbízhatóság vagy használhatóság. → *requirements-based testing*

követelmény-menedzsment eszköz: olyan eszköz, amely támogatja a követelmények, a követelmény jellemzőinek (pl. prioritás, tudás gazda) rögzítését és magyarázó jegyzetek készítését, valamint megkönnyíti a különböző szintű követelmények nyomon követhetőségét és változás menedzsmentjét. Vannak olyan követelmény-menedzsment eszközök, melyek a statikus elemzéshez is segítséget nyújtanak, például konzisztencia ellenőrzéssel vagy előre definiált, követelményekre vonatkozó szabályokkal. → *requirements management tool*

követelmény feltárási fázis: a szoftver életciklus azon fázisa, melyben a szoftvertermékkel kapcsolatos követelmények definiálása és dokumentálása történik. [IEEE 610] → *requirements phase*

kritikus sikertényező: a szervezet küldetésének eléréséhez nélkülözhetetlen elem. Ezek olyan kritikus tényezők, vagy tevékenységek, amelyek biztosítják a sikert. Lásd még: tartalom-alapú modell → *critical success factor*

kritikus tesztfolyamat: nagyjából 12 kulcsfontosságú folyamatból felépített, a tesztfolyamat fejlesztés szolgáló tartalom-alapú modell. Célja, hogy jól áttekinthetően mutassa a menedzsment részére a szervezet küldetése szempontjából legfontosabb folyamatokat, ahol a teljesítmény leginkább befolyásolja a cég nyereségét és hírnevét. → *Critical Testing Process (CTP)*

kulcsfontosságú teljesítményjelző: magas szintű mutató, amely a fejlesztés eredményességét és/vagy hatékonyságát mutatja. Azt jelzi, hogy adott célt elértünk-e és milyen feltételekkel (mondjuk idő vagy költség szempontból). [CMMI] → *key performance indicator*

kulcsszó alapú teszt: egy olyan, szkripteken alapuló teszt technológia, amelynél a tesztszkriptekben nem csak tesztadatok és az elvárt eredmény található, hanem az alkalmazással kapcsolatos vezérlő kulcsszavak is. → *action word driven testing, keyword driven testing*

LCSAJ: a LCSAJ (Linear Code Sequence and Jump) kifejezés három összetevőt takar (amelyet a forráskód egy-egy sorával azonosíthatunk): az egymást követő lineáris utasítás sorok kezdő és végpontját, valamint azt a kód helyet, ahova a vezérlés kerül a zárósor után → *LCSAJ*

LCSAJ lefedettség: egy komponensben levő összes LCSAJ kódrészletből azok százalékos aránya, amelyek a teszt futtatása során végrehajtásra kerültek. 100% LCSAJ lefedettség 100%-os döntési lefedettséget jelent. Lásd: döntési lefedettség → *LCSAJ coverage*

LCSAJ teszt: fehérdoboz teszttervezési módszer, amelynek célja, hogy a tesztesetek LCSAJ kódsorokat járjanak be → *LCSAJ testing*

lefedettség: annak százalékos mérőszáma, hogy az adott lefedettségi elemet milyen arányban hajtotta végre a teszteszköz → *coverage*

lefedettség elemzés: az adott lefedettségi elem a teszt során elért lefedettségének elemzése, amely utal az előre meghatározott szempontra, eldönti, hogy további

tesztek szükségesek-e, és ha igen, milyen formában → *coverage analysis*

lefedettség-mérő eszköz: lásd lefedettségi eszköz → *coverage measurement tool*

lefedettségi elem: olyan entitás, vagy jellemző, amely a test lefedettség alapjául szolgál, pl. ekvivalencia partíciók, vagy kód utasítások → *coverage item*

lefedettségi eszköz: olyan eszköz, amely objektíven méri, hogy mely strukturális elemeket – pl. utasításokat, elágazásokat - hajt végre a tesztkészlet → *coverage tool*

lépcsőzetes felépítés: egy modell struktúra, amiben adott folyamatok céljainak megvalósításával egy érettségi szintet érünk el; minden egyes szint a következő szint alapját képezi. [CMMI] → *staged representation*

logikai teszteset: lásd magas szintű teszteset → *logical test case*

magas szintű teszteset: olyan teszteset, amely nem rendelkezik konkrét, megvalósítási szintű bemenő adattal, illetve elvárt eredménnyel. Logikai műveleteket használ. Az aktuális értékek példányai nem meghatározottak és/vagy nem hozzáférhetők. Lásd még: alacsony szintű teszteset → *high level test case*

megbízhatóság: a szoftvertermék azon képessége, hogy a szükséges, előírt funkcionalitást meghatározott időtartam vagy meghatározott műveletszám mellett az előre megállapított követelmények szerint látja el. [ISO 9126] → *reliability*

megbízhatóság-növekedés modell: olyan modell, mely megmutatja, hogy folyamatos tesztelés mellett és azon hibák javításával, melyek a megbízhatóságot érintették, hogyan növekszik egy komponens vagy rendszer megbízhatósága. → *reliability growth model*

megbízhatósági teszt: a teszt folyamata meghatározza egy szoftvertermék megbízhatóságát. → *reliability testing*

megfelelőség: a szoftvertermék azon képessége, hogy megfelelő funkciókat biztosítson meghatározott feladatokhoz és felhasználói célokhoz. Lásd még funkcionalitás. [ISO 9126] → *suitability*

megfelelőségi teszt: a szoftver megfelelőségét vizsgáló teszt. → *suitability testing*

megjelenés: annak vizsgálata, hogy a szoftver mennyire attraktív a felhasználó felé → *attractiveness*

meghajtó: egy szoftver komponens, vagy teszt eszköz, amely kiváltja azt a komponens, amely egy másik komponens, vagy a rendszer vezérlését, és/vagy felhívását végzi → *driver*

meghatározott bemeneti érték: egy bemenet, amire a specifikáció megadja az eredményt. → *specified input*

meghibásodás: a komponens, illetve a rendszer eltér az elvárt eredménytől, vagy szolgáltatástól → *failure*

meghibásodási ráta: adott kategóriájú meghibásodások aránya egy adott mértékegységben kifejezve, pl. időegységhez, a tranzakciók számához, vagy a futó gépek számához viszonyítva → *failure rate*

Meghibásodások Közti Átlagidő: a rendszer meghibásodásai között eltelt átlagidő (számtani közép). A Meghibásodások Közti Átlagidő (Mean Time Between Failures, MTBF) jellemzően a megbízhatóság-növekedés modell része, amely azt feltételezi, hogy a rendszert a hibajavító folyamat keretében azonnal javítják, Lásd még megbízhatóság-növekedés modell. → *Mean Time Between Failures (MTBF)*

memóriaszivárgás: Memóriáhozáférési meghibásodás, amelyet a program dinamikus tárhely allokációjában levő programhiba okoz, amely miatt nem szabadul fel az adott memóriaterület a használat után. A memóriaszivárgás a program és/vagy a párhuzamos folyamatok leállításához vezethet → *memory leak*

menedzsment felülvizsgálat: a beszerzési-, szolgáltatási-, fejlesztési-, üzemeltetési- és karbantartási eljárások rendszeres kiértékelése a menedzsment által vagy annak megbízásából. A kiértékelés célja, hogy ellenőrizze, hogy a feladatok ellátása az (ütem)tervnek megfelelően történik-e, jóváhagyja a követelmények teljesítését, valamint, hogy kiértékelje a menedzsment (saját maga) által a cél elérésére tett lépéseinek eredményességét. [After IEEE 610, IEEE 1028] → *management review*

mennyiségi teszt: a rendszer funkcióit nagy mennyiségű adattal vizsgáló teszt módszer. Lásd még erőforráshasználat teszt → *volume testing*

mérés: az az eljárás amivel egy egyed egy tulajdonságához számosságot vagy kategóriát rendelünk. [ISO 14598] → *measurement*

mérési mellékhatás: egy rendszer vagy komponens mérése során a mérőeszköz által okozott hatás; például teljesítményteszt során a tesztelő illetve mérőeszköz rontja a rendszer vagy komponens teljesítményét (még ha nem is onnan futtatjuk, de monitorozzuk). → *probe effect*

mérési skála: az adatelemzés típusát meghatározó skála (névleges, sorrendi, különbségi vagy aranyskála). [ISO 14598] → *measurement scale*

mérföldkő: a projekt egy időpontja, amelyre meghatározott minőségű és mennyiségű (köztes) leszállítandót és eredményt kell felmutatni → *milestone*

mérőszám: tetszőleges egyed egy tulajdonságához mérés során hozzárendelt számossága vagy kategóriája. [ISO 14598] → *measure*

metrika: mérésnél használatos skála vagy módszertan. [ISO 14598] → *metric*

migrációs teszt: lásd konverziós teszt → *migration testing*

minőség: az a szint, amikor a komponens, rendszer vagy folyamat megfelel a meghatározott követelményeknek és/vagy a felhasználó/ügyfél igényeinek és elvárásainak. [IEEE 610] → *quality*

minőségbiztosítás: a minőségmenedzsment azon eleme, mely azért felel, hogy kielégítsük a minőséggel szemben támasztott követelményeket. [ISO 9000] → *quality assurance*

minőségellenőrző kapu: a projekt egy speciális mérföldköve. Azon fázisok közé szokták beékelni, ahol az egyik fázis eredménye nagyban függ az előtte levő fázis kiementétől. A minőségellenőrző kapu formálisan ellenőrzi a megelőző fázis

dokumentumait. → *quality gate*

minőségi jellemző: olyan tulajdonság, vagy jellemző, amely befolyásolja egy elem minőségét. [IEEE 610] → *quality attribute, quality characteristic*

minőség költsége: a minőséggel kapcsolatos tevékenységek teljes költsége, amelyek általában megelőzési költségből, a kiértékelési költségekből, a belső és külső meghibásodások költségeiből áll → *cost of quality*

minőségmenedzsment: koordinált tevékenységek egy szervezet irányítására és ellenőrzésére a minőség szempontjából. Az irányítás és ellenőrzés minőségi szempontjai általában a következő tevékenységeket foglalják magukba: minőségi előírások, irányelvek készítése, minőségtervezés, minőségellenőrzés, minőségbiztosítás, minőségjavítás. [ISO 9000] → *quality management*

minősítés: egy folyamat, mely tanúsítja az elvárt követelményeknek való megfelelést. A „minősített” fogalmat a megfelelő státusz megnevezésére használják. [ISO 9000] → *qualification*

modellező eszköz: egy alkalmazás, vagy egy rendszer modelljeinek létrehozását, javítását és verifikálását támogató eszköz.. [Graham] → *modelling tool*

moderátor: egy inspekció, vagy felülvizsgálat levezetéséért felelős kulcsembert vagy vezető → *moderator*

modul: lásd: komponens → *module*

modulteszt: lásd: komponens teszt → *module testing*

módosított döntési feltétel lefedettség: lásd: feltétel meghatározási lefedettség → *modified condition decision coverage*

módosított döntési feltétel teszt: lásd: feltétel meghatározási teszt → *modified condition decision testing*

módosított többszörös feltétel lefedettség: lásd: feltétel-kombinációs lefedettség → *modified multiple condition coverage*

módosított többszörös feltétel teszt: lásd: feltétel meghatározási teszt → *modified multiple condition testing*

Módszeres Teszt és Kiértékelési Folyamat: egy olyan strukturált teszt módszert, amelyet tartalom-alapú modellként a teszt folyamat fejlesztésére használnak. A Módszeres Teszt és Kiértékelési Folyamat (STEP) nem követeli meg, hogy a fejlesztések meghatározott sorrendben történjenek. Lásd még: tartalom-alapú modell → *Systematic Test and Evaluation Process (STEP)*

munkamenet-alapú teszt vezetése: a munkamenet alapú teszt mérésének és irányításának módszere → *session-based test management*

munkamenet-alapú teszt: olyan tesztelési megközelítés, amely során a teszttevékenységeket a műszaki teszttervezés és a végrehajtás megszakítás nélküli menetként tervezik. Általában a felderítő tesztelssel kapcsolatosan használják → *session-based test*

Munkaelem Strukturálás: egy munka elemeinek, továbbá az egymáshoz és a termékhez való kapcsolatának elrendezése. [CMMI]. → *Work Breakdown Structure (WBS)*

mutációs elemzés: egy tesztkészlet alaposságát vizsgálhatjuk ezen módszerrel úgy, hogy megfigyeljük, hogy a tesztkészlet mennyire tesz különbséget (mennyire hoz más eredményt) eredeti program kód és variánsain (mutánsain) történő futtatása között → *mutation analysis*

mutációs teszt: lásd összehasonlító teszt → *mutation testing*

mutató: adatelem mely egy másik adatelem helyét határozza meg, például: egy ilyen adatelem tartalmazza a következő feldolgozandó alkalmazott rekord címét. [IEEE 610] → *pointer*

működési átvételi teszt: az átvételi tesztfázisban általában egy szimulált, életszerű környezetben az üzemeltetők és vagy adminisztrátorok által végrehajtott működési teszt, amely működési szempontból (pl.: visszaállíthatóság, erőforrás-kezelés, telepíthetőség és technikai szabványosság) vizsgálja a rendszert,. Lásd még: működési teszt → *operational acceptance testing*

működési környezet: a felhasználók vagy vevők telephelyén levő szoftver és hardver termékek, amelyekben vagy amelyekkel a tesztelendő komponens vagy rendszer futni fog. Ilyen szoftverek például az operációs rendszer illetve az adatbáziskezelő rendszer → *operational environment*

működési profil: egy komponens, vagy rendszer által végrehajtott különböző feladatok, és ezek valószínűségének reprezentációja, amely a felhasználói várható interakcióin alapul. Ez a feladat inkább logikai típusú, amely lefutása akár több gépen és nem folytonos időintervallumban is történhet → *operational profile*

működési profil teszt: egy komponens, vagy rendszer által végrehajtott különböző feladatok, és ezek valószínűségének reprezentációja, amely a várható felhasználói interakciókon alapul. Ez a feladat inkább logikai típusú, amely lefutása akár több gépen és nem folytonos időintervallumban is történhet → *operational profile*

működési teszt: egy komponensen, vagy rendszeren futtatott teszt, amivel megfigyelhetjük a komponens vagy rendszer működési környezetben való viselkedését. [IEEE 610] → *operational testing*

működtethetőség: egy szoftverterméknek a felhasználók számára nyújtott működtetési, vezérelhetőségi tulajdonsága. [ISO 9126] Lásd még használhatóság → *operability*

műszaki teszterv (test design) specifikáció: a tesztfeltételeket és követelmények teljesítését definiáló dokumentáció, amely tartalmazza a tesztelési megközelítést és a magas szintű teszteseteket (IEEE 829 alapján) → *test design specification*

műszaki tesztervezés:

(1) Lásd műszaki teszterv (test design) specifikáció

(2) Az általános tesztcélok átalakítása mérhető tesztfeltételekké és tesztesetekké

→ *test design*

műszaki tesztervezési technika: a tesztesetek készítésére, származtatására és kiválasztására alkalmazott eljárás. → *test design technique*

műszaki tesztervező eszköz: a műszaki tesztervezési tevékenységet támogató eszköz, amely tesztbemeneteket generál egyéb forrásokból, pl. CASE eszközökből, követelmény menedzsment eszközökből, tesztfeltételekből valamint magából a programkódból. → *test design tool*

N-lépéses lefedettség: egy tesztkészlet által bejárt egymás utáni N+1 átmenet. [Chow]
→ *N-switch coverage*

N-lépéses teszt: az állapotátmenet teszt egy fajtája, amelyben a teszteseteket úgy tervezték, hogy az N+1 átmenet összes valós utasítását bejárják. [Chow] Lásd még: állapotátmenet teszt → *N-switch testing*

nagy bumm teszt: az integrációs teszt egyik fajtája, ahol a szoftver és a hardver elemeket akár egyetlen rendszerbe integrálva teszteljük → *big-bang testing*

napi build: olyan fejlesztői tevékenység, amely során minden nap során a teljes rendszert lefordítják és összeszerkesztik (általában éjszaka), hogy mindig elérhető legyen egy olyan konzisztens rendszer, amely tartalmazza a legfrissebb változtatásokat → *daily build*

negatív teszt: olyan teszt, amelynek az a célja, hogy megmutassa, hogy a szoftver nem működik. A negatív teszt inkább a tesztelők hozzáállásában, nem pedig egy teszt technikában vagy módszerben rejlik, pl.: rossz, nem elvárt bemenő értékekkel való teszt. → *dirty testing, negative testing*

nem-szabályosság: egy meghatározott követelmény be nem teljesülése. [ISO 9000] → *non-conformity*

nem-funkcionális követelmény: olyan követelmény, amely a funkcionalitáshoz nem, de a megbízhatósághoz, hatékonysághoz, használhatósághoz, karbantarthatósághoz és hordozhatósághoz kapcsolódik → *non-functional requirement*

nem-funkcionális teszt: egy komponens vagy rendszer funkcionalitáshoz nem kapcsolódó tulajdonságainak tesztje, mint például megbízhatóság, hatékonyság, használhatóság, karbantarthatóság és hordozhatóság → *non-functional testing*

nem-funkcionális tesztervezési technika: olyan eljárás, amellyel a nem-funkcionális teszt tesztkészletéből kiválasztunk, vagy származtatunk teszteseteket, amiket a tesztelendő komponens, vagy rendszer specifikációjának elemzése alapján határoztunk meg anélkül, hogy ismernénk a belső struktúrát. Lásd feketedoboz tesztervezési technika → *non-functional test design technique*

nyomonkövethetőség: a dokumentáció és a szoftver összefüggő egységeinek vizsgálata, pl. a követelmények és a hozzájuk tartozó tesztesetek közötti kapcsolatra. Lásd függőleges nyomonkövethetőség, vízszintes nyomonkövethetőség → *traceability*

ok-okozat diagram: egy probléma különböző lehetséges kiváltó okai közötti kapcsolat grafikai megjelenítését szolgálja. A tényleges, vagy lehetséges hiba, vagy

meghibásodás lehetséges okait kategóriákként és al-kategóriákként egy fastruktúrában jeleníti meg, ahol a (lehetséges) hiba, vagy meghibásodás a csomópont (Juran szerint), → *cause-effect diagram*

ok-okozat döntési tábla: olyan táblázat, amely a bemenetek, és/vagy a különböző kiváltó okok és a hozzájuk kapcsolt eredmények, és/vagy hatások kapcsolatát mutatja → *cause-effect decision table, decision table*

ok-okozati gráf: a bemenetek és/vagy a különböző folyamatok kiváltói (okok) grafikai megjelenítése a hozzájuk kapcsolt kimenetekkel (okozatok) együtt, amely tesztesetek tervezéséhez használható → *cause-effect graph*

ok-okozat teszt: olyan feketedoboz teszttechnika, amely az ok-okozati gráf alapján tervezi a teszteseteket → *cause-effect graphing*

okozati elemzés: a hibák analízise a kiváltó ok meghatározása érdekében → *causal analysis*

orákulum: lásd teszt-orákulum → *oracle*

ortogonális mátrix: egy kétdimenziós tömb (mátrix) a következő matematikai tulajdonsággal: bármely két oszlopot kiválasztva a soronkénti párok a tömb által tartalmazott számok összes kombinációját lefedik → *orthogonal array*

ortogonális mátrix teszt: a változó-párok összes lehetséges kombinációjának módszeres tesztje ortogonális mátrixok használatával. Ez a módszer jelentősen kevesebb tesztesetet igényel, mint az egyes változók összes lehetséges kombinációjának tesztje. Lásd még páronkénti teszt → *orthogonal array testing*

osztályozási fa: olyan fa, amely a hierarchikusan rendezett ekvivalencia partíciókat mutatja. Lásd még: osztályozási fa módszer → *classification tree*

osztályozási fa módszer: feketedoboz teszttervezési technika, amely során az osztályozási fa segítségével tervezünk teszteseteket annak érdekében, hogy reprezentatív be- és kimeneteket vizsgáljunk → *classification tree method*

összefoglaló tesztjelentés: a teszttvékenységeket és eredményeket tartalmazó dokumentum. Ebben a dokumentumban található a kilépési feltételeknek megfelelően ellenőrzött tesztelemelek kiértékelése is. → *test summary report*

összehasonlító: lásd: teszt összehasonlító eszköz → *comparator*

összehasonlító teszt: funkcionálisan megegyező szoftverkomponensek kimenő adatainak összehasonlítása ugyanolyan bemenő adatok esetén → *back-to-back testing*

Pareto elemzés: döntéshozatali statisztikai módszer, amely során kiválasztják azon korlátozott számú tényezőt, amelyeknek szignifikáns kihatása van az eredményre. A minőségfejlesztés tekintetében a parablémák nagy többségét (80%) csupán néhány hibaok (20%) okozza. → *Pareto analysis*

páros programozás: egy olyan szoftverfejlesztési módszer, amelyben egy komponens minden egyes programsorát (alkalmazás, vagy teszt) két programozó egy számítógép előtt ülve írja meg. Ez egyben egy valós-idejű kód felülvizsgálatot is

magában foglal → *pair programming*

páros teszt: két személy, pl.: két tesztelő, egy fejlesztő és egy tesztelő, vagy egy végfelhasználó és egy tesztelő együttesen hibakereséssel foglalkoznak. Általában egy gép előtt ülve felváltva folytatják a tesztet → *pair testing*

páronkénti teszt: feketedoboz technika, amely során olyan teszteseteket készítünk, amelyek minden egyes bemeneti paraméter pár összes lehetséges diszkrét kombinációját letesztelik. Lásd még: ortogonális mátrix teszt → *pairwise testing*

partíciós teszt: lásd ekvivalencia partíció. [Beizer] → *partition testing*

pontosság: annak vizsgálata, hogy a szoftver mennyire képes az elvárt eredményeket, illetve hatásokat a kívánt pontossággal létrehozni. Lásd még: funkcionális teszt → *accuracy*

pontossági teszt: a szoftver pontosságát meghatározását célzó teszt → *accuracy testing*

prioritás: egy elemhez rendelt (üzleti) fontosság, például hiba prioritás. → *priority*

probléma: lásd programhiba → *problem*

probléma jelentés: lásd hibajelentés → *problem report*

probléma menedzsment: lásd hiba menedzsment → *problem management*

programhiba: a program olyan belső hibája, amely azt eredményezheti, hogy a szoftver nem tudja teljesíteni az elvárt viselkedését, azaz a program meghibásodásához vezethet → *bug, defect, fault*

programlogika lefedettségi teszt: lásd fehérdoboz teszt. [Myers] → *logic-coverage testing*

programlogika alapú teszt: lásd fehérdoboz teszt. [Myers] → *logic-driven testing*

program-instrumentáló: lásd instrumentáló → *program instrumenter*

programteszt: lásd komponens teszt → *program testing*

progresszív integráció alapú teszt: a komponensek integrációs tesztjéhez tartozó tevékenység, amely a követelmények részalmazainak megvalósítását követi, nem a komponensek integrációjának hierarchikus szerkezetét. → *thread testing*

projekt: a projekt egy időben, költségekben és erőforrásokban korlátozott, adott követelményeknek megfelelő cél érdekében kezdő és végidőpontokkal ellátott koordinált és kontrollált tevékenységek halmaza. → *project*

projektkockázat: egy (teszt) projekt menedzsmentjénél, irányításánál felmerülő kockázat, mint például emberhiány, szoros határidők, változó követelmények, stb. Lásd még kockázat → *project risk*

projekt teszterv (test plan): lásd fő teszterv (master test plan) → *project test plan*

projekt visszatekintő: a tanulságok levonásának és a speciális akciótervek elkészítésének strukturált módja a következő projektfázis, vagy projekt fejlesztése

érdekében. → *project retrospective*

pszeudó-véletlen: egy számsorozat, amely véletlenszerűnek tűnik, de valójában egy előre legenerált rendezett számsor egyes elemeit tartalmazza → *pseudo-random*

puffer: eszköz, vagy tárterület, amely adatok átmeneti tárolására szolgál → *buffer*

puffer túlcsoordulás: memória hozzáférési hiba, amely azáltal jön létre, hogy a lehetségesnél nagyobb mennyiségű, illetve méretű adatot próbálunk meg tárolni fix hosszúságú tárolóban, ezáltal a szomszédos memóriaterületek felülírásódnak, illetve túlcsoordulási hiba jelentkezik. Ld még puffer → *buffer overflow*

Rational Egységesített Folyamat (Rational Unified Process, RUP): egy szabadalmaztatott rugalmas iteratív szoftverfejlesztési folyamat keretrendszer, amely 4 projekt életciklus fázisból áll: Elkészítés Kidolgozás Megvalósítás Átadás → *Rational Unified Process (RUP)*

regressziós teszt: egy korábban már tesztelt program módosítást követő tesztje annak biztosítása érdekében, hogy a módosítás nem okozott hibát a szoftver nem módosított részeiben. A teszt végrehajtása a szoftver vagy a szoftverkörnyezet változtatásakor történik. → *regression testing*

rendelkezésre állás: annak meghatározása, hogy a komponens, illetve a szoftver mennyire elérhető és működőképes, amikor szükségessé válik a használata. Gyakran százalékban adják meg → *availability*

rendellenesség: a követelmény-specifikációtól, a tervezési, vagy felhasználói dokumentációtól, a szabványoktól, illetve bármilyen elvárástól, vagy tapasztalattól való eltérés. Rendellenességet lehet találni a felülvizsgálat, a tesztelés, az elemzés, a fordítás, vagy a program használata során. Lásd még: hiba, eltérés, meghibásodás, incidens, probléma → *anomaly*

rendszer: bizonyos funkciók, vagy funkciók halmazának elvégzése érdekében összeszervezett komponensek összessége. [IEEE 610] → *system*

rendszerek rendszere: hálózatokban több szinten egymásba ágyazott, több tartományban összekapcsolt összetett heterogén elosztott rendszerek, amiket átfogó, több szakterületen átívelő általános problémák és feladatok megoldására hozunk létre, jellemzően közös irányítási struktúra nélkül. → *system of systems*

rendszer-integrációs teszt: rendszerek és szoftvercsomagok integrációs tesztje; külső szervezetek interfészeinek tesztje (pl. Internet, Electronic Data Interchange) → *system integration testing*

rendszertereszt: integrált rendszer tesztje abból a célból, hogy ellenőrizzük a követelményeknek való megfelelést. [Hetzel] → *system testing*

robosztusság: annak fokmérője, hogy egy komponens vagy rendszer mennyire képes az elvárt működésre érvénytelen bemenetek vagy szűkös környezeti erőforrások mellett. [IEEE 610] Lásd még hibátűrés → *robustness*

robosztussági teszt: egy szoftvertermék robusztusságának meghatározására vonatkozó teszt. → *robustness testing*

SCRUM: iteratív, inkrementális keretrendszer az általában agilis szoftverfejlesztést alkalmazó projektekhez. Lásd még: agilis szoftverfejlesztés → *SCRUM*

siker teszt: egy teszt sikeresnek minősül, ha a kapott eredmény megfelel az elvárt eredménynek → *pass*

siker/bukás feltétel: döntési szabályok, amelyekkel eldöntjük, hogy egy tesztelem (funkció) vagy feature megfelel-e vagy megbukik-e egy teszten. [IEEE 829] → *pass/fail criteria*

sikertelen teszt: lásd bukás → *test fail*

skálázhatóság: a szoftver azon képessége, hogy az upgrade után alkalmazkodjon a megnövekedett terheléshez. [Gerrard szerint]. → *scalability*

skálázhatósági teszt: olyan teszt, amellyel a szoftvertermék skálázhatóságát határozzuk meg. → *scalability testing*

smoke teszt: a definiált, illetve tervezett tesztek egy olyan halmaza, amely a komponens, illetve a rendszer fő funkcionalitásait hivatott tesztelni abból a célból, hogy meggyőződjünk arról, hogy a program legkritikusabb részei működnek-e. A teszt során nem megyünk bele a részletekbe. A napi integráció és a smoke tesztek a leggyakrabban használt ipari eljárások közé tartoznak. → *smoke test*

specifikáció: egy dokumentum, ami ideális esetben teljesen, precízen és ellenőrizhetően határozza meg egy komponens vagy rendszer követelményeit, konstrukcióját, viselkedését vagy egyéb karakterisztikáit. A követelmények ellenőrzésének a folyamatát is meghatározhatja. [IEEE 610] → *specification*

specifikáció alapú teszt: lásd feketedoboz tesztervezés. → *specification-based testing*

specifikáció alapú technika: lásd feketedoboz tesztervezés. → *specification-based technique*

specifikáció alapú tesztervezési technika: lásd feketedoboz tesztervezési technika. → *specification-based test design technique*

stabilitás: a szoftvertermék azon képessége, hogy a szoftverben történt módosítások nem várt hatásait elkerülje. Lásd még karbantarthatóság. [ISO 9126] → *stability*

statikus elemzés: a szoftverelemek (például követelmények vagy kód) elemzése azok futtatása nélkül. Általában támogató eszközök segítségével végzik. → *static analysis*

statikus elemző eszköz: olyan eszköz, ami statikus elemzést hajt végre → *static analysis tool, static analyzer*

statikus forráskód elemzés: a forráskód elemzése anélkül, hogy a szoftvert futtatnánk. → *static code analysis*

statikus forráskód elemző: olyan eszköz, ami ellenőrzi a forráskód bizonyos tulajdonságait, például, hogy nincsenek-e adatfolyam anomáliák a kódban, vagy hogy megfelel-e a kódolási szabványoknak, minőségi előírásoknak. → *static code analyzer*

statikus teszt: egy komponens vagy rendszer tesztje specifikáció vagy megvalósítási szinten a szoftver futtatása nélkül. Például felülvizsgálat, vagy statikus forráskód

elemzés. → *static testing*

statisztikai teszt: olyan műszaki tesztervezési technika, amiben az inputok statisztikai eloszlás modelljét használjuk reprezentatív tesztesetek készítéséhez. Lásd még működési profil teszt → *statistical testing*

stressz teszt: olyan teljesítményteszt típus, amikor úgy vizsgálunk egy komponenst vagy rendszert, hogy az előre elvárt vagy annál nagyobb terheléssel, vagy csökkentett erőforrás rendelkezésre bocsátással teszteljük. Például korlátozott szerver memória hozzáférés. Lásd még teljesítményteszt, terheléses teszt. [IEEE 610] → *stress testing*

stressz teszteszköz: stressz tesztet támogató eszköz. → *stress testing tool*

struktúra alapú teszt: lásd fehérdoz teszt → *structurebased testing*

struktúra alapú technika: lásd fehérdoz tesztervezési technika. → *structure-based technique*

struktúra alapú tesztervezési technika: Lásd fehérdoz tesztervezési technika → *structure-based test design technique*

struktúra szintű lefedettség: egy komponens vagy rendszer belső struktúrája alapján meghatározott lefedettség → *structural coverage*

struktúra szintű tesztervezési technika: lásd fehérdoz tesztervezési technika. → *structural test design technique*

struktúra alapú teszt: lásd fehérdoz teszt. → *structural testing*

strukturált átvizsgálás: lásd átvizsgálás → *structured walkthrough*

súlyosság: valamely hiba hatásának a mértéke a rendszer vagy komponens fejlesztésére vagy működésére . [IEE 610] → *severity*

szabályosság: a szoftver mennyire tartja be a szabványokat, a megállapodásokat, a szabályokat és hasonló előírásokat → *compliance*

szabályossági teszt: olyan teszt, amely a szoftver szabályosságát hivatott vizsgálni → *compliance testing, conformance testing, regulation testing*

szabvány: formális, esetleg szerint kötelező jellegű követelmények, amelyeket egy adott munka elvégzési módjának következetes előírására, illetve irányelvek meghatározására használnak (pl. ISO/IEC szabványok, IEEE szabványok és szervezeti szabványok) [CMMI] → *standard*

szabványossági teszt: lásd szabályossági teszt → *standards testing*

számítógéppel támogatott szoftverfejlesztés (CASE): mozaikszó - Computer Aided Software Engineering, azaz számítógép által támogatott szoftvertervezés → *CASE*

számítógéppel támogatott szoftvertesztelés (CAST): mozaikszó - Computer Aided Software Testing, azaz számítógép által támogatott szoftvertesztelés. Lásd még: tesztautomatizálás → *CAST*

széleskörű Delphi eljárás: olyan szakértő alapú tesztbecslési technika, amely a csapat

tagjainak együttes tudását használja fel a minél pontosabb becslések készítésére. → *Wide Band Delphi*

szimuláció: egy fizikai vagy egy absztrakt rendszer kiválasztott viselkedési karakterisztikáinak egy másik rendszer által történő ábrázolása. [ISO 2382/1] → *simulation*

szimulátor: a teszt során használt eszköz, számítógépes program vagy rendszer, ami úgy viselkedik, vagy úgy működik, mint egy adott rendszer, melyet megfelelő inputtal vezérlünk. Lásd még emulátor. [IEEE 610, DO178b] → *simulator*

szintaxis teszt: egy feketedoboz tesztervezési technika, melyben a tesztesetek tervezése a definiált be- vagy kimeneti értelmezési tartomány alapján történik. → *syntax testing*

szkript alapú teszt: olyan tesztfutttatás, amelyet a tesztek korábban dokumentált sorozatának megfelelően hajtunk végre. → *scripted testing*

szkript nyelv: a tesztvégrehajtási eszközök (pl. felvevő/lejátszó eszközök) által is használt programozási nyelv, amiben futtatható teszt szkripteket lehet írni. → *scripting language*

szoftver: számítógépes programok, folyamatok és esetlegesen a számítógépes rendszer üzemelésére vonatkozó dokumentációk és adatok. [IEEE 610] → *software*

Szoftverfolyamat Fejlesztés: a szervezet szoftverrel összefüggő folyamatainak hatékonyságának és érettségének fejlesztését megcélzó tevékenységsorozat, és annak eredménye [CMMI] → *Software Process Improvement (SPI)*

szoftveres támadás: lásd támadás. → *software attack*

szoftver feature: lásd feature. → *software feature*

szoftver-életciklus: egy időperiódus, ami akkor kezdődik, amikor a szoftverterméket kigondoljuk és akkor fejeződik be, amikor a szoftvert már nem használják. A szoftver életciklus jellemzően a következő fázisokat tartalmazza: koncepció, követelmény, tervezés, megvalósítás, teszt, installáció és ellenőrzés, operáció és üzemeltetés, valamint időnként egy leállítási fázist. Megjegyzés: a fázisok átfedhetnek egymást, vagy akár ismétlődhetnek is. → *software life cycle*

Szoftver Használhatósági Kérdőív: egy kérdőív alapú használhatósági teszt technika, amivel egy komponens vagy rendszer használhatóságát lehet kiértékelni. Például: felhasználói elégedettséget mérő kérdőív [Veenendaal] → *Software Usability Measurement Inventory (SUMI)*

szoftver hibafa elemzés: lásd hibafa elemzés → *software fault tree analysis (SFTA)*

szoftverminőség: a szoftvertermék funkcionalitásainak és jellemzőinek az összessége, amelyek az explicit és implicit módon megadott szükségleteket elégítik ki. [ISO 9126] → *software quality*

szoftver minőségi jellemző: lásd minőségi jellemző → *software quality characteristic*

szoftvertermék jellemző: lásd minőségi jellemző. → *software product characteristics*

szoftverteszt incidens: lásd incidens → *software test incident*

szoftverteszt incidens jelentés: lásd incidensjelentés → *software test incident report*

Szoftver Üzemzavar és Hatáselemzés: lásd Üzemzavar és Hatáselemzés. → *Software Failure Mode and Effect Analysis (SFMEA)*

Szoftver Üzemzavar, Hatás és Kritikusság Elemzés: lásd: Szoftver Üzemzavar, Hatás és Kritikusság Elemzés. → *Software Failure Mode Effect, and Criticality Analysis (SFMECA)*

sztochasztikus teszt: a csimpánz-teszthez hasonló, rendszerint automatikusan végrehajtott nagyszámú, véletlenszerű feketedoboz tesztek sorozata. Az egyes tesztek külön-külön általában sikeresek, a cél annak vizsgálata, hogy nagyszámú ilyen teszt esetén hogyan viselkedik a rendszer. → *stochastic testing*

tanulás (IDEAL): az IDEAL modell egyik fázisa, amely során a tapasztatokból tanulunk, illetve az új folyamatok és technológiák jövőbeli alkalmazásának képességét fejlesztjük. A következő tevékenységeket foglalja magában: analízis, validálás, javaslat jövőbeli tevékenységekre. Lásd még: IDEAL → *learning (IDEAL)*

tanulhatóság: az alkalmazás azon tulajdonsága, amely lehetővé teszi, hogy a felhasználó könnyen megismerje és megtanulja az alkalmazás használatát. [ISO 9126] Lásd még: használhatóság → *learnability*

tanúsítvány: annak igazolása, hogy az adott szoftver, illetve személy megfelel a követelményeknek (pl. egy vizsga letételével) → *certification*

tapasztalat-alapú technika: lásd tapasztalat-alapú tesztervezési módszer → *experienced-based technique*

tapasztalat-alapú tesztervezési technika: olyan tesztervezési módszer, amely során a tesztelők tapasztalata, tudása és megérzései alapján származtatunk, illetve választunk teszteseteket → *experienced-based test design technique*

tartalom-alapú modell: a jó mérnöki gyakorlatok – pl. tesztelési gyakorlatok – részletes leírását tartalmazó folyamatmodell → *content-based model*

tároló: lásd erőforrás kihasználtság. → *storage*

tároló-használati teszt: lásd erőforrás kihasználtsági teszt. → *storage testing*

technikai felülvizsgálat: egyenrangú résztvevők csoportos megbeszélése, akik a technikai megközelítés tekintetében törekednek a közös álláspont kialakítására [Gilb és Graham, IEEE 1028] Lásd egyenrangú felülvizsgálat. → *technical review*

telepítési útmutató: egy valamilyen médiumon fellelhető utasításhalmaz, amely a telepítő személynek útmutatást nyújt a telepítés folyamatán. Ez lehet egy kézikönyv, lépésről lépésre haladó eljárásleírás, telepítő varázsló, vagy más hasonló folyamat leírás → *installation guide*

telepíthetőség: annak vizsgálata, hogy a szoftver egy adott környezetben mennyire telepíthető [ISO 9126] → *installability*

telepíthetőségi teszt: a szoftver telepíthetőségének vizsgálatát végző teszt. Lásd még:

hordozhatósági teszt → *installability testing*

telepítési varázsló: egy adott médiumon található szoftver, amely a telepítő személyt végigvezeti a telepítés folyamatán. Általában futtatja a telepítési folyamatot, a telepítés eredményéről visszajelzést ad, illetve opciókat kínál fel → *installation wizard*

Teljeskörű Minőségirányítás: a teljes szervezetre kiható irányítási megközelítés, amely a minőségre koncentrálna, minden dolgozó részvételére épít, a hosszútávú siker feltételének a megrendelői elégedettség tekinti, és a szervezet minden tagja érezheti az előnyeit. A Teljeskörű Minőségmenedzsment tervezési, szervezési, utasítási, irányítási és minőségbiztosítási tevékenységekből áll [ISO 8402]. → *Total Quality Management (TQM)*

teljes teszt: lásd kimerítő teszt → *complete testing*

teljesítési feltétel: lásd kilépési feltétel → *completion criteria*

teljesítmény: egy komponens vagy rendszer azon tulajdonsága, hogy bizonyos funkciókat milyen hatékonyan hajt végre (tranzakciók ideje, száma, válaszidő, stb). [After IEEE 610] Lásd még hatékonyság. → *performance*

teljesítmény mutató: magas szintű mutató, amely a fejlesztés eredményességét és/vagy hatékonyságát mutatja, például szoftverfejlesztésnél a kritikus úton történő csúszás. [CMMI] → *performance indicator*

teljesítmény profilozás: profilozás során naplózott adatok összeállítása teljesítmény, terheléses és/vagy stressz teszthez. A naplózott adatoknak az elvárt vagy a komponens illetve rendszer működési profiljának megfelelő aktuális értékeket kell tükröznie, azaz az elvárt terhelést. Lásd még terhelési profil, működési profil. → *performance profiling*

teljesítményteszt: tesztfolyamat, mellyel a szoftvertermék teljesítményét lehet meghatározni. Lásd még hatékonyság teszt. → *performance testing*

teljesítményteszt eszköz: teljesítményteszthez alkalmazott eszköz, melynek két különböző képessége van: terhelést generál és méri a tranzakciók számát. Az eszköz terhelést generál több felhasználó szimulációjával, illetve nagy mennyiségű bemenő adatot elküldésével. A futás során kiválasztott tranzakciók válaszidejét mérik és naplózzák, amelyből általában a terhelés és a válaszidő kapcsolatát leíró napló file-okon és ábrákon alapuló jelentések készülnek → *performance testing tool*

terheléses teszt: a teljesítményteszt azon típusa, amely a komponens vagy rendszer viselkedését vizsgálja növekedő terhelés alatt (például a felhasználók számának, vagy kérések számának növelésekor). A teszt célja, hogy kiderüljön, a komponens vagy rendszer hogyan reagál a magas terhelésre (lefagy, nő a válaszidő, stb.). Lásd még teljesítményteszt, stressz teszt. → *load testing*

terheléses teszt eszköz: lásd teljesítményteszt eszköz → *load testing tool*

terhelési profil: a tesztelendő alkalmazás egy olyan specifikációja, amely az alkalmazás életszerű használata alatt történő műveleteket írja le. A terhelési profil egy meghatározott számú virtuális felhasználóból áll, amelyek adott időben bizonyos előre definiált műveleteket hajtanak végre a működése profilnak megfelelően. Lásd

még működési profil → *load profile*

termékkockázat: a teszt tárgyához (magához a termékhez) közvetlenül kapcsolódó kockázat. Lásd még kockázat. → *product risk*

termék átvételi teszt: lásd működési átvételi teszt. → *production acceptance testing*

termék-alapú minőség: a minőség egy formája, amely során a minőséget jól meghatározott minőségi attribútumok alapján határozzák meg. Ezen attribútumokat objektíven és mennyiségileg mérhetjük. Az ugyanolyan típusú termékek minősége közötti különbségek a megvalósított minőségi attribútumokra vezethető vissza. → *product-based quality*

terv-alapú teszt: olyan tesztmódszer, amely során a teszteseteket a szoftver tervére, illetve architektúrájára alapozva készítjük el (pl. a szoftver interface-einek tesztje) → *design-based testing*

teszt: egy vagy több teszteset halmaza. [IEEE 829] → *test*

tesztadat: olyan adat, amely a teszt előtt is létezik (például egy adatbázisban) és amely kölcsönhatásban van a tesztelés alatt álló rendszerrel, vagy a rendszerkomponenssel. → *test data*

tesztadat előkészítő eszköz: olyan teszteszköz, amellyel a teszthez adatot állíthatunk elő egy már meglévő adatbázisból. Az eszköz alkalmas lehet egyéb adatok generálására, adatok manipulálására, valamint szerkesztésére is. → *test data preparation tool*

teszt alapkörnyezet: a teszt végrehajtásához szükséges, a meghajtókat és csomópontokat tartalmazó tesztkörnyezet → *test harness*

teszt automatizálás: valamilyen szoftver használata különböző teszttevékenységek támogatására, mint pl. tesztmenedzsment, műszaki teszttervezés, tesztek végrehajtása, teszteredmények vizsgálata → *test automation*

tesztbázis: az összes olyan dokumentum, amelyből a komponensekre vagy rendszerekre vonatkozó követelmények származnak. Ezek azok a dokumentumok, amelyeken a tesztesetek alapulnak. Ha egy ilyen dokumentumot csak formális változáskezelési folyamat során módosíthatnak, a tesztbázist ún. fagyasztott tesztbázisnak nevezik. [TMap alapján] → *test basis*

teszt bemenet: a külső forrásból érkezett adat a teszt végrehajtása során. A külső forrás lehet hardver, szoftver vagy emberi forrás is. → *test input*

tesztcél: a műszaki teszttervezés, illetve tesztvégrehajtás célja → *test objective*

tesztciklus: a tesztfolyamat végrehajtása a teszt tárgyának egy konkrét verziójára → *test cycle*

tesztelem kiadási jelentés: lásd. kiadási megjegyzés → *test item transmittal report*

tesztelem: a tesztelendő egység. Rendszerint a teszt tárgyához több tesztelelem tartozik. → *test item*

tesztbecslés: a teszteredmény számítással történő becslése (pl. ráfordított idő,

befejezés dátuma, költségek, tesztesetek száma), amely akkor is használható, ha a tesztadatok nem állnak teljes mértékben rendelkezésre, vagy bizonytalanok. → *test estimation*

tesztfolyamat fejlesztési kiáltvány: az agilis kiáltvány mintájára kiadott kijelentések, amelyek a tesztfolyamat fejlesztésének értékét határozzák meg. Ezen értékek:

- rugalmasság a részletes folyamatok helyett
- bevált gyakorlat a sablonok helyett
- megvalósítás-irányultság folyamat-irányultság helyett
- egyenrangú felülvizsgálat a minőségbiztosítás helyett
- üzlet-vezéreltség a modell-vezéreltség helyett

→ *test process improvement manifesto*

tesztkimenet: lásd eredmény → *test outcome*

tesztlezárás: a tesztlezárási fázisban gyűjtjük össze a teszt során előállított adatokat, hogy a tesztverből, számokból, tényekből és egyéb tapasztalatokból összegyűjtött adatokat konszolidáljuk. A tesztlezárási fázisban véglegesítjük és archiváljuk a tesztvert, értékeljük ki az eredményeket és készítjük elő az összefoglaló tesztjelentést. Lásd még tesztfolyamat. → *test closure*

tesztirányítás: egy olyan tesztmenedzseri feladat, amelynek során javító intézkedéseket fejlesztünk és alkalmazunk annak érdekében, hogy a tesztprojektet visszatereljük a tervezett pályára, amennyiben a teszt felügyelete során a tervektől való eltérést tapasztalunk → *test control*

teszt megismételhetősége: a teszt egy attribútuma, amely azt jelzi, hogy a teszt többszöri végrehajtása során mindig ugyanazok az eredmények keletkeznek-e → *test reproduceability*

teszt megvalósítása: a teszteljárások fejlesztésének és priorizálásának folyamata, tesztadatok készítése és - opcionálisan - automata tesztszkriptek, valamint a tesztámogató szoftverkörnyezetek kialakítása → *test implementation*

teszt naplózása: a tesztnaplóba történő információ rögzítés folyamata → *test logging, test recording*

teszt összehasonlítás: a rendszeren illetve rendszerkomponensen végzett elvárt és aktuális teszteredményt összehasonlító eljárás. Az összehasonlítás elvégezhető a teszt futtatása közben (dinamikus összehasonlítás) valamint a teszt futtatása után is → *test comparison*

teszt összehasonlító eszköz: egy teszteszköz, amellyel a teszt elvárt és aktuális eredményeit automatikusan össze lehet hasonlítani. → *test comparator*

teszt tárgy: a teszt tárgyát képező rendszer vagy rendszerkomponens. Lásd még: tesztelelem → *test object*

teszttervezés: tesztterv (test plan) készítése vagy módosítása → *test planning*

tesztvázlat: a tesztcélok listája valamint ötletek a teszt végrehajtására. A tesztvázlatok a felderítő tesztben használatosak. Lásd felderítő teszt → *test charter, charter*

tesztvezető: lásd tesztmenedzser → *test leader*

tesztvezérelt fejlesztés: szoftverfejlesztési módszertan, amelyben a teszteseteket azelőtt készítik el (és többnyire automatizálják), mielőtt a szoftver fejlesztési folyamata befejeződne, és le lehetne futtatni a teszteseteket → *test driven development*

tesztelés: az összes szoftverfejlesztési életciklushoz kapcsolódó akár statikus, akár dinamikus folyamat, amely kapcsolatban áll a szoftvertermékek tervezésével, elkészítésével és kiértékelésével, hogy megállapítsa, hogy a szoftvertermék teljesíti-e a meghatározott követelményeket, megfelel-e a célnak. A tesztelés felelős a szoftvertermékkel kapcsolatos hibák megtalálásáért. → *testing*

teszteljárás specifikáció: a teszt futtatásának tevékenységsorozatát rögzítő dokumentum. Teszt szkript illetve manuális teszt szkript néven is ismert. → *test procedure specification*

teszteljárás: lásd teszteljárás specifikáció → *test procedure*

tesztfeltétel: a rendszer egy olyan eleme, vagy eseménye, amelyet egy tesztesettel ellenőrizni lehet, például funkció, tranzakció, jellemző, minőségi attribútum vagy strukturális elem. → *test condition, test situation*

tesztelési irányelvek: magas szintű dokumentum, amely a szervezet elveit, megközelítésmódját, valamint céljait mutatja be a tesztelésre vonatkozóan. → *test policy*

tesztelési átadandó: bármilyen teszt (munka)termék, amelyet a teszt (munka)termék készítőjén kívüli személy részére át kell adni → *test deliverable*

tesztfolyamat javítása (TPI): elsősorban rendszerteszt és felhasználói teszt folyamatainak javítására készített keretrendszer, amely a hatékony tesztfolyamat kulcselemeit tartalmazza. → *Test Process Improvement (TPI)*

teszt infrastruktúra: a teszt elvégzéséhez szükséges szervezeti tényezők, beleértve a tesztkörnyezeteket, teszteszközöket, irodai környezetet és eljárásokat. → *test infrastructure*

tesztver: a tesztfolyamat során keletkezett különböző termékek, például dokumentáció, programkód, inputok, várt eredmények, eljárások, fájlok, adatbázisok, környezetek illetve bármilyen egyéb szoftver. [Fewster és Graham után] → *testware*

tesztelési megközelítés: a tesztstratégia megvalósítása egy konkrét projektre. Jellemzően a projekt céljain és a kockázatelemzésen alapuló döntéseket, a tesztfolyamatok kiindulópontjait, az alkalmazandó műszaki teszttervezési technikákat, belépési és kilépési kritériumokat valamint a teszt fajtáit tartalmazza. → *test approach*

tesztnapló: a tesztvégrehajtáshoz kapcsolódó részletek időrendi rögzítése → *test log, test record*

tesztstratégia: magas szintű dokumentum, amely a végrehajtandó tesztszinteket írja le,

valamint azok részleteit tartalmazza a szervezetre vagy a programra (egy vagy több projektre) vonatkozóan. → *test strategy*

tesztszakasz: a teszt végrehajtása egy megszakítás nélküli idő intervallumban. A felderítő teszt során minden tesztszakasz az egyes fejezetekre összpontosít, de a tesztelők egyéb tevékenységekkel, illetve új lehetőségekkel is foglalkozhatnak. Ebben az esetben a tesztelő a teszteseteket menet közben készíti, hajtja végre valamint dokumentálja azok eredményét. → *test session*

tesztszint: közös szervezés és menedzsment alatt álló teszttevékenységek csoportja. A tesztszint felelősségi körökhöz kapcsolódik a projekten belül. Tesztszintekre példa a komponens-teszt, integrációs teszt, rendszerteszt és felhasználó teszt. [TMap alapján] → *test level, test stage*

teszttechnika: lásd teszttervezési technika → *test technique*

teszt ütemezése: teszttevékenységek, feladatok és események listája, meghatározva azok tervezett kezdési és befejezési időpontját, valamint függőségi viszonyait. → *test schedule*

tesztelési végfeltétel: lásd kilépési feltétel → *test completion criteria*

tesztelhető követelmények: a követelmények olyan szintű leírása, amely alapján a teszt tervezhető és tesztesetek készíthetők, valamint a teszt végrehajtása során meghatározható, hogy a követelmények teljesültek-e. → *testable requirements*

tesztelhetőség: a szoftvertermék azon képessége, amely lehetővé teszi a tesztelést az esetleges módosítások során. → *testability*

tesztelhetőségi felülvizsgálat: a tesztbázis részletes vizsgálata, amelynek során azt vizsgálják, hogy a tesztbázist képező dokumentumok megfelelő minőségűek-e a tesztfolyamat végrehajtásához. → *testability review*

tesztelhetőségi teszt: a smoke tesztek egy speciális fajtája, amely azt vizsgálja, hogy a rendszert érdemes-e részletesen tovább tesztelni. Tipikusan a teszt végrehajtási fázisának kezdetén használják. Lásd még: smoke teszt → *intake test*

tesztelő: egy képzett szakértő, aki a rendszer vagy rendszerkomponens tesztelésében vesz részt. → *tester*

teszteredmény: lásd eredmény → *test result*

teszt-érettségi modell (TMM): a Capability Maturity Model-hez (CMM) kapcsolódó, ötszintes keretrendszer a tesztfolyamatok javítására, amely a hatékony teszt folyamat kulcselemeit írja le. → *Test Maturity Model (TMM)*

tesztfejlesztési terv: mind a szervezet tesztfolyamatainak, mind a hozzá tartozó eszközök aktuális erősségeinek és gyengeségeinek alapos megismerésén alapuló tesztfejlesztési célok elérését szolgáló terv → *test improvement plan*

Tesztfolyamat Csoport: (Teszt) specialisták csapata, akik lehetővé teszik a szervezet tesztfolyamatainak meghatározását, fenntartását és fejlesztését [CMMI], → *Test Process Group (TPG)*

tesztfolyamat fejlesztő: a tesztfelkészítési terven alapuló tesztfolyamat fejlesztését végrehajtó személy → *test process improver*

tesztincidens: lásd incidens → *test incident*

tesztincidens jelentés: lásd incidensjelentés → *test incident report*

teszteset: bemeneti értékek, végrehajtási előfeltételek, várt eredmények és végrehajtási utófeltételek halmaza, amelyeket egy konkrét célért vagy a tesztért fejlesztettek (például egy program forgatókönyv végrehajtása, vagy egy követelménynek való megfelelés). [IEEE 610 alapján] → *test case*

teszteset specifikáció: egy tesztesemre vonatkozó, a teszteseteket meghatározó dokumentáció (cél, bemenetek, teszttevékenységek, várt eredmények, végrehajtás előfeltételei) [IEEE 829 alapján]. → *test case specification*

teszteset tervezési technika: lásd műszaki tesztervezési technika → *test case design technique*

teszteszköz: egy vagy több teszttevékenységet támogató szoftvertermék. Ilyen tevékenység lehet a tervezés és irányítás, specifikáció, bemeneti fájlok és adatok előállítása, teszt végrehajtás és tesztesetvizsgálat. Lásd még CAST → *test tool*

tesztfázis: teszttevékenységek egy különálló halmaza a projekt egy együtt kezelhető fázisába összegyűjtve. Ilyen például egy tesztszinthez tartozó tesztvégrehajtási tevékenységek halmaza [Gerrard alapján] → *test phase*

tesztfázis terv: olyan tesztervezési terv (test plan), amely csupán egy tesztfázist fed le. Lásd még tesztervezési terv (test plan). → *phase test plan*

tesztfelügyelet: a tesztprojekt státuszát rendszeresen vizsgáló, a teszttevékenységet kezelő tesztmenedzser feladat. A tesztfelügyelet részét képezi az elvárt eredményeket az aktuális eredményekkel összehasonlító jelentés készítése. Lásd tesztmenedzser → *test monitoring*

tesztfolyamat: az alapvető tesztfolyamat a következő tevékenységeket öleli fel: a műszaki tesztervezést, a tesztirányítást, a tesztesetvizsgálatot és tervezést, a teszt megvalósítását és végrehajtását, a kilépési feltételek kiértékelését, valamint a teszt lezárási tevékenységeket. → *test process*

teszt-forgatókönyv: lásd teszteljárás specifikáció → *test scenario*

tesztfuttatás: a teszt tárgyának egy meghatározott verzióján a teszt elvégzése → *test run*

tesztfutási napló: lásd tesztnapló → *test run log*

tesztgenerátor: lásd tesztadat előkészítő eszköz → *test generator*

tesztjelentés: lásd összefoglaló tesztjelentés → *test report*

tesztkészlet: rendszerre vagy rendszerkomponensekre készített tesztesetek halmaza, amelyben gyakran az egyik teszt utófeltétele egyben a másik teszt előfeltétele → *test suite*

teszt-kiértékelési jelentés: a tesztfolyamat végén készített dokumentáció, amely összegzi a összes teszttevékenységet és eredményt. Ezen kívül tartalmazza a tesztfolyamat kiértékelését és a teszt során szerzett tapasztalatokat. → *test evaluation report*

tesztkörnyezet: a teszt támogató eszközök együttese, beleértve minden hardver és szoftver eszközt (IEEE 610). → *test bed, test environment, test rig*

teszt lefedettség: lásd lefedettség → *test coverage*

teszt meghajtó: lásd meghajtó → *test driver*

tesztmenedzser: a teszttevékenységekért, erőforrások menedzseléséért valamint a teszt tárgyának értékeléséért felelős személy. A tesztmenedzser irányítja, adminisztrálja, tervezi és menedzseli a teszt tárgyának vizsgálatát → *test manager*

tesztmenedzsment: a teszttevékenységek tervezése, becslése, monitorozása és irányítása, amelyet általában a tesztmenedzser végez. → *test management*

tesztmenedzsment eszköz: olyan szoftver eszköz, amely támogatja a teszt menedzsmentet és irányítja a tesztfolyamat egy részét. Gyakran rendelkezik olyan funkciókkal, mint a tesztkörnyezet menedzsmentje, tesztek ütemezése, eredmények naplózása, folyamatkövetés, hibakezelés és teszteredmények jelentése. → *test management tool*

teszt-orákulum: a várt és az aktuális eredmények összehasonlítását támogató forrás. Teszt-orákulum lehet egy másik, benchmark-ként funkcionáló rendszer; lehet felhasználó kézikönyv, vagy akár egy személy speciális tudása is. A program forráskódja viszont nem lehet teszt-orákulum [Adrion alapján] → *test oracle*

tesztpont elemzés (TPA): a funkciópont elemzés alapján készített, képletéken alapuló tesztbecslési módszer → *Test Point Analysis (TPA)*

teszt specifikáció: dokumentum, amely tartalmazza a műszaki tesztterv (test design) specifikációt, a teszt eset specifikációt és/vagy az eljárás specifikációt. → *test specification*

teszt specifikációs technika: lásd műszaki teszttervezési technika → *test specification technique*

teszt státusz jelentés: rendszeres időközönként készített dokumentum, amely összefoglalja az addigi teszttevékenységeket és eredményeket, valamint összehasonlítja az eredeti tervvel. Ezen kívül a kockázatokat és alternatívákat is összefoglalja a menedzsment számára, hogy támogassa a szükséges döntések meghozatalát. → *test progress report*

tesztszint-tesztterv: olyan tesztterv, amely általában egy adott (alacsony-, vagy magas-) szintű tervezést foglal magába. Lásd még: tesztterv (test plan) → *level test plan*

tesztszkript: legtöbbször teszt eljárás specifikációra használt kifejezés, elsősorban automatizált teszt esetén. → *test script*

teszt támogató szoftver környezet: a teszt végrehajtásához szükséges, a meghajtókat és csonkokat tartalmazó tesztkörnyezet → *test harness*

teszt-teljesítmény mutató: a tesztfolyamat hatékonyságát mérő és irányító magasszintű mutató, pl. hiba megtalálási arány (Defect Detection Percentage, DDP)
→ *test performance indicator*

tesztterv (test plan): a teszt hatáskörét, megközelítését, erőforrásait valamint a tevékenységek tervezett ütemezését tartalmazó dokumentum. Ezen kívül meghatározza a tesztelemekeket, a tesztelendő funkciókat, feladatokat, a tesztet végrehajtó személyek függetlenségét, a tesztkörnyezetet, a műszaki teszttervezési technikákat, a belépési és kilépési feltételeket, valamint kockázatokat. A teszttervezési folyamat meghatározó dokumentuma (IEEE 829 alapján) → *test plan*

teszt típus: egy meghatározott tesztobjektumon végrehajtott, rendszerkomponensre vagy rendszerre fókuszáló teszttevékenységek csoportja. Példa teszt típusokra: funkcionális teszt, használhatósági teszt, regressziós teszt, stb. Egy teszt típus több tesztszinten is végrehajtható. → *test type*

tesztvégrehajtás: a rendszeren vagy annak komponensén végzett tesztfolyamat, amelyből megkapjuk az aktuális eredményeket. → *test execution*

tesztvégrehajtási eszköz: olyan teszteszköz, amelynek segítségével másik szoftvert futtathatunk egy automatizált szkript (pl. felvétel/lejátszás) segítségével. → *test execution tool*

tesztvégrehajtási fázis: a szoftverfejlesztési életciklus azon szakasza, amelyben a szoftverkomponenseket tesztelik, és a szoftvertermék követelményeknek való megfelelését vizsgálják [IEEE 610] → *test execution phase*

tesztvégrehajtási technika: az a módszer, amelyet a teszt végrehajtására használunk, legyen az automatizált vagy manuális teszt → *test execution technique*

tesztvégrehajtás ütemezése: a teszteljárások végrehajtásának ütemezése. A teszteljárások egymásutániságuknak megfelelően, sorrendbe rendezve jelennek meg → *test execution schedule*

téves hiba eredmény: olyan teszteredmény, amely hibát jelez, bár az adott hiba valójában nem létezik a termékben → *false-fail result, false-positive result*

téves siker eredmény: olyan teszteredmény, amely nem találja meg az adott, a termékben meglévő hibát → *false-pass result, false-negative result*

többszörös feltétel: két, vagy több egyszerű feltétel, amelyet egy logikai operator kapcsol össze (AND, OR or XOR), e.g. 'A>B AND C>1000'. → *compound condition*

többszörös feltétel lefedettség: a teszt által meghívott utasításokban szereplő egyes feltétel eredmények összes kombinációinak lefedettségi aránya. 100% kombinációs feltétel lefedettség 100%-os feltétel meghatározási lefedettséget jelent. → *branch condition combination coverage, multiple condition coverage*

többszörös feltétel teszt: fehérdoboz teszttervezési technika, amely során olyan teszteseteket készítünk, amelyek egy összetett predikátumon belül minél több feltétel eredmény kombinációt lefednek → *branch condition combination testing, multiple condition testing*

tranzakció analízis: emberek, illetve emberi elméken belüli közötti tranzakciók analízise; a tranzakciót az inger és a rá adott válasz együtteseként értelmezzük. Tranzakciók történhetnek emberek között, mint ahogy egy adott ember elméjében is az én-állapotai, azaz a személyiséget meghatározó szegmensei között. → *transactional analysis*

transzcendencia-alapú minőség: a minőség azon formája, amit nem lehet egyértelműen meghatározni, mégis tudjuk, ha látjuk, vagy érzékeljük a hiányát, ha az nem megfelelő. A minőség az egyén, vagy egyének csoportjának érzékelésén és szubjektív érzésein múlik, [Garvin szerint]. Lásd még: gyártás-alapú minőség, termék-alapú minőség, felhasználó-alapú minőség, érték-alapú minőség. → *transcendent-based quality*

utasítás: a programozási nyelvek egy entitása, ami tipikusan a futtatás legkisebb oszthatatlan egysége. → *statement*

utasítás lefedettség: a tesztkészlet által kipróbált futtatható utasítások százaléka. → *statement coverage*

utasítás szintű teszt: egy fehérdoboz teszt tervezési technika, amely során úgy tervezzük meg a teszteseteket, hogy utasításokat hajtsanak végre. → *statement testing*

utófeltétel: környezeti vagy állapotbeli feltételek, amelyeket teljesíteni kell, miután egy komponensen vagy rendszeren tesztet vagy teszteljárást futtatunk. → *postcondition*

újrakezdési feltételek: azon teszttevékenység, melyeket meg kell ismételni, amikor egy valamilyen okból felfüggesztett/félbeszakított/elhalasztott tesztet újrakezdünk. [IEEE 829] → *resumption criteria*

újratesztelés: olyan teszt, amikor azokat a teszteseteket futtatjuk, amelyek a legutóbbi tesztfuttatásnál elbuktak. Célja a hibajavítás sikerességének ellenőrzésére. → *re-testing*

útvonal: események egy sora, például egy komponens vagy rendszer kezdő és végpontja közötti végrehajtható utasítások → *path*

útvonal lefedettség: egy tesztkészlet által bejárt utak aránya az összeshez képest. 100%-os útvonal lefedettség 100%-os LCSAJ lefedettséget jelent → *path coverage*

útvonal kikényszerítés: olyan bemeneti értékek kiválasztása, amely egy adott út végrehajtásához vezet → *path sensitizing*

útvonal teszt: fehérdoboz teszttervezési technika, amely során a teszteseteket úgy tervezzük, hogy egy-egy végrehajtási utat járjanak be → *path testing*

üzemeltethetőségi teszt: lásd karbantarthatósági teszt. → *serviceability testing, maintainability testing*

üzemzavar: a meghibásodás fizikai, vagy funkcionális megjelenése. Például a rendszer hibamegjelenése lehet a műveletek lelassulása, helytelen kimentek megjelenése, vagy akár a végrehajtás teljes leállása → *failure mode*

Üzemzavar és Hatáselemzés (FMEA): a meghibásodások lehetséges módjainak

kockázat-meghatározásán és elemzésén, valamint az előfordulásuk megakadályozásán alapuló megközelítés. Lásd még: Üzemzavar, Hatás és Kritikusság Elemzés → *Failure Mode and Effect Analysis (FMEA)*

Üzemzavar, Hatás és Kritikusság Elemzés (FMECA): az Üzemzavar és Hatás Elemzés kibővítése a kritikus elemzéssel, amely a meghibásodások valószínűségét és a következményei súlyosságát ábrázolja grafikusán. A grafikon láthatóvá teszi azon meghibásodásokat, amelyek viszonylag magas valószínűséggel és súlyos következménnyel bírnak, és lehetővé teszi, hogy a megelőző tevékenységet oda koncentráljuk, ahol erre a legnagyobb szükség van. Lásd még: Üzemzavar és Hatás Elemzés → *Failure Mode, Effect and Criticality Analysis (FMECA)*

üzleti folyamat alapú teszt Olyan műszaki tesztervezési módszer, amely során az üzleti folyamatok leírásai, illetve ismerete alapján tervezünk teszteseteket → *business process-based testing*

validáció: annak vizsgálata és konfirmálása, hogy a szoftver tervezett felhasználási céljának megfelelő követelmények teljesülnek-e [ISO 9000] → *validation*

változás irányítás: lásd: konfiguráció irányítás → *change control*

változás irányító testület: azon emberek csoportja, akik felelősek a konfigurációs elemekben tervezett változtatások kiértékeléséért és jóváhagyásáért, vagy elutasításáért, továbbá a tervezett változtatások megvalósításáért → *change control board (CCB)*

változás menedzsment: (1) struktúrált megközelítés az egyének, csapatok, szervezetek jelen állapotból a kívánt jövőbeli állapotba történő átmenetének kezelésére (2) a termékben, vagy szolgáltatásban történő változások, vagy tervezett változások irányított végrehajtása. Lásd még konfiguráció menedzsment → *change management*

változó: a számítógép által tárolt elem, amelyet a szoftver a nevére hivatkozva ér el. → *variable*

változtathatóság: a szoftvertermék azon képessége, hogy mennyire képes adott módosítások megvalósítására → *changeability*

verifikáció: az adott követelmények teljesülésének vizsgálata és konfirmálása [ISO 9000] → *verification*

verziókövetés: lásd konfiguráció irányítás → *version control*

vezérlési folyamat: az események (útvonalak) sorrendje a végrehajtás során → *control flow*

vezető értékelő: az értékelést vezető személy. Bizonyos esetekben, például CMMi, vagy TMMi esetén, ha formális értékelést hajtunk végre, a vezető értékelőt formálisan képezni és akkreditálni kell. → *lead assessor*

vezérlési folyamat elemzés: statikus elemzés, amely a végrehajtás során fellépő események (útvonalak) sorrendjének megjelenítésével foglalkozik. A vezérlési folyamat analízis kiértékeli a vezérlési folyamat strukturák integritását, olyan vezérlési folyamat

rendellenességeket keres, mint pl. a zárt hurkok, vagy a logikailag elérhetetlen folyamatlépések. → *control flow analysis*

vezérlési folyam gráf: a végrehajtás során fellépő összes esemény (útvonal) összes lehetséges sorrendjének absztrakt megjelenítése → *control flow graph*

vezérlési út: lásd útvonal → *control flow path*

végrehajtás utáni összehasonlítás: az elvárt és aktuális eredmények összehasonlítása a szoftver futása után. → *post-execution comparison*

véletlenszerű teszt: egy olyan feketedoboz tesztervezési technika, ahol a teszteseteket a működési profilhoz illeszkedően – esetleg pszeudo-véletlen generáló algoritmus használatával - választják ki. Ezt a technikát olyan nem-funkcionális tulajdonságok tesztjénél is használják, mint a megbízhatóság és a teljesítmény. → *random testing*

véges állapotú automata: olyan számítási modell, amely véges számú állapotot és ezen állapotok közötti átmenetet tartalmaz, esetleg kísérő műveletekkel együtt → *finite state machine*

véges állapotú teszt: lásd: állapotátmenet teszt → *finite state machine*

végrehajtható utasítás: olyan utasítás, amelyet a gépi kódra való lefordítás után a szoftver futás közben műveletként végrehajt, és rendszerint valamilyen adatműveletet hajt végre → *executable statement*

végrehajtható útvonal: olyan útvonal, amely végrehajtásához a bemeneti értékek és az előfeltételek rendelkezésre állnak → *feasible path*

végrehajtott: egy programelemre – utasításra, döntésre, vagy más strukturális elemre – akkor mondjuk, hogy a teszteset által végrehajtott, ha a bemeneti érték a programelem végrehajtását okozza → *exercised*

viselkedés: a szoftver válasza adott előfeltételekre és bemeneti értékekre → *behavior*

viszonyított értékelőlap: stratégiai teljesítmény menedzsment eszköz, amellyel azt mérhetjük, hogy a vállalat működési tevékenységei összhangban vannak-e a céljaival a vállalat jövőképevel, illetve stratégiájával. Lásd még egyesített áttekintőtábla, értékelőlap → *balanced scorecard*

visszaállíthatóság: egy szoftvertermék azon képessége, hogy külső hiba esetén a hiba által okozott rendszert érintő következményeket, azaz a rendszer teljesítményének meghatározott szintjét és az adatokat helyreállítsa. [ISO 9126] Lásd még megbízhatóság → *recoverability*

visszaállíthatósági teszt: az a tesztfolyamat, amely egy szoftvertermék visszaállíthatóságát vizsgálja. Lásd még megbízhatósági teszt → *recoverability testing, recovery testing*

visszatekintő megbeszélés: egy projekt végén, a projekttagokkal tartott megbeszélés, melyen értékelik a projektet és a tanulságokat egy következő projekt számára. → *retrospective meeting, post-project meeting*

vízszintes nyomonkövethetőség: a követelmények egy adott tesztszinten történő

nyomon követése a különböző tesztdokumentációkon keresztül (pl. teszterv, műszaki teszterv specifikáció, teszteset specifikáció, teszteljárás specifikáció, vagy tesztszkript) → *horizontal traceability*

V-modell: egy keretrendszer a szoftverfejlesztési ciklus részletes leírására, a követelményspecifikációtól kezdve az üzemeltetésig. Az V-modell illusztrálja, hogy a teszttevékenység hogyan illeszthető a szoftverfejlesztési életciklusok bármely fázisába → *V-model*

4. English-Hungarian dictionary / Angol-magyar szótár

A

abstract test case: absztrakt teszteset

acceptance: átvétel

acceptance criteria: átvételi kritérium

acceptance testing: átvételi teszt

accessibility testing: hozzáférhetőségi teszt

accuracy: pontosság

accuracy testing: pontossági teszt

acting (IDEAL): cselekvés (IDEAL)

action word driven testing: akciószó alapú teszt

actual outcome: aktuális eredmény

actual result: aktuális eredmény

ad hoc review: ad hoc felülvizsgálat

ad hoc testing: ad hoc teszt

adaptability: adaptálhatóság

agile testing: agilis tesztelés

agile manifesto: agilis kiáltvány

agile software development: agilis szoftverfejlesztés

algorithm test [TMap]: algoritmus teszt

alpha testing: alfa teszt

analyzability: elemezhetőség

analyzer: elemző

anomaly: rendellenesség

arc testing: elágazás teszt
assessment report: értékelő jelentés
assessor: értékelő
attack: támadás
attractiveness: megjelenés
audit: audit
audit trail: audit nyomvonal
automated testware: automatizált tesztver
availability: rendelkezésre állás

B

back-to-back testing: összehasonlító teszt
balanced scorecard: viszonyított értékelőlap
baseline: baseline
basic block: elemi blokk
basis test set: alap tesztkészlet
bebugging: hiba-elhelyezés
behavior: viselkedés
benchmark test: "benchmark" teszt
bespoke software: egyedi szoftver
best practice: bevált gyakorlat
beta testing: béta teszt
big-bang testing: nagy bumm teszt
black-box technique: feketedoboz technika
black-box testing: feketedoboz teszt
black-box test design technique: feketedoboz teszttervezési technika
blocked test case: blokkolt teszteset
bottom-up testing: alulról-felfelé haladó teszt
boundary value: határérték
boundary value analysis: határérték elemzés
boundary value coverage: határérték lefedettség
boundary value testing: határérték teszt

branch: elágazás
branch condition: elágazási feltétel
branch condition combination coverage: feltétel-kombinációs lefedettség
branch condition combination testing: feltétel-kombinációs teszt
branch condition coverage: elágazási feltétel lefedettség
branch coverage: elágazás lefedettség
branch testing: elágazási teszt
buffer: puffer
buffer overflow: puffer túlcsoordulás
bug: programhiba
bug report: hibajelentés
bug taxonomy: hibaosztályozás
bug tracking tool: hibakövető eszköz
business process-based testing: üzleti folyamat alapú teszt

C

call graph: hívási gráf
Capability Maturity Model (CMM): képesség-érettség modell (CMM)
Capability Maturity Model Integration (CMMI): integrált képességi-érettségi modell (CMMI)
capture/playback tool: felvevő/lejátszó eszköz
capture/replay tool: felvevő/lejátszó eszköz
CASE: számítógéppel támogatott szoftverfejlesztés (CASE)
CAST: számítógéppel támogatott szoftvertesztelés (CAST)
causal analysis: okozati elemzés
cause-effect analysis: ok-okozat elemzés
cause-effect decision table: ok-okozat döntési tábla
cause-effect diagram: ok-okozat diagram
cause-effect graph: ok-okozati gráf
cause-effect graphing: ok-okozat teszt
certification: tanúsítvány
change control: változás irányítás

change control board (CCB): változás irányító testület
change management: változás menedzsment
changeability: változtathatóság
charter: tesztvázlat
checker: felülvizsgáló
checklist-based testing: ellenőrző lista alapú teszt
Chow's coverage metrics: Chow lefedettségi metrikái
clear-box testing: fehérdoboz teszt
classification tree: osztályozási fa
classification tree method: osztályozási fa módszer
co-existence: együttélés
code: kód
code analyzer: kód elemző
code coverage: kód lefedettség
code-based testing: kód alapú teszt
codependent behavior: kodependens viselkedés (kapcsolati függőség)
commercial off-the-shelf software: kereskedelmi dobozos szoftver
comparator: összehasonlító
compatibility testing: kompatibilitási teszt
compiler: fordítóprogram
complete testing: teljes teszt
completion criteria: teljesítési feltétel
complexity: komplexitás
compliance: szabályosság
compliance testing: szabályossági teszt
component: komponens
component integration testing: komponens integrációs teszt
component specification: komponens specifikáció
component testing: komponens teszt
compound condition: többszörös feltétel
concrete test case: konkrét teszteset
concurrency testing: konkurrencia teszt
condition: feltétel

condition combination coverage: módosított kombinációs feltétel lefedettség

condition combinaton testing: módosított kombinációs feltétel teszt

condition coverage: feltétel lefedettség

condition determination coverage: feltétel meghatározási lefedettség

condition determination testing: feltétel meghatározási teszt

condition outcome: feltétel eredménye

condition testing: feltétel teszt

confidence test: bizonyossági teszt

configuration: konfiguráció

configuration auditing: konfiguráció ellenőrzés

configuration control: konfiguráció irányítás

configuration control board (CCB): konfiguráció irányító testület

configuration identification: konfiguráció azonosítás

configuration item: konfigurációs elem

configuration management: konfiguráció menedzsment

configuration management tool: konfiguráció menedzsment eszköz

configuration testing: konfiguráció teszt

confirmation testing: ellenőrző teszt

conformance testing: szabályossági teszt

consistency: konzisztencia

content-based model: tartalom-alapú modell

continuous representation: folytonos reprezentáció

control flow: vezérlési folyamat

control flow analysis: vezérlési folyamat elemzés

control flow graph: vezérlési folyamat gráf

control flow path: vezérlési út

conversion testing: konverziós teszt

corporate dashboard: egyesített áttekintőtábla

cost of quality: minőség költsége

COTS: dobozos szoftver

coverage: lefedettség

coverage analysis: lefedettség elemzés

coverage measurement tool: lefedettség-mérő eszköz

coverage item: lefedettségi elem
coverage tool: lefedettségi eszköz
critical success factor: kritikus sikertényező
Critical Testing Process (CTP): kritikus tesztfolyamat
custom software: egyedi szoftver
cyclomatic complexity: ciklomatikus komplexitás
cyclomatic number: ciklomatikus szám

D

daily build: napi build
dashboard: áttekintőtábla
data definition: értékadás
data driven testing: adatvezérelt teszt
data flow: adatfolyam
data flow analysis: adatfolyam elemzés
data flow coverage: adatfolyam lefedettség
data flow testing: adatfolyam teszt
data integrity testing: adat-integritás teszt
database integrity testing: adatbázis-integritás teszt
dd-path: döntésközi útvonal
dead code: elérhetetlen kód
debugger: hibakereső
debugging: hibakeresés
debugging tool: hibakereső eszköz
decision: döntés
decision condition coverage: döntési feltétel lefedettség
decision condition testing: döntési feltétel teszt
decision coverage: döntési lefedettség
decision outcome: döntési eredmény
decision table: döntési tábla
decision table testing: döntési tábla teszt

decision testing: döntési teszt

defect: programhiba

defect based technique: hiba alapú technika

defect based test design technique: hiba alapú tesztervezési technika

defect density: hibasűrűség

Defect Detection Percentage (DDP): hiba észlelési arány

defect management: hiba-menedzsment

defect management tool: hiba-menedzsment eszköz

defect masking: hibamaszkolás

defect report: hibajelentés

defect taxonomy: hibaosztályozás

defect tracking tool: hibakövető eszköz

definition-use pair: definíció-használat pár

deliverable: átadandó

Deming cycle: Deming ciklus

design-based testing: terv-alapú teszt

desk checking: ellenőrzés papíron

development testing: fejlesztői teszt

deviation: eltérés

deviation report: eltérési jelentés

diagnosing (IDEAL): diagnózis (IDEAL)

dirty testing: negatív teszt

documentation testing: dokumentáció teszt

domain: értéktartomány

driver: meghajtó

dry run: képzeletbeli futtatás

dynamic analysis: dinamikus elemzés

dynamic analysis tool: dinamikus elemző eszköz

dynamic comparison: dinamikus összehasonlítás

dynamic testing: dinamikus teszt

E

efficiency: hatékonyság

efficiency testing: hatékonysági teszt

EFQM (European Foundation for Quality Management) excellence model: EFQM
kiválósági modell

elementary comparison testing: elemi összehasonlító teszt

emotional intelligence: érzelmi intelligencia

emulator: emulátor

entry criteria: belépési feltétel

entry point: belépési pont

equivalence class: ekvivalenciaosztály

equivalence partition: ekvivalencia partíció

equivalence partition coverage: ekvivalencia partíció lefedettség

equivalence partitioning: ekvivalencia partícionálás

error: emberi eredetű hiba

error guessing: hibasejtés

error seeding: hibabeszúrás

error seeding tool: hibaelhelyező eszköz

error tolerance: hibatűrés

establishing (IDEAL): kiépítés (IDEAL)

evaluation: kiértékelés

exception handling: kivételkezelés

executable statement: végrehajtható utasítás

exercised: végrehajtott

exhaustive testing: kimerítő teszt

exit criteria: kilépési feltétel

exit point: kilépési pont

expected outcome: elvárt eredmény

expected result: elvárt eredmény

experienced-based technique: tapasztalat-alapú technika

experience-based test design technique: tapasztalat-alapú tesztervezési technika

exploratory testing: felderítő teszt

extreme programming: extrém programozás

F

fail: bukás

failure: meghibásodás

failure mode: üzemzavar

Failure Mode and Effect Analysis (FMEA): Üzemzavar és Hatás Elemzés (FMEA)

Failure Mode, Effect and Criticality Analysis (FMECA): Üzemzavar, Hatás és Kritikusság Elemzés (FMECA)

failure rate: meghibásodási ráta

false-fail result: téves hiba eredmény

false-pass result: téves siker eredmény

false-positive result: téves-hiba eredmény

false-negative result: téves-siker eredmény

fault: programhiba

fault attack: hibatámadás

fault density: hibasűrűség

Fault Detection Percentage (FDP): hiba megtalálási arány

fault masking: hibamaszkolás

fault seeding: hibabeszúrás

fault seeding tool: hibaeszűrő eszköz

fault tolerance: hibatűrés

Fault Tree Analysis (FTA): hibafa elemzés

feasible path: végrehajtható útvonal

feature: feature

field testing: béta teszt

finite state machine: véges állapotú automata

finite state testing: véges állapot teszt

fishbone diagram: halszálka diagram

formal review: formális felülvizsgálat

frozen test basis: fagyasztott tesztbázis

Function Point Analysis (FPA): funkciópont elemzés

functional integration: funkcionális integráció

functional requirement: funkcionális követelmény

functional test design technique: funkcionális tesztervezési technika

functional testing: funkcionális teszt

functionality: funkcionalitás

functionality testing: funkcionalitási teszt

G

glass box testing: fehérdoboz teszt

Goal Question Metric (GQM): Cél Kérdés Metrika

H

hazard analysis: kockázatelemzés

heuristic evaluation: heurisztikán alapuló kiértékelés

high level test case: magas szintű teszteset

horizontal traceability: vízszintes nyomonkövethetőség

hyperlink: hiperlink

hyperlink test tool: hiperlink teszteszköz

hyperlink tool: hiperlink ellenőrző eszköz

I

IDEAL: IDEAL

impact analysis: hatáselemzés

incident: incidens

incident logging: incidens naplózása

incident management: incidens menedzsment

incident management tool: incidens menedzsment eszköz

incident report: incidensjelentés

incremental development model: inkrementális fejlesztési modell

incremental testing: inkrementális teszt

independence of testing: független tesztelés

indicator: intézkedés előrejelző

infeasible path: bejárhatatlan útvonal

informal review: informális felülvizsgálat
initiating (IDEAL): indítás (IDEAL)
input: bemenet
input domain: bemeneti értéktartomány
input value: bemeneti érték
inspection: inspekción
inspection leader: inspekción vezető
inspector: inspektor
installability: telepíthetőség
installability testing: telepíthetőségi teszt
installation guide: telepítési útmutató
installation wizard: telepítési varázsló
instrumentation: instrumentálás
instrumenter: instrumentáló
intake test: tesztelhetőségi teszt
integration: integráción
integration testing: integrációs teszt
integration testing in the large: rendszer integrációs teszt
integration testing in the small: komponens integrációs teszt
interface testing: interfész teszt
interoperability: együttműködőképesség
interoperability testing: együttműködőképességi teszt
invalid testing: érvénytelen teszt
Ishikawa diagram: Ishikawa diagram
isolation testing: elszigetelt teszt
item transmittal report: kiadási jelentés
iterative development model: iteratív fejlesztési modell

K

key performance indicator: kulcsfontosságú teljesítményjelző
keyword driven testing: akciószó alapú teszt

L

LCSAJ: LCSAJ

LCSAJ coverage: LCSAJ lefedettség

LCSAJ testing: LCSAJ teszt

lead assessor: vezető értékelő

learnability: tanulhatóság

learning (IDEAL): tanulás (IDEAL)

level test plan: tesztszint-tesztterv

link testing: komponens kapcsolati teszt

lifecycle model: életciklus modell

load profile: terhelési profil

load testing: terheléses teszt

load testing tool: terheléses teszt eszköz

logic-coverage testing: programlogika lefedettségi teszt

logic-driven testing: programlogika alapú teszt

logical test case: logikai teszteset

low level test case: alacsony szintű teszteset

M

maintainability: karbantarthatóság

maintainability testing: karbantarthatósági teszt

maintenance: karbantartás

maintenance testing: karbantartási teszt

management review: menedzsment felülvizsgálat

manufacturing-based quality: gyártás-alapú minőség

master test plan: fő tesztterv (test plan)

maturity: érettség

maturity level: érettségi szint

maturity model: érettségi modell

Mean Time To Repair (MTTR): Helyreállási idő

Mean Time Between Failures (MTBF): Meghibásodások Közti Átlagidő

measure: mérőszám

measurement: mérés
measurement scale: mérési skála
memory leak: memóriaszivárgás
metric: metrika
migration testing: migrációs teszt
milestone: mérföldkő
mind-map: gondolattérkép
mistake: emberi eredetű hiba
modelling tool: modellező eszköz
moderator: moderátor
modified condition decision coverage: módosított döntési feltétel lefedettség
modified condition decision testing: módosított döntési feltétel teszt
modified multiple condition coverage: módosított többszörös feltétel lefedettség
modified multiple condition testing: módosított többszörös feltétel teszt
module: modul
module testing: modulteszt
monitor: felügyeleti eszköz
monitoring tool: felügyeleti eszköz
monkey testing: csimpánz-teszt
multiple condition: többszörös feltétel
multiple condition coverage: többszörös feltétel lefedettség
multiple condition testing: többszörös feltétel teszt
mutation analysis: mutációs elemzés
mutation testing: mutációs teszt

N

N-switch coverage: N-lépéses lefedettség
N-switch testing: N-lépéses teszt
negative testing: negatív teszt
non-conformity: nem-szabályosság
non-functional requirement: nem-funkcionális követelmény

non-functional test design technique: nem-funkcionális tesztervezési technika

non-functional testing: nem-funkcionális teszt

O

off-the-shelf software: dobozos szoftver

operability: működtethetőség

operational acceptance testing: működési átvételi teszt

operational environment: működési környezet

operational profile: működési profil

operational profile testing: működési profil teszt

operational testing: működési teszt

oracle: orákulum

orthogonal array: ortogonális mátrix

orthogonal array testing: ortogonális mátrix teszt

outcome: eredmény

output: kimenet

output domain: kimeneti értéktartomány

output value: kimeneti érték

P

pair programming: páros programozás

pair testing: páros teszt

pairwise testing: páronkénti teszt

Pareto analysis: Pareto elemzés

partition testing: partíciós teszt

pass: sikeres teszt

pass/fail criteria: siker/bukás feltétel

path: útvonal

path coverage: útvonal lefedettség

path sensitizing: útvonal kikényszerítés

path testing: útvonal teszt

peer review: egyenrangú felülvizsgálat
performance: teljesítmény
performance indicator: teljesítmény mutató
performance profiling: teljesítmény profilozás
performance testing: teljesítményteszt
performance testing tool: teljesítményteszt eszköz
phase test plan: tesztfázis terv
pointer: mutató
portability: hordozhatóság
portability testing: hordozhatósági teszt
post-execution comparison: végrehajtás utáni összehasonlítás
postcondition: utófeltétel
post-project meeting: visszatekintő megbeszélés
precondition: előfeltétel
predicted outcome: előrejelzett kimenet
pretest: előzetes teszt
priority: prioritás
probe effect: mérési mellékhatás
problem: probléma
problem management: probléma menedzsment
problem report: probléma jelentés
procedure testing: eljárás teszt
process: folyamat
process assessment: folyamatértékelés
rocess cycle test: folyamat ciklus teszt
process improvement: folyamatjavítás
process flow model: folyamatlefutási modell
process model: folyamatmodell
product-based quality: termék-alapú minőség
product risk: termékkockázat
production acceptance testing: termék átvételi teszt
program instrumenter: program-instrumentáló
program testing: programteszt

project: projekt
project retrospective: projekt visszatekintő
project risk: projektkockázat
project test plan: projekt teszterv (test plan)
pseudo-random: pszeudó-véletlen

Q

qualification: minősítés
quality: minőség
quality assurance: minőségbiztosítás
quality attribute: minőségi jellemző
quality characteristic: minőségi jellemző
quality management: minőségmenedzsment

R

random testing: véletlenszerű teszt
Rational Unified Process (RUP): Rational Egységesített Folyamat (Rational Unified Process, RUP)
re-testing: újratesztelés
record/playback tool: felvevő/lejátszó eszköz
recorder: jegyzőkönyv vezető
recoverability: visszaállíthatóság
recoverability testing: visszaállíthatósági teszt
recovery testing: visszaállíthatósági teszt
regression testing: regressziós teszt
regulation testing: szabályozási teszt
release note: kiadási megjegyzés
reliability: megbízhatóság
reliability growth model: megbízhatóság-növekedés modell
reliability testing: megbízhatósági teszt
replaceability: helyettesíthetőség
requirement: követelmény

requirements management tool: követelmény-menedzsment eszköz

requirements phase: követelményfeltárási fázis

requirements-based testing: követelmény alapú teszt

resource utilization: erőforrás kihasználtság

resource utilization testing: erőforráshasználati teszt

result: eredmény

resumption criteria: újrakezdési feltételek

retrospective meeting: visszatekintő megbeszélés

review: felülvizsgálat

review tool: felülvizsgáló eszköz

reviewer: felülvizsgáló

risk: kockázat

risk analysis: kockázatelemzés

risk category: kockázat kategória

risk control: kockázatkezelés

risk identification: kockázat azonosítás

risk level: kockázati szint

risk management: kockázatkezelés

risk mitigation: kockázatcsökkentés

risk type: kockázattípus

risk-based testing: kockázat alapú teszt

robustness: robusztusság

robustness testing: robusztussági teszt

root cause: kiváltó ok

root cause analysis: kiváltó ok elemzés

S

safety: biztonság

safety critical system: biztonság-kritikus rendszer

safety testing: biztonsági teszt

sanity test: alapvető teszt

scalability: skálázhatóság

scalability testing: skálázhatósági teszt

scenario testing: forgatókönyv alapú teszt

scorecard: értékelőlap

scribe: jegyzőkönyv vezető

scripted testing: szkript alapú teszt

scripting language: szkript nyelv

SCRUM: SCRUM

security: biztonság

security testing: biztonsági teszt

security testing tool: biztonsági teszteszköz

security tool: biztonsági eszköz

serviceability testing: üzemeltethetőségi teszt

session-based test management: munkamenet-alapú teszt vezetése

session-based test: munkamenet-alapú teszt

severity: súlyosság

simulation: szimuláció

simulator: szimulátor

site acceptance testing: helyszíni átvételi teszt

smoke test: smoke teszt

software: szoftver

software attack: szoftveres támadás

Software Failure Mode and Effect Analysis (SFMEA): Szoftver Üzemzavar és Hatáselemzés

Software Failure Mode, Effect and Criticality Analysis (SFMECA): Szoftver Üzemzavar, Hatás és Kritikusság Elemzés

Software Fault Tree Analysis (SFTA): szoftverhibafa elemzés

software feature: szoftver feature

software life cycle: szoftver-életciklus

Software Process Improvement (SPI): Szoftverfolyamat Fejlesztés

software product characteristics: szoftvertermék jellemző

software quality: szoftverminőség

software quality characteristic: szoftverminőségi jellemző

software test incident: szoftverteszt incidens

software test incident report: szoftverteszt incidensjelentés

Software Usability Measurement Inventory (SUMI): Szoftver Használhatósági Kérdőív

source statement: forráskód utasítás

specification: specifikáció

specification-based technique: specifikáció alapú technika

specification-based test design technique: specifikáció alapú tesztervezési technika

specification-based testing: specifikáció alapú teszt

specified input: meghatározott bemeneti érték

stability: stabilitás

staged representation: lépcsőzetes felépítés

standard: szabvány

standard software: általános célú szoftvertermék

standards testing: szabványossági teszt

state diagram: állapotdiagram

state table: állapottábla

state transition: állapotátmenet

state transition testing: állapotátmenet teszt

statement: utasítás

statement coverage: utasítás lefedettség

statement testing: utasítás szintű teszt

static analysis: statikus elemzés

static analysis tool: statikus elemző eszköz

static analyzer: statikus elemző eszköz

static code analysis: statikus forráskód elemzés

static code analyzer: statikus forráskód elemző

static testing: statikus teszt

statistical testing: statisztikai teszt

status accounting: állapot rögzítés

stochastic testing: sztochasztikus teszt

storage: tároló

storage testing: tároló-használati teszt

stress testing: stressz teszt

stress testing tool: stressz teszteszköz
structural coverage: struktúra szintű lefedettség
structural test design technique: struktúra szintű tesztervezési technika
structural testing: struktúra alapú teszt
structure-based technique: struktúra alapú technika
structure-based test design technique: struktúra alapú műszaki tesztervezési technika
structure-based testing: struktúra alapú teszt
structured walkthrough: strukturált átvizsgálás
stub: csonk
subpath: al-útvonal
suitability: megfelelőség
suitability testing: megfelelőségi teszt
suspension criteria: felfüggesztési feltételek
syntax testing: szintaxis teszt
system: rendszer
system integration testing: rendszer-integrációs teszt
system of systems: rendszerek rendszere
system testing: rendszerteszt
Systematic Test and Evaluation Process (STEP): Módszeres Teszt és Kiértékelési Folyamat

T

technical review: technikai felülvizsgálat
test: teszt
test approach: tesztelési megközelítés
test automation: teszt-automatizálás
test basis: tesztbázis
test bed: tesztkörnyezet
test case: teszteset
test case design technique: teszteset tervezési technika
test case specification: teszteset specifikáció
test case suite: tesztkészlet
test charter: tesztvázlat

test closure: tesztlezárás

test comparator: tesztösszehasonlító eszköz

test comparison: teszt összehasonlítás

test completion criteria: tesztelési végfeltétel

test condition: tesztfeltétel

test control: tesztirányítás

test coverage: tesztlefedettség

test cycle: tesztciklus

test data: tesztadat

test data preparation tool: tesztadat előkészítő eszköz

test deliverable: tesztelési átadandó

test design: műszaki tesztterv (test design)

test design specification: műszaki tesztterv (test design) specifikáció

test design technique: műszaki teszttervezési technika

test design tool: műszaki teszttervező eszköz

test driven development: tesztvezérelt fejlesztés

test driver: teszt meghajtó

test environment: tesztkörnyezet

test estimation: tesztbecslés

test evaluation report: teszt-kiértékelési jelentés

test execution: tesztvégrehajtás

test execution automation: automatizált tesztvégrehajtás

test execution phase: tesztvégrehajtási fázis

test execution schedule: tesztvégrehajtás ütemezése

test execution technique: tesztvégrehajtási technika

test execution tool: tesztvégrehajtási eszköz

test fail: sikertelen teszt

test generator: tesztgenerátor

test harness: teszt támogató szoftverkörnyezet

test implementation: teszt megvalósítása

test improvement plan: tesztfejlesztési terv

test incident: tesztincidens

test incident report: tesztincidens jelentés

test infrastructure: teszt infrastruktúra

test input: tesztbemenet

test item: tesztelelem

test item transmittal report: tesztelelem kiadási jelentés

test leader: tesztvezető

test level: tesztszint

test log: tesztnapló

test logging: teszt naplózása

test management: tesztmenedzsment

test management tool: tesztmenedzsment eszköz

test manager: tesztmenedzser

Test Maturity Model (TMM): teszt-érettségi modell (TMM)

Test Maturity Model Integrated (TMMi): Integrált Teszt-Érettségi Modell (TMMI)

test monitoring: tesztfelügyelet

test object: teszt tárgya

test objective: teszt cél

test oracle: teszt-orákulum

test outcome: teszt kimenet

test pass: sikeres teszt

test performance indicator: teszt-teljesítmény mutató

test phase: tesztfázis

test plan: tesztterv (test plan)

test planning: teszttervezés

Test Point Analysis (TPA): tesztpont elemzés (TPA)

test policy: tesztelési irányelvek

test procedure: teszteljárás

test procedure specification: teszteljárás specifikáció

test process: tesztfolyamat

Test Process Group (TPG): Tesztfolyamat Csoport

Test Process Improvement (TPI): tesztfolyamat javítása (TPI)

test process improvement manifesto: tesztfolyamat fejlesztési kiáltvány

test process improver: tesztfolyamat fejlesztő

test progress report: teszt státusz jelentés

test record: tesztnapló
test recording: teszt naplózása
test report: tesztjelentés
test reproduceability: teszt megismételhetősége
test requirement: teszt követelmény
test result: teszteredmény
test rig: tesztkörnyezet
test run: tesztfuttatás
test run log: tesztfutási napló
test scenario: teszt-forgatókönyv
test schedule: teszt ütemezése
test script: tesztszkript
test session: tesztszakasz
test set: tesztkészlet
test situation: tesztfeltétel
test specification: tesztspecifikáció
test specification technique: tesztspecifikációs technika
test stage: tesztszint
test strategy: tesztstratégia
test suite: tesztkészlet
test summary report: összefoglaló tesztjelentés
test target: tesztcél
test technique: teszttechnika
test tool: teszteszköz
test type: tesztípus
testability: tesztelhetőség
testability review: tesztelhetőségi felülvizsgálat
testable requirements: tesztelhető követelmények
tester: tesztelő
testing: tesztelés
testware: tesztver
thread testing: progresszív integráció alapú teszt
time behavior: időbeli viselkedés

top-down testing: felülről-lefelé haladó teszt

Total Quality Management (TQM): Teljeskörű Minőségirányítás

traceability: nyomonkövethetőség

transactional analysis: tranzakció analízis

transcendent-based quality: transzcendencia-alapú minőség

U

understandability: érthetőség

unit: egység

unit test framework: egységteszt keretrendszer

unit testing: egységteszt

unreachable code: elérhetetlen kód

usability: használhatóság

usability testing: használhatósági teszt

use case: használati eset

use case testing: használati eset teszt

user acceptance testing: felhasználói átvételi teszt

user-based quality: felhasználó-alapú minőség

user scenario testing: felhasználói forgatókönyv teszt

user test: felhasználói teszt

V

V-model: V-modell

validation: validáció

value-based quality: érték-alapú minőség

variable: változó

verification: verifikáció

version control: verziókövetés

vertical traceability: függőleges nyomonkövethetőség

volume testing: mennyiségi teszt

W

walkthrough: átvizsgálás

white-box techniques: fehérdozoz technikák

white-box test design technique: fehérdozoz teszttervezési technika

white-box testing: fehérdozoz teszt

Wide Band Delphi: széleskörű Delphi eljárás

wild pointer: érvénytelen mutató

Work Breakdown Structure (WBS): Munkaelem Strukturálás