



ITAlian Software Testing Qualifications Board

Glossario Standard dei termini usati nel Software Testing

Versione 2.1 (Maggio, 2010)

DATI IDENTIFICATIVI

CODICE DOCUMENTO	DATA DI EMISSIONE	STATO	
ITASTQB-GLOSS-100501-02	01/05/2010	<input checked="" type="checkbox"/> REDATTA	<input checked="" type="checkbox"/> APPROVATA

REDATTORI	DATA
M. SOGLIANI	01/05/2010
A. COLLINO	01/05/2010

APPROVATORI	DATA
COMITATO SCIENTIFICO	10/05/2010

Associazione ITA-STQB

Sede Legale e Amministrativa: 25125 Brescia - Via Brozzoni 9
C.F. 03265510176 P.I 03121180982
e-mail info@ita-stqb-org

STORIA DELLE MODIFICHE

CHI	DATA	VERSIONE	CONTENUTO
Alessandro Collino, Marco Sogliani	30 Set 2008	01	Non applicabile in quanto si tratta della prima versione del documento
Marco Sogliani	01 Mag 2010	02	Il Glossario è stato allineato alla nuova versione ufficiale emessa da ISTQB

Redattore Glossario in lingua inglese : Erik van Veenendaal (Paesi Bassi)

Autore traduzione in lingua italiana : Alessandro Collino (Italia), Marco Sogliani

Nota di Copyright

Questo documento può essere copiato nella sua interezza, o possono esserne estratte delle parti, solo previa autorizzazione di ITA-STQB e se la fonte viene citata.

INDICE DEI CONTENUTI

1.	Contributori al Glossario Originale in Lingua Inglese.....	5
2.	Storico delle modifiche del Glossario Originale in Lingua Inglese.....	6
3.	Note alla traduzione italiana	10
4.	Prefazione.....	11
5.	Introduzione.....	11
6.	Scopo.....	11
7.	Organizzazione	11
8.	Riferimenti Normativi.....	12
9.	Marchi registrati	12
10.	Definizioni	12
10.1.	A.....	12
10.2.	B.....	16
10.3.	C.....	17
10.4.	D.....	22
10.5.	E.....	23
10.6.	F.....	24
10.7.	G.....	25
10.8.	H.....	27
10.9.	I.....	27
10.10.	L.....	28
10.11.	M.....	29
10.12.	N.....	31
10.13.	O.....	31
10.14.	P.....	32
10.15.	Q.....	34
10.16.	R.....	35
10.17.	S.....	38
10.18.	T.....	41
10.19.	U.....	52
10.20.	V.....	53
10.21.	W.....	53
11.	Annesso A (Informativo)	54

12.	Annesso B (Metodo di commento al Glossario Originale in Lingua Inglese)	55
13.	Annesso C (Metodo di commento al presente Glossario)	56

1. CONTRIBUTORI AL GLOSSARIO ORIGINALE IN LINGUA INGLESE

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

2. STORICO DELLE MODIFICHE DEL GLOSSARIO ORIGINALE IN LINGUA INGLESE

Version 1.3 d.d. May, 31st 2007	
<p><i>New terms added:</i></p> <ul style="list-style-type: none"> - action word driven testing - bug tracking tool - coverage measurement tool - modelling tool - monkey testing - scripted testing - specification-based technique - stress testing tool - structure-based technique - unit test framework - white box technique 	<p><i>Terms changed:</i></p> <ul style="list-style-type: none"> - basic block - control flow graph - defect management tool - independence of testing - project risk - risk-based testing - test comparator - test process
Version 2.0 d.d. December, 2nd 2007	
<p><i>New terms added:</i></p> <ul style="list-style-type: none"> - attack - buffer - buffer overflow - bug taxonomy - classification tree - control flow analysis - continuous representation - cost of quality - defect based technique - defect based test design technique - defect taxonomy - error seeding tool - Failure Mode, Effect and Criticality Analysis (FMECA) - false-fail result - false-pass result - false-negative result - false-positive result - fault attack - fault seeding - fault seeding tool - hazard analysis - hyperlink - hyperlink tool - load profile - operational acceptance testing - operational profile - orthogonal array 	<p><i>Terms changed:</i></p> <ul style="list-style-type: none"> - bebugging - error seeding - Failure Mode and Effect Analysis (FMEA) - Fault Tree Analysis (FTA) - modified multiple condition testing - process cycle test - root cause - specification-based technique - stress testing - test charter

<ul style="list-style-type: none"> - orthogonal array testing - pairwise testing - performance profiling - pointer - procedure testing - process improvement - production acceptance testing - qualification - reliability growth model - retrospective meeting - risk level - risk type - root cause analysis - safety critical system - software attack - Software Failure Mode and Effect Analysis (SFMEA) - Software Failure Mode Effect and Criticality Analysis (SFMECA) - Software Fault Tree Analysis (SFTA) - software life cycle - staged representation - system of systems - test design - test estimation - test implementation - Test Maturity Model Integration (TMMi) - test progress report - test rig - test schedule - test session - wild pointer 	
<p>Version 2.1 d.d. April, 1st 2010</p>	
<p><i>New terms added:</i></p> <ul style="list-style-type: none"> - accuracy testing - acting (IDEAL) - agile manifesto - agile software development - assessment report - assessor - balanced scorecard - call graph 	<p><i>Terms changed:</i></p> <ul style="list-style-type: none"> - attack - buffer overflow - control flow analysis - cyclomatic complexity - desk checking - entry point - exit point - Failure Mode, Effects, and

<ul style="list-style-type: none"> - causal analysis - cause-effect diagram - change management - charter - checklist-based testing - clear box testing - codependent behavior - content-based model - corporate dashboard - critical success factor - critical testing processes - CTP - dashboard - dd-path - Deming cycle - diagnosing (IDEAL) - EFQM excellence model - emotional intelligence - establishing (IDEAL) - extreme programming - fishbone diagram - Goal Question Metric - GQM - hyperlink test tool - IDEAL - indicator - initiating (IDEAL) - Ishikawa diagram - lead-assessor - learning (IDEAL) - lifecycle model - load testing tool - manufacturing-based quality 	<p>Criticality</p> <p>Analysis (FMECA)</p> <ul style="list-style-type: none"> - invalid testing - memory leak - modeling tool - monkey testing - operational acceptance testing - performance testing tool - risk-based testing - risk type - scribe - Software Failure Mode, Effects, and Criticality Analysis (SFMECA) - static analysis - static testing - systems of systems - test estimation - use case - use case testing
---	--

<ul style="list-style-type: none"> - maturity level - maturity model - maturity level - Mean Time Between Failures (MTBF) - Mean Time To Repair - mind-map: - MTBF - MTTR - Pareto analysis - Post- project meeting - process assessment - process model - product-based quality - project retrospective - quality gate - Rational Unified Process - risk category - RUP - scorecard - SCRUM - session-based test management - session-based testing - Software Process Improvement - SPI - standard - STEP - structure-based test design technique - suitability testing - SUMI - Systematic Test and Evaluation Process - test deliverable 	
---	--

<ul style="list-style-type: none">- test improvement plan- Test Process Group- test process improvement manifesto- test process improver- Total Quality Management- TPG- TQM- transactional analysis- transcendent-based quality- user-based quality- value-based quality- Work Breakdown Structure<ul style="list-style-type: none">- WBS	
---	--

3. NOTE ALLA TRADUZIONE ITALIANA

Alcuni termini (molto pochi) del Glossario Originale in lingua inglese nella Versione 2.0, non sono stati tradotti direttamente per vari motivi (generalmente per sinonimia eccessiva con termini molto più usati nella letteratura tecnica).

In ogni caso :

1. La presente traduzione del Glossario in lingua italiana è completamente autocontenuta, per cui nel presente documento non esiste nessun riferimento a termini che non siano compresi nella traduzione medesima.
2. La corrispondenza dei termini tra la versione 2.0 del Glossario in lingua inglese e la presente corrispondente traduzione in lingua italiana, è descritta in un ulteriore documento chiamato :

“DOCUMENTO DI TRACCIABILITÀ dei termini tra GLOSSARIO ISTQB in LINGUA ORIGINALE e GLOSSARIO ITA-STQB tradotto in LINGUA ITALIANA”

che è inteso a supporto e completamento del presente documento.

Questo a garanzia del fatto che tutti e soli i termini del Glossario in lingua inglese in Ver. 2.0 siano stati valutati per l'eventuale traduzione, e che a partire dai documenti prodotti ci sia la completa possibilità di effettuare una congruente preparazione dei syllabus, dei corsi in lingua italiana e della traduzione delle domande per gli esami.

4. PREFERAZIONE

Nel compilare questo glossario il gruppo di lavoro ha cercato le idee e i commenti di uno spettro di opinioni il più largo possibile nei settori e nelle organizzazioni in campo industriale e del commercio e nelle agenzie governative, con l'obiettivo di produrre uno standard internazionale per il testing che potesse essere accettato nel campo più ampio possibile. Sarà raro, se non impossibile, trovare un accordo unanime nella compilazione di un documento di questa natura. I contributi ricevuti per questo glossario provengono dalle comunità di testing di Australia, Belgio, Finlandia, Francia, Germania, India, Israele, Paesi Bassi, Norvegia, Portogallo, Svezia, Svizzera, Regno Unito, Spagna e USA.

Molti software tester hanno usato BS 7925-1 fin dalle pubblicazioni originali nel 1998. La qualifica sia a livello Foundation che a livello Practitioner è servita anche come referenze di primo piano per l'Information Systems Examination Board (ISEB). Lo standard è stato inizialmente sviluppato soprattutto per il collaudo di componenti, ma dalla sua pubblicazione sono state presentate molte critiche e idee per nuove definizioni, volte sia a migliorare che a espandere lo standard al fine di coprire uno spettro più ampio di software testing. La nuova versione del glossario sul testing ha incorporato molti degli aggiornamenti suggeriti. Il glossario sarà usato come modello di base per il piano di qualifica per software testing del "International Software Testing Qualification Board" (ISTQB)-

5. INTRODUZIONE

Quando sorgono ambiguità come conseguenza dell'incapacità di differenziare adeguatamente tra alcuni termini come "statement coverage" e "decision coverage", "test suite", "test specification", "test plan" e altri termini simili che servono a formare un'interfaccia tra vari settori della società, molto tempo e sforzo viene sprecato all'interno e tra l'industria, il commercio e le istituzioni accademiche. Inoltre l'utilizzo professionale o tecnico di questi termini è spesso soggetto a disaccordi, e con differenti significati attribuitigli.

6. SCOPO

Questo documento presenta concetti, termini e definizioni, creati per aiutare la comunicazione nell'ambito del software testing e di discipline correlate.

7. ORGANIZZAZIONE

Il glossario è stato organizzato in una singola sezione di definizioni ordinate alfabeticamente. Alcuni termini sono stati preferiti ad altri sinonimi, nel qual caso viene la definizione nel termine preferenziale, mentre i sinonimi rimandano a quello. Ad esempio *testing strutturale* rimanda a *testing white-box*. Per i sinonimi viene usato l'indicatore "Vedere".

Sono inoltre usati i riferimenti incrociati "Vedere anche". Essi sono usati per navigare rapidamente al termine corretto. I riferimenti incrociati "Vedere anche" sono costruiti per relazioni come da termine generico a termine specifico, e per significati sovrapposti tra i due termini.

8. RIFERIMENTI NORMATIVI

Al momento della pubblicazione, l'edizione indicata era quella valida. Tutti gli standard sono soggetti a revisione, e i contraenti che si basano su questo standard sono invitati a valutare la possibilità di applicare l'edizione più recente degli standard elencati più sotto. I membri di IEC e ISO mantengono i registri degli standard internazionali correntemente validi.

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

9. MARCHI REGISTRATI

In questo documento vengono usati i seguenti marchi registrati:

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

10. DEFINIZIONI

10.1. A

abbinamento definizione-utilizzo: L'associazione della definizione di una variabile con l'utilizzo stesso della variabile. L'utilizzo della variabile può essere "computazionale" (es. moltiplica) o "predicato" (es. definire la scelta di esecuzione di un cammino).

accettazione: Vedere testing di accettazione.

accuratezza: La capacità del prodotto software di fornire il risultato corretto o previsto, oppure con il necessario grado di precisione. [ISO 9126] Vedere anche testing di funzionalità.

adattabilità: La capacità del prodotto software di essere adattato per ambienti differenti senza l'utilizzo di altre azioni o l'impiego di altre risorse rispetto a quelle già inserite per questo scopo nel software considerato. [ISO 9126] Vedere anche portabilità.

affidabilità: La capacità del prodotto software di eseguire le funzionalità richieste, sotto prefissate condizioni, per uno specifico periodo di tempo o per uno specificato numero di operazioni. [ISO 9126]

agile testing: Pratica di testing adottata nell'ambito di un progetto che utilizzi metodologie agile, come la extreme programming (XP), trattando lo sviluppo come il "cliente" del testing e quindi enfatizzando il paradigma di progettazione test-first. Vedere anche *sviluppo guidato dal test*.

agile manifesto: Una dichiarazione sui valori che sono alla base dello sviluppo del software agile.

Il valori

sono:

- Individui e interazioni verso processi e strumenti
- Software funzionante verso documentazione completa
- Collaborazione con i clienti verso negoziazione contrattuale
- Risposta al cambiamento verso il rispetto di un piano.

Vedere anche *sviluppo SW agile*.

albero di classificazione: Un albero che mostra le partizioni di equivalenza ordinate in modo gerarchico e che viene usato per progettare test case nel metodo dell'albero di classificazione. Vedere anche *metodo dell'albero di classificazione*.

alpha testing: Testing operativo (effettivo o simulato) svolto da parte di potenziali utenti o di un test team indipendente nell'ambiente di sviluppo, ma al di fuori dell'organizzazione di sviluppo. Tale testing viene spesso utilizzato per software off-the-shelf come una forma di testing di accettazione interno.

ambiente di test: Un ambiente comprendente hardware, strumentazione, simulatori, tool software ed altri elementi di supporto necessari per condurre un test. [In accordo con IEEE 610]

ambiente operativo: L'insieme dei prodotti hardware e software installati presso i siti degli utenti o dei clienti, laddove il componente o sistema sotto test sarà usato. Il software può includere sistemi operativi, sistemi di gestione di database ed altri e componenti di base.

analisi ai valori limite: Una tecnica di progettazione black-box nella quale i test case sono progettati sulla base dei valori limite. Vedere anche *valore limite*.

analisi causa-effetto: Vedere *progettazione basata sul grafo causa-effetto*.

analisi causale: L'analisi dei difetti per determinare la loro causa originale. [CMMi]

analisi del flusso di controllo: Una forma di analisi statica basata sulla rappresentazione di cammini (sequenza di eventi) nella esecuzione di un componente o di un sistema. Essa valuta l'integrità delle strutture di controllo di flusso, in cerca di possibile anomalie, come circuiti chiusi o passi di processo logicamente irraggiungibili.

analisi del flusso di dati: Una forma di analisi statica basata sulla definizione e l'utilizzo di variabili.

analisi del rischio: Il processo di valutazione dei rischi identificati per stimare il loro impatto e la loro possibilità di verificarsi (intesa come probabilità).

analisi delle cause originali: Una tecnica di analisi che mira all'identificazione delle cause originali dei difetti. Applicando delle misure correttive alle cause originali, si spera che la probabilità di ricorrenze dei difetti verrà minimizzata.

analisi delle mutazioni: Un metodo per determinare l'accuratezza di un insieme di test, basato sulla misurazione di quanto l'insieme di test possa individuare piccole modifiche (mutazioni) ad un programma.

analisi di copertura: Misura della copertura raggiunta di un specificato elemento di copertura durante l'esecuzione dei test per determinare se è necessario, con riferimento a criteri prefissati, del testing aggiuntivo e, se questo è richiesto, quali test case sono necessari.

analisi di impatto: La valutazione delle modifiche ai diversi livelli della documentazione di sviluppo, della documentazione di test e dei componenti, a seguito dell'implementazione di una data modifica ai requisiti specificati.

analisi dinamica: Il processo di valutazione del comportamento, di un sistema o componente durante l'esecuzione. , ad esempio dell'occupazione di memoria, di utilizzo della CPU etc [In accordo con IEEE 610]

analisi di Pareto: Una tecnica statistica nel processo decisionale che viene utilizzata per la selezione di un numero limitato di fattori che producono un significativo effetto globale. Nel caso di miglioramento della qualità, la grande maggioranza dei problemi (80%) sono prodotti da un ristretto numero di cause (20%).

analisi progettuale retrospettiva: Una metodologia strutturata per capitalizzare l'esperienza acquisita e per creare specifici piani di miglioramento dei prossimi progetti o prossime fasi del progetto.

analisi statica: Analisi di elementi correlati al software, come ad esempio requisiti o codice, condotta senza l'esecuzione di questi elementi. L'analisi statica è di solito effettuata per mezzo di appositi tools.

analisi statica del codice: Analisi del codice sorgente condotta senza l'esecuzione di questo software.

analisi transazionale: L'analisi delle transazioni tra le persone e all'interno delle loro menti; una transazione è definita come uno stimolo ed una risposta. Le transazioni avvengono tra le persone e tra gli stati dell'Io (segmenti di personalità) all'interno della mente di una sola persona.

analizzabilità: La capacità del prodotto software di essere diagnosticato per carenze o cause di esiti negativi nel software, o per parti del software che possano essere identificate per poter essere modificate. [ISO 9126] Vedere anche *mantenibilità*.

analizzatore: Vedere *analizzatore statico*.

analizzatore di codice: Vedere *analizzatore statico di codice*.

analizzatore statico: Un tool che effettua l'analisi statica.

analizzatore statico di codice: Un tool che effettua analisi statica del codice. Il tool verifica il codice sorgente, per certe proprietà come la conformità a standard di codifica, a metriche di qualità o di anomalie del flusso dati.

anomalia: Ogni condizione che devia dalle aspettative basate sulle specifiche dei requisiti, documenti di progetto, documenti utente, standards, etc. o dalla percezione o esperienza di qualcuno. Le anomalie possono essere trovate durante, ma non solo, revisioni, testing, analisi, compilazione, uso di prodotti software o di documentazione di riferimento.[IEEE 1044] Vedere anche *baco*, *difetto*, *scostamento*, *errore*, *guasto*, *esito negativo*, *incidente*, *problema*.

apprendibilità: La capacità del prodotto software di consentire all'utente di imparare il suo l'utilizzo [ISO 9126] Vedere anche *usabilità*.

apprendimento (IDEAL): La fase all'interno del modello IDEAL in cui si impara dalle esperienze e si migliora la propria capacità di adottare nuovi processi e tecnologie per il futuro. La fase di “apprendimento” comprende le attività di: analizzare / validare e proporre le azioni future. Vedere anche *IDEAL*.

approccio di test: L'implementazione della strategia di test per uno specifico progetto. Esso include tipicamente delle decisioni basate:

- sugli obiettivi del progetto (di test) e sulla stima del rischio,
- sui punti di partenza relativi al processo di test,
- sulle tecniche di progettazione di test che devono essere applicate,
- sui criteri di uscita e sui tipi di test da eseguire.

assessment: Vedere *assessment di processo*

assessment di processo: Una valutazione disciplinata dei processi software di un'organizzazione con riferimento ad un modello. [In accordo con ISO 15504]

assessor: Una persona che effettua un assessment; qualsiasi membro di un team di assessment.

attacco: Tentativo diretto e focalizzato a valutare la qualità, in particolare l'affidabilità, di un sistema od una sua componente (oggetto di test), tentando di forzare il verificarsi di specifici esiti negativi. Vedere anche *testing negativo*.

attacco basato sui guasti: Vedere *attacco*.

attacco software: Vedere *attacco*.

attivazione dei cammini: Scelta di un insieme di valori di input per forzare l'esecuzione di un dato cammino.

attrattività: *La capacità del prodotto software di essere attraente, piacevole per l'utente.* [ISO 9126]
Vedere anche *usabilità*.

attributo di qualità: Una caratteristica che riguarda la qualità di un elemento. [IEEE 610]

audit: Una valutazione indipendente di prodotti software o di processi per accertare l'aderenza a standards, linee guida, specifiche e/o procedure basate su criteri oggettivi, compresi documenti che specificano:

- (1) la forma o il contenuto dei prodotti che devono essere realizzati,
- (2) il processo attraverso il quale i prodotti devono essere realizzati,
- (3) le modalità con cui l'aderenza agli standard o alle linee guida deve essere misurata [IEEE 1028].

audit trail: Un percorso attraverso il quale l'input originale ad un processo (ad esempio, dati) può essere ripercorso a ritroso, prendendo come punto di partenza l'output del processo stesso. Questo facilita l'analisi dei difetti e consente lo svolgimento di un processo di audit.[In accordo con TMap]

auditing della configurazione: La funzione di controllo per verificare il contenuto di librerie di elementi della configurazione; ad esempio conformità ad uno standard. [IEEE 610]

automazione dei test: L'uso di software per eseguire o supportare attività di test, come ad esempio la gestione dei test, la progettazione dei test, l'esecuzione dei test e la verifica dei risultati.

automazione dell'esecuzione dei test: L'uso di software, come ad esempio tool di cattura/riproduzione, per controllare l'esecuzione dei test, il confronto tra i risultati attuali e quelli attesi, la descrizione delle precondizioni e di altri controlli dei test e funzioni di reportistica.

azione (IDEAL): La fase all'interno del modello IDEAL in cui i miglioramenti sono identificati, messi in pratica e distribuiti in tutta l'organizzazione. La fase "azione" comprende le seguenti attività: creare la soluzione, test pilota della soluzione, perfezionare e implementare la soluzione. Vedere anche IDEAL.

10.2. B

baco: Vedere *difetto*

balanced scorecard: Vedere anche *cruscotto aziendale, dashboard*.

barriera di qualità: Una speciale milestone di un progetto. Le barriere di qualità si posizionano tra queste fasi di un progetto fortemente dipendenti dal risultato di una fase precedente. Una barriera di qualità include un controllo formale dei documenti della fase precedente.

base del gruppo di test: Un gruppo di test case derivati dalla struttura interna di un componente o di una sua specifica, per assicurare che il 100% di uno specifico criterio di copertura sarà raggiunto.

base di test: Tutti i documenti dai quali possono essere desunti i requisiti di un componente o sistema. La documentazione sulla quale sono basati i test case. Se un documento può essere modificato solo attraverso una procedura formale di emendamento, allora la base di test è chiamata una base di test congelata. [In accordo con TMap]

base di test congelata: Un documento base di test che può essere modificato solamente da un formale processo di controllo delle modifiche. Vedere anche *baseline*.

baseline: Una specifica del prodotto software che è stata formalmente revisionata o che è stata concordata/pattuita, che nel seguito serve come la base per ulteriore sviluppo, e che può essere cambiata solamente attraverso un processo di controllo formale delle modifiche. [In accordo con IEEE 610]

bebugging: Vedere *disseminazione dei difetti*. [Abbott]

best practice: Un metodo o una pratica innovativa che contribuisce a migliorare le prestazioni di una organizzazione in un dato contesto; essi sono normalmente considerati il metodo o la pratica 'migliore' da altre organizzazioni che operano nel medesimo contesto.

beta testing: Testing operativo da parte di potenziali e/o esistenti utenti/clienti svolto presso un sito esterno senza coinvolgimento degli sviluppatori, per determinare se un componente o un sistema soddisfa o meno le esigenze dell'utente/cliente ed è conforme ai processi di business. Il beta testing viene spesso impiegato come una forma di testing di accettazione esterno per software off-the-shelf, con l'obiettivo di avere un riscontro dal mercato.

blocco base: Una sequenza di uno o più istruzioni eseguibili consecutivamente e che non contengono rami. Nota: un nodo in un grafo del flusso di controllo rappresenta un blocco base.

buffer: Un dispositivo o un'area di memorizzazione usata per memorizzare temporaneamente dati, che serve quando si hanno differenze nei tassi dei flussi dati, tempo o verificarsi di eventi, o quantità di dati che possono essere gestiti dai dispositivi o processi coinvolti nel trasferimento o nell'uso dei dati. [IEEE 610]

buffer overflow: Un esito negativo nell'accesso alla memoria provocato dal tentativo da parte di un processo di memorizzare dati oltre i limiti di un buffer di lunghezza fissata, che ha come effetto quello di sovrascrivere delle aree di memoria adiacenti o quello del sollevarsi di una eccezione di overflow. Vedere anche *buffer*.

build giornaliera: Una attività di sviluppo nella quale un sistema completo viene compilato e "linkato" ogni giorno (generalmente durante la notte) in modo che in ogni momento si possa disporre di un sistema consistente che includa tutte le ultime modifiche.

10.3. C

cammino: Una sequenza di eventi (ad esempio istruzioni eseguibili) di un componente o sistema, da un punto di ingresso ad un punto di uscita.

cammino del flusso di controllo: Vedere *cammino*.

cammino non percorribile: Un cammino che non può essere attivato da nessun insieme dei possibili valori di input.

cammino percorribile: Un cammino per il quale esiste un insieme di precondizioni e di valori di input che possano consentirne l'esecuzione.

Capability Maturity Model (CMM) : Un modello a cinque livelli che descrive gli elementi chiave di un processo software efficace. Il Capability Maturity Model copre le best practice per la pianificazione, l'ingegnerizzazione e la gestione dello sviluppo del software e per la sua manutenzione. [CMM] Vedere anche *Capability Maturity Model Integration (CMMi)*.

Capability Maturity Model Integration (CMMi) : Un modello che descrive gli elementi chiave di un efficace sviluppo di prodotto e di un efficace processo di manutenzione. Il Capability Maturity Model Integration copre le best practice per la pianificazione, l'ingegnerizzazione e la gestione dello sviluppo del prodotto e per la sua manutenzione. Il CMMi è il successore designato del CMM. [CMMi] Vedere anche *Capability Maturity Model (CMM)*.

caratteristica: Un attributo di un componente o sistema specificato (o derivante) dalla documentazione dei requisiti. (per esempio affidabilità, usabilità o vincoli di progettazione) [In accordo con IEEE 1008]

caratteristica del prodotto software: Vedere *attributo di qualità*.

caratteristica del software: Vedere *caratteristica*.

caratteristica di qualità: Vedere *attributo di qualità*.

caratteristica di qualità del software: Vedere *attributo di qualità*.

CASE : Acronimo per Computer Aided Software Engineering.

caso d'uso: Una sequenza di transazioni in una interazione tra un attore ed il sistema (o sua componente) con un risultato tangibile, dove l'attore può essere un utente o chiunque possa scambiare informazioni col sistema.

CAST : Acronimo per Computer Aided Software Testing. Vedere anche *automazione dei test*.

categoria di rischio: Vedere *tipo di rischio*.

causa originale: Una sorgente di un difetto tale per cui la sua rimozione riduce od annulla il verificarsi di quel tipo di difetto. [CMMi]

certificazione: Il processo di conferma che un componente, sistema o persona soddisfi i suoi requisiti specifici; ad esempio, passando un esame.

chiusura del test: Durante la fase di chiusura di un processo di test vengono documentati i dati raccolti dalle attività completate per consolidare esperienza, testware, fatti e numeri. La fase di chiusura del test consiste nella finalizzazione e nell'archiviazione del testware e nella valutazione del processo di test, comprendente la preparazione di un report di valutazione del test. Vedere anche *processo di test*.

ciclo di Deming : Un processo iterativo in quattro fasi di problem-solving, (Plan-Do-Check-Act), in genere utilizzato nel miglioramento dei processi. [In accordo con Deming]

ciclo di test: Esecuzione del processo di test rispetto ad un singolo ed identificabile rilascio dell'oggetto del test.

ciclo-di-vita software: Il periodo di tempo che inizia quando un prodotto software viene concepito e termina quando il software non è più disponibile per essere utilizzato. Il ciclo di vita del software tipicamente include una fase concettuale, una fase di requisiti, una fase di progettazione, una fase di implementazione, una fase di test, una fase di installazione e di verifica, una fase operativa e di manutenzione e, a volte, una fase di ritiro del software. Si noti che queste fasi possono sovrapporsi o essere svolte in modo iterativo.

classe di equivalenza: Vedere *partizione di equivalenza*.

clear box testing: Vedere *white box testing*.

codice: Istruzioni di un calcolatore e definizione dei dati espresse in un linguaggio di programmazione oppure in una forma di output da parte di un assemblatore, compilatore o altro traduttore. [IEEE 610]

codice morto: Vedere *codice irraggiungibile*.

codice irraggiungibile: Codice che non può essere raggiunto e perciò risulta impossibile da eseguire.

coesistenza: La capacità del prodotto software di coesistere con altro software indipendente in un ambiente comune condividendo risorse comuni. [ISO 9126] Vedere anche *portabilità*.

comparatore: Vedere *comparatore di test*.

comparatore di test: Un tool di test per eseguire un confronto di test automatico dei risultati attuali con i risultati attesi.

compilatore: Un tool software che traduce i programmi espressi in un linguaggio di alto livello in programmi equivalenti espressi in linguaggio macchina. [IEEE 610]

complessità: Il grado con cui un componente o sistema ha una progettazione e/o una struttura interna che è difficile da comprendere, mantenere e verificare. Vedere anche *complessità ciclomatica*.

complessità ciclomatica: Il numero di cammini indipendenti nell'ambito di un programma. La complessità ciclomatica viene definita come : $L - N + 2P$, dove

- L = il numero degli archi/collegamenti in un grafo

- N = il numero di nodi in un grafo

- P = il numero di parti disconnesse del grafo (ad esempio, un grafo richiamato od una subroutine)

[In accordo con McCabe]

componente: Un elemento software minimale che può essere testato da solo.

comportamento: La risposta di un componente o sistema ad un insieme di valori di input e/o di precondizioni.

comportamento dipendente: L'eccessiva dipendenza emotiva o psicologica da un'altra persona, in particolare nel cercare di modificare il comportamento corrente (indesiderato), di quella persona, mentre si continua a sostenerla nel mantenere quel comportamento. Ad esempio nel software testing, lamentarsi del ritardo nella consegna del test e contemporaneamente gloriarsi dell'"eroico" lavoro straordinario per recuperare il ritardo, che in realtà aggrava il ritardo stesso.

comportamento temporale: Vedere *prestazioni*.

comprensibilità: La capacità del prodotto software di consentire all'utente di capire se il software è adeguato e come questo essere usato per particolari attività e condizioni di uso. [ISO 9126] Vedere anche *usabilità*.

condizione: Una espressione logica che può essere valutata come Vera o Falsa; ad esempio, $A > B$. Vedere anche *condizione di test*.

condizione composta: Due o più condizioni singole unite tramite un operatore logico (AND, OR o XOR); ad esempio, ' $A > B$ AND $C > 1000$ '.

condizione del ramo: Vedere *condizione*.

condizione di test: Un elemento o evento di un componente o sistema che potrebbe essere verificato da uno o più test case, come ad esempio una funzione, una transazione, una caratteristica, un attributo di qualità o un elemento strutturale.

condizione multipla: Vedere *condizione composta*.

configurazione: La composizione di un componente o di un sistema come definito dal numero, dalle caratteristiche e dalle interrelazioni delle sue parti (SW, HW o altro) costituenti.

conformità: La capacità del prodotto software di aderire a standard, convenzioni o norme legali e prescrizioni similari. [ISO 9126].

confronto del test: Il processo di identificazione delle differenze tra i risultati effettivi prodotti dal componente o sistema sotto test ed i risultati attesi. Il confronto del test può essere svolto durante l'esecuzione del test (confronto dinamico) oppure dopo l'esecuzione del test.

confronto dinamico: Confronto dei risultati attuali e attesi, eseguito mentre il software da testare è in esecuzione, per esempio da parte di un tool di esecuzione di test.

confronto post-esecuzione: Confronto dei risultati effettivi e dei risultati attesi, eseguito dopo che il software ha terminato la sua esecuzione.

consistenza: Il grado di uniformità, standardizzazione e coerenza tra documenti o parti di un componente o di un sistema. [IEEE 610]

contabilità della configurazione: Un elemento di gestione della configurazione, consistente nella registrazione e nella reportistica di informazioni necessarie a gestire operativamente ed efficacemente una configurazione. Questa informazione include un elenco delle identificazioni di configurazione approvate, lo stato delle modifiche proposte alla configurazione e lo stato di implementazione delle modifiche approvate. [IEEE 610]

controllo del rischio: Il processo attraverso il quale vengono identificate le decisioni e vengono realizzate le misure protettive per la riduzione dei rischi a determinati livelli o anche per il mantenimento dei rischi a livelli specificati.

controllo del test: Una attività di gestione del test che comprende lo sviluppo e l'applicazione di un insieme di azioni correttive per mantenere un progetto di test in linea con l'andamento previsto, quando il monitoraggio evidenzia uno scostamento da quello che è stato pianificato. Vedere anche *gestione del test*.

controllo della configurazione: Un'attività della gestione della configurazione, consistente nel coordinamento, nella valutazione, coordinamento, approvazione ed implementazione delle modifiche ad elementi della configurazione, dopo che sia stata formalmente stabilita la loro identificazione. [IEEE 610]

controllo delle modifiche: Vedere *controllo di configurazione*.

controllo di versione: Vedere *controllo di configurazione*.

copertura: Il grado, espresso come percentuale, con il quale uno specifico elemento di copertura è stato attivato da un insieme di test.

copertura dei cammini: La percentuale di cammini che sono stati attivati da un insieme di test. Il 100% di una copertura dei cammini implica il 100% di una copertura LCSAJ.

copertura dei rami: La percentuale di rami che è stata stimolata da un insieme di test. Una copertura dei rami del 100% implica sia una copertura delle decisioni al 100% che una copertura delle istruzioni al 100%.

copertura dei valori limite: La percentuale di valori limite che sono stati stimolati da un insieme di test.

copertura del flusso dati: La percentuale di abbinamenti definizioni/utilizzi che sono state attivati da un insieme di test.

copertura della combinazione di condizioni: Vedere *copertura delle condizioni multiple*.

copertura delle combinazioni delle condizioni dei rami : Vedere *copertura delle condizioni multiple*.

copertura delle condizioni: La percentuale degli esiti delle condizioni che sono stati esercitate da un insieme di test. La copertura delle condizioni del 100% richiede che ogni singola condizione in ogni istruzione di decisione venga testata sia come Vera che come Falsa.

copertura delle condizioni dei rami: Vedere *copertura delle condizioni*.

copertura delle condizioni multiple: La percentuale di combinazioni di tutti gli esiti delle condizioni singole, in una data istruzione, che sono stati attivati da un insieme di test. Il 100% di una copertura delle condizioni multiple implica il 100% di una copertura di determinazione delle condizioni.

copertura delle condizioni multiple modificate: Vedere *copertura di determinazione delle condizioni*.

copertura delle decisioni: La percentuale di esiti delle decisioni che sono stati attivati da un insieme di test. Il 100% della copertura delle decisioni implica sia una copertura dei rami al 100% che una copertura delle istruzioni al 100%.

copertura delle decisioni condizionate: La percentuale di tutti gli esiti delle condizioni e gli esiti delle decisioni che sono stati attivati da un insieme di test. Il 100% della copertura delle decisioni condizionate implica sia una copertura delle condizioni al 100% che una copertura delle decisioni al 100%.

copertura delle decisioni in condizioni modificate: Vedere *copertura di determinazione delle condizioni*.

copertura delle istruzioni: La percentuale di istruzioni eseguibili che sono state attivate da un insieme di test.

copertura delle partizioni di equivalenza: La percentuale di partizioni di equivalenza che sono state attivate da un insieme di test.

copertura di codice: Un metodo di analisi che determina quali parti del software sono state eseguite (coperte) dall'insieme di test e quali parti non sono state eseguite; ad esempio, copertura delle istruzioni, copertura delle decisioni, copertura delle condizioni.

copertura di determinazione delle condizioni: La percentuale di tutti gli esiti delle condizioni singole, che influenzano in modo indipendentemente l'esito di una decisione, che sono state attivate da un insieme di test case. Il 100% di una copertura di determinazione delle condizioni implica il 100% di una decisione di copertura delle condizioni.

copertura di N-commutazioni: La percentuale di sequenze di N+1 transizioni che sono state attivate da un insieme di test. [Chow]

copertura LCSAJ: La percentuale di LCSAJ di un componente che è stato attivato da un insieme di test. Il 100% di una copertura LCSAJ implica il 100% di una copertura delle decisioni.

copertura strutturale: Misure di copertura basate sulla struttura interna di un componente o sistema.

costo della qualità: Il costo totale delle attività relative alla Qualità. Esso è di norma suddiviso in: costi di prevenzione, costi di verifica, costi di difettosità interna e di difettosità esterna.

COTS : Acronimo per Commercial Off-The-Shelf software. Vedere *software off-the-shelf*.

criterio di accettazione: Il criterio di uscita che un componente o sistema deve soddisfare per poter essere accettato da un utente, un cliente od un'altra entità autorizzata. [IEEE 610]

criterio di completamento: Vedere *criterio di uscita*.

criterio di completamento del test: Vedere *criterio di uscita*.

criterio di ingresso: L'insieme di condizioni generiche e specifiche per consentire ad un processo di eseguire una funzionalità predefinita; ad esempio una fase di test. Lo scopo di criteri di ingresso è di impedire l'esecuzione di una funzionalità quando essa causerebbe più lavoro (inutile) rispetto al lavoro necessario per rimuovere i criteri di ingresso con esito negativo. [Gilb e Graham]

criterio di ripresa: Le attività di testing che devono essere ripetute quando il testing viene fatto ripartire dopo una sospensione. [In accordo con IEEE 829]

criterio di sospensione: Il criterio usato per fermare (temporalmente) tutte o in parte le attività di testing sugli elementi di test. [In accordo con IEEE 829]

criterio di superamento/fallimento: Regole di decisione usate per determinare se un elemento di test (funzione o caratteristica) ha superato o fallito un test. [IEEE 829]

criterio di uscita: L'insieme di condizioni generiche e specifiche, concordate con gli stakeholders, per permettere ad un processo di essere ufficialmente rilasciato. Lo scopo di criteri di uscita è di impedire che una funzionalità venga considerata rilasciabile, quando ci sono parti della funzionalità (sospese od arretrate) che non sono ancora state completate. I criteri di uscita sono usati per predefinire le modalità e per pianificare quando concludere il testing. [In accordo con Gilb e Graham])

cruscotto aziendale: una rappresentazione delle prestazioni di business e/o di processi di una organizzazione.. Vedere anche *balanced scorecard, dashboard*.

CTP- Critical Testing Processes: Un modello, basato sui contenuti, di miglioramento del processo di test, focalizzato su dodici processi aziendali critici. Essi includono processi di alta visibilità, in base ai quali il management ed altri interlocutori giudicano le competenze, e processi fondamentali, le cui prestazioni impattano il profitto e la visibilità aziendale.

10.4. D

dashboard: Una rappresentazione di misure dinamiche di performance operativa per una attività od organizzazione, utilizzando metriche rappresentate con metafore visive come "quadranti", "grafi" e altri dispositivi simili a quelli del cruscotto di un'automobile, in modo che gli effetti di eventi o attività possano essere facilmente compresi e correlati agli obiettivi operativi. Vedere anche *cruscotto aziendale, scorecard*.

dati di test: Dati che esistono (per esempio, in un database) prima che un test venga eseguito e che coinvolgono o sono coinvolti dal componente o sistema sotto test.

dd-path : Un percorso esecutivo (di solito attraverso un grafo che rappresenta il programma, come un flow-chart), che non include alcun nodo condizionale, come il percorso tra le due punti di decisione.

debugger: Vedere *tool di debugging*.

debugging: Il processo di scoperta, analisi e rimozione delle cause degli esiti negativi nel funzionamento del software.

decisione: Un punto del programma nel quale il flusso di controllo ha due o più percorsi alternativi. Un nodo con due o più collegamenti per separare rami.

definizione (IDEAL): La fase all'interno del modello IDEAL in cui sono programmate le specifiche di come un'organizzazione raggiungerà il suo obiettivo di miglioramento. Tale fase comprende le attività: definire le priorità, sviluppare l'approccio e dettagliare le azioni del piano. Vedere anche *IDEAL*.

definizione dei dati: Una istruzione eseguibile in cui ad una variabile viene assegnato un valore.

densità dei difetti: Il numero di difetti identificati in un componente o sistema diviso per la dimensione del componente o sistema (espressa in termini di una misura standard; ad esempio, linee di codice, numero di classi o function points).

densità dei guasti: Vedere *densità dei difetti*.

desk checking: Testing di software o delle sue specifiche attraverso una simulazione manuale della sua esecuzione. Vedere anche *testing statico*.

diagnosi (IDEAL): La fase all'interno del modello IDEAL in cui si accerta dove si è, rispetto a dove si vorrebbe essere. La fase di diagnosi consiste nelle attività di: caratterizzazione dello stato attuale e di quello desiderato e sviluppo delle raccomandazioni. Vedere anche *IDEAL*.

diagramma a lisca-di-pesce: Vedere *diagramma causa-effetto*.

diagramma causa-effetto: Una rappresentazione grafica utilizzata per organizzare e visualizzare le interrelazioni delle varie possibili originali di un problema. Le possibili cause di un difetto reale o potenziale o di un esito negativo sono organizzate in categorie e sottocategorie in una struttura ad albero orizzontale, con il (potenziale) difetto o esito negativo come nodo principale. [da Juran]

diagramma di Ishikawa: Vedere *diagramma causa-effetto*.

diagramma di stato: Un diagramma che raffigura gli stati che un componente o sistema può assumere e mostra gli eventi o le circostanze che causano e/o impattano un cambiamento da uno stato ad un altro. [IEEE 610]

dichiarazione di test: Una dichiarazione delle finalità del test e possibilmente delle idee su come eseguire il test. Le dichiarazioni dei test sono usate nel testing esplorativo. Vedere anche *testing esplorativo*.

difetto: Una falla in un componente o sistema che può causare un comportamento negativo del componente o sistema nell'esecuzione delle sue funzionalità previste, come ad esempio una istruzione o una definizione dei dati scorretta. Un difetto, se incontrato durante l'esecuzione, può causare un esito negativo del componente o sistema.

dirty testing: Vedere *testing negativo*.

disponibilità: Il grado col quale un sistema od un suo componente è operativo ed accessibile quando ne viene richiesto l'uso. Spesso viene espressa come valore percentuale.[IEEE 610]

disseminazione degli errori: Vedere *disseminazione dei guasti*.

disseminazione dei guasti: Il processo di inserimento intenzionale di difetti noti a quelli già presenti nel componente o sistema, con lo scopo di monitorare il tasso di rilevamento e rimozione dei difetti e di stimare il numero dei difetti rimanenti. [IEEE 610]

documentatore: La persona che registra ogni difetto menzionato ed ogni suggerimento per il miglioramento del processo durante un incontro di revisione, sotto forma di registro. Il documentatore dovrebbe assicurare che la forma di registrazione sia leggibile e comprensibile.

dominio: L'insieme dal quale possono essere selezionati i valori di input e/o output validi.

dominio di input: L'insieme dal quale possono essere selezionati i valori di input validi. Vedere anche *dominio*.

dominio di output: L'insieme dal quale possono essere selezionati i valori di output validi. Vedere anche *dominio*.

driver: Un componente software o un tool di test che sostituisce il componente stesso, che ha il compito di controllare e/o di richiamare un componente o sistema. [In accordo con TMap]

driver del test: Vedere *driver*.

10.5. E

effetto sonda: L'effetto sul componente o sistema da parte dello strumento di misurazione quando il componente o sistema viene misurato, ad esempio da un tool di testing delle prestazioni o da un monitor. Per esempio le prestazioni possono essere lievemente peggiorate quando vengono usati i tool di testing delle prestazioni.

efficienza: La capacità del prodotto software di fornire appropriate prestazioni, relativamente alla quantità di risorse utilizzate nell'ambito di condizioni prefissate. [ISO 9126]

EFQM (European Foundation for Quality Management): Un modello di eccellenza non prescrittivo per un sistema di gestione della qualità di un'organizzazione, definito e di proprietà della Fondazione Europea per la Gestione della Qualità, basato su cinque criteri 'Abilitativi' (che comprendono ciò che un'organizzazione fa), e quattro criteri 'Risultati' (che comprendono ciò che un'organizzazione realizza).

elementary comparison testing: Una tecnica di progettazione di test black-box nella quale i test cases sono progettati per eseguire combinazioni di input usando la tecnica di copertura di determinazione delle condizioni. [TMap]

elemento di configurazione: Un insieme di hardware e/o software, che viene progettato e trattato come una singola entità nel processo di gestione della configurazione. [IEEE 610]

elemento di copertura: Un' entità o proprietà usata come base per il test di copertura; ad esempio, istruzione di codice o partizioni di equivalenza.

elemento di test: L'elemento individuale che deve essere testato. Generalmente c'è un oggetto del test e molti elementi di test. Vedere anche *oggetto del test*.

emulatore: Un dispositivo, un programma SW od un sistema che accetta gli stessi input e produce gli stessi output di un dato sistema. [IEEE 610] Vedere anche *simulatore*.

error guessing: Una tecnica di progettazione nella quale l'esperienza del tester viene usata per anticipare quali difetti possano essere presenti nel componente o sistema da testare (come risultato di errori commessi nella sua realizzazione) e per progettare test specifici volti a fare emergere tali errori.

errore: Una azione umana che produce un risultato scorretto. [In accordo con IEEE 610]

esecuzione del test: Il processo che consiste nell'eseguire un test sul componente o sistema sotto test, producendo un risultato effettivo (o più risultati effettivi).

esito: Vedere *risultato*.

esito atteso: Vedere *risultato atteso*.

esito attuale: Vedere *risultato attuale*.

esito del test: Vedere *risultato*.

esito della decisione: Il risultato di una decisione, la quale perciò determina i rami che devono essere successivamente percorsi.

esito di una condizione: La valutazione di una condizione a Vero o Falso.

esito negativo: Scostamento del componente o sistema dal risultato o comportamento atteso o dal servizio che dovrebbe fornire. [In accordo con Fenton]

esito previsto: Vedere *risultato atteso*.

evidenza di esito negativo: La manifestazione fisica o funzionale di un esito negativo. Per esempio, un sistema in tale stato di evidenza di esito negativo può essere caratterizzato da operazioni lente, output scorretti o dalla interruzione dell'esecuzione. [IEEE 610]

extreme programming: Una metodologia di ingegneria del software utilizzata all'interno dello sviluppo software "agile", le cui pratiche fondamentali sono: la programmazione a coppie, le revisioni estese del codice, lo unit test di tutte le parti di codice e la semplicità e la chiarezza nella codifica. Vedere anche *sviluppo software agile*.

10.6. F

Failure Mode and Effect Analysis (FMEA) : Un approccio sistematico di identificazione del rischio e di analisi all'identificazione delle diverse modalità di esiti negativi e delle relative attività di prevenzione. Vedere anche *Failure Mode, Effect and Criticality Analysis (FMECA)*.

Failure Mode, Effect and Criticality Analysis (FMECA) : Una estensione dell'approccio FMEA, che in aggiunta include una analisi di criticità, usata per tracciare la probabilità di esiti negativi rispetto alla severità delle loro conseguenze. Il risultato focalizza l'attenzione su quegli esiti negativi con alta probabilità di severità delle conseguenze , consentendo di indirizzare e concentrare lo sforzo di rimedio/correzione laddove esso produrrà il maggior valore. Vedere anche *Failure Mode and Effect Analysis (FMEA)*.

fallimento : Un test è considerato fallito se il suo risulta attuale non corrisponde al suo risultato atteso.

fallimento del test: Vedere *fallimento*.

fase di definizione dei requisiti: Il periodo di tempo nel ciclo di vita del software durante il quale i requisiti di un prodotto software sono definiti e documentati. [IEEE 610]

fase di esecuzione dei test: Il periodo di tempo, nel ciclo di vita di sviluppo del software, durante il quale i componenti di un prodotto software sono eseguiti e il prodotto software viene valutato per determinare se i suoi requisiti sono stati o soddisfatti o meno. [IEEE 610]

fase di test: Un separato insieme di attività di test di incluse in una fase di un progetto, come ad esempio le attività di esecuzione di un livello di test. [In accordo con Gerrard]

fattore critico di successo: Un elemento che è necessario ad un'organizzazione o ad un progetto per realizzare la sua missione. Essi sono gli elementi o le attività critiche richieste per garantire il loro successo. Vedere anche il *modello basato sul contenuto*.

Fault Tree Analysis (FTA) : Una tecnica usata per analizzare le cause dei guasti (difetti). La tecnica modella visivamente in che modo le relazioni logiche tra gli esiti negativi, gli errori umani e gli eventi esterni si possono combinare per causare specifici guasti con l'obiettivo di renderle visibili.

finalità del test: Un obiettivo o scopo della progettazione ed esecuzione di un test.

flusso dati: Una rappresentazione astratta della sequenza e delle possibili modifiche allo stato di oggetti dati, dove lo stato è uno dei seguenti: creazione, utilizzo o cancellazione. [Beizer]

flusso di controllo: Una sequenza di eventi (cammini) nella esecuzione di un componente od un sistema.

framework di unit test: Un tool che fornisce un ambiente per il testing di unità o componente nel quale un componente può essere testato in isolamento o con stub e driver adeguati. Esso fornisce anche altro supporto allo sviluppatore, come funzionalità di debugging. [Graham]

Function Point Analysis (FPA): Metodo che si prefigge di misurare la dimensione della funzionalità di un sistema informatico. Le misurazione è indipendente dalla tecnologia. Questa misura può essere usata come base per la misurazione della produttività, la stima delle risorse necessarie e per il controllo del progetto.

funzionalità: La capacità del prodotto software di fornire funzioni che soddisfino le esigenze esplicite ed implicite quando il software viene utilizzato sotto specifiche condizioni. [ISO 9126]

10.7. G

generatore di test: Vedere *tool di preparazione dei dati di test*.

gestione degli incidenti: Il processo di riconoscimento, investigazione, attivazione di contromisure e risoluzione degli incidenti. Esso implica la registrazione degli incidenti, la loro classificazione e l'identificazione del loro impatto. [In accordo con IEEE 1044]

gestione dei difetti: Il processo di riconoscimento, diagnosi dei difetti e di avvio e completamento delle relative azioni correttive di tali difetti. Esso comprende la registrazione dei difetti, la loro classificazione e l'analisi del loro impatto. [In accordo con IEEE 1044]

gestione dei problemi: Vedere *gestione dei difetti*.

gestione del rischio: Applicazione sistematica di procedure e pratiche per le attività di identificazione, analisi, definizione delle priorità e controllo del rischio.

gestione del test: La pianificazione, stima, monitoraggio e controllo delle attività di test, generalmente condotte da un responsabile del test.

gestione della configurazione: Una disciplina che si avvale di compiti amministrativi, tecnici ed di sorveglianza per identificare e documentare le caratteristiche fisiche e funzionali di un elemento di configurazione, controllare le modifiche apportate a tali caratteristiche, registrare e commentare l'implementazione e lo stato delle modifiche, e verificarne la conformità a specifici requisiti. [IEEE 610]

gestione della qualità: Attività coordinate per gestire e controllare una organizzazione riguardo alla qualità. La gestione ed il controllo della qualità generalmente include la creazione della politica della qualità e degli obiettivi di qualità, della pianificazione della qualità, del controllo della qualità, della quality assurance e del miglioramento della qualità. [ISO 9000]

gestione delle eccezioni: Comportamento di un componente o sistema in risposta ad input errati, da parte sia di un operatore umano che di un altro componente o sistema, o ad un esito negativo interno.

gestione delle modifiche: (1) un approccio strutturato per gestire le transizioni di individui, team ed organizzazioni da uno stato corrente ad uno desiderato stato futuro. (2) una modalità controllata per effettuare una modifica, od una proposta di modifica, ad un prodotto o servizio. Vedere anche *gestione della configurazione*.

gestione del testing a sessioni: Un metodo per la gestione e la misurazione di sessioni di test, ad esempio testing sperimentale.

GQM (Goal Question Metric) : un approccio alla misurazione del software che utilizza un modello a tre livelli: livello concettuale (obiettivo), livello operativo (domanda) e il livello quantitativo (metrica).

grafo causa-effetto: Una rappresentazione grafica di input e/o stimoli (cause) con i loro associati output (effetti), che può essere usata per la progettazione di test case.

grafo del flusso di controllo: Una rappresentazione grafica di tutte le possibili sequenze di eventi (cammini) nella esecuzione di un componente od un sistema.

grafo delle chiamate: Una rappresentazione astratta delle relazioni di richiamo fra le subroutine di un programma.

gruppo di controllo della configurazione: Un gruppo di persone responsabile della valutazione, dell'approvazione o meno di modifiche apportate ad elementi della configurazione. [IEEE 610]

gruppo di controllo delle modifiche: Vedere *gruppo di controllo di configurazione*.

gruppo di test: Vedere *insieme di test*.

guasto: Vedere *difetto*.

guida di installazione: Istruzioni fornite su qualsiasi supporto adeguato, che guidano l'installatore attraverso il processo di installazione. Questa può essere una guida manuale, una procedura passo-passo, un wizard di installazione, o qualsiasi altra simile descrizione del processo.

10.8. H

hazard analysis : Una tecnica utilizzata per caratterizzare gli elementi di rischio. I risultati di una hazard analysis guideranno i metodi usati per lo sviluppo ed il testing di un sistema. Vedere anche *analisi del rischio*.

hyperlink: Un puntatore in una pagina web che conduce ad altre pagine web.

hyperlink test tool: Un tool usato per verificare che in un sito web non siano presenti hyperlink che non puntino correttamente ad altre pagine web.

10.9. I

IDEAL,: Un modello di miglioramento organizzativo, che serve come una guida per l'avvio, la pianificazione e l'implementazione di azioni di miglioramento. Il modello IDEAL è così chiamato per le iniziali (inglesi) delle sue 5 fasi: inizio, diagnosi, definizione, azione, apprendimento.

identificazione del rischio: Il processo di identificazione dei rischi usando tecniche come il brainstorming, le liste di controllo e lo storico degli esiti negativi.

identificazione della configurazione: Un'attività della gestione della configurazione, consistente nella selezione degli elementi di configurazione di un sistema da modificare e nella registrazione delle loro caratteristiche funzionali e strutturali in documentazione tecnica. [IEEE 610]

idoneità: La capacità del prodotto software di fornire un appropriato insieme di funzioni per specifiche attività e obiettivi dell'utente. [ISO 9126] Vedere anche *funzionalità*.

implementazione dei test: Il processo di sviluppo e assegnazione di priorità alle procedure di test, creando dati di test e, opzionalmente, preparando le test harness e scrivendo script di test automatici.

incidente: Ogni evento il cui verificarsi richiede un'indagine. [In accordo con IEEE 1008]

incidente di test: Vedere *incidente*.

incidente di test del software: Vedere *incidente*.

indicatore: una misura che può essere usata per stimare o prevedere un'altra misura. [ISO 14.598]

indicatore chiave delle prestazioni: Vedere *indicatore delle prestazioni*.

indicatore delle prestazione del test: Una metrica di alto livello di efficacia e/o efficienza usata per guidare e controllare lo sviluppo progressivo del test, come ad esempio la percentuale di rilevamento dei difetti.

indicatore di prestazione: Una metrica di alto livello di efficacia e/o efficienza usata per guidare e controllare il progresso dello sviluppo, come ad esempio il ritardo rispetto al piano di sviluppo. [CMMi]

indipendenza del testing: Separazione delle responsabilità, che incoraggia il raggiungimento del testing imparziale. [In accordo con DO-178b]

infrastruttura di test: Gli elementi organizzativi necessari per svolgere il testing, consistenti di ambienti di test, tool di test, ambienti d'ufficio e procedure.

inizio (IDEAL): La fase all'interno del modello IDEAL in cui si gettano le basi per uno sforzo di miglioramento successo. La fase di "inizio" delle attività è costituito: definire il contesto, ottenere la sponsorizzazione e impostare le infrastrutture. Vedere anche *IDEAL*.

- input:** Una variabile (memorizzata internamente su un supporto od esterna) che è letta da un componente.
- input del test:** I dati ricevuti da una sorgente esterna all'oggetto del test durante l'esecuzione. La sorgente esterna può essere hardware, software o umana.
- input specificato:** Un input per il quale la specifica prevede un risultato.
- insieme di test:** Un insieme di diversi test case per un componente o sistema sotto test, nel quale la postcondizione di un test viene spesso usata come la preconditione per il successivo.
- insieme di test case:** Vedere *insieme di test*.
- installabilità:** La capacità del prodotto software di essere installato in un specifico ambiente. [ISO 9126]. Vedere anche *portabilità*.
- instrumentatore:** Un tool software per realizzare l'strumentazione.
- strumentazione:** L'inserimento di codice aggiuntivo all'interno del programma con lo scopo di raccogliere informazioni relative al comportamento del programma durante l'esecuzione, come ad esempio la misurazione della copertura del codice.
- intake test :** Una speciale istanza di uno smoke test per decidere se il componente o sistema è pronto per un ulteriore e dettagliato testing. Un intake test viene generalmente effettuato all'inizio della fase di esecuzione dei test. Vedere anche *smoke test*.
- integrazione:** Il processo di combinazione di componenti o sistemi in aggregati più grandi
- integrazione funzionale:** Un approccio di integrazione che combina i componenti o sistemi con l'obiettivo di ottenere una funzionalità base funzionante in anticipo. Vedere anche *testing di integrazione*.
- intelligenza emotiva:** la competenza, la capacità e l'abilità di individuare, valutare e gestire le emozioni del proprio io, di altri e di gruppi di collaboratori.
- interoperabilità:** La capacità del prodotto software di interagire con uno o più componenti o sistemi esterni. [In accordo con ISO 9126] Vedere anche *funzionalità*.
- ipertesto:** Un puntatore in una pagina web che punta ad altre pagine web
- ispettore:** Vedere *revisore*.
- ispezione:** Un tipo di peer review che si basa su un esame visuale di documenti per rilevare difetti, come ad esempio violazioni di standard di sviluppo e non-conformità di documentazione di più alto livello. È la più formale tecnica di revisione ed è quindi sempre basata su una procedura documentata. [In accordo con IEEE 610, IEEE 1028] Vedere anche *peer review*.
- istruzione:** Una entità in un linguaggio di programmazione, la quale è tipicamente la più piccola indivisibile unità di esecuzione.
- istruzione eseguibile:** Un'istruzione che, quando compilata, viene tradotta in codice oggetto e che quindi verrà eseguita in modo procedurale quando il programma sarà in esecuzione.
- istruzione sorgente:** Vedere *istruzione*.

10.10. L

- lancio del test:** Esecuzione di un test su una specifica versione dell'oggetto del test.

LCSAJ: Acronimo di Linear Code Sequence And Jump, che consiste dei seguenti tre elementi (convenzionalmente identificati dai numeri di riga nel listato del codice sorgente): la partenza della sequenza lineare di istruzioni eseguibili, la fine della sequenza lineare e la riga (obiettivo) alla quale il flusso di controllo viene trasferito al termine della sequenza lineare.

lead assessor: La persona che conduce un assessment. In alcuni casi, per esempio CMMi e TMMi, quando sono condotte valutazioni formali, il lead-assessor deve essere accreditato e formalmente addestrato.

leader dell'ispezione: Vedere *moderatore*.

linguaggio di scripting: A linguaggio di programmazione nel quale vengono scritti script di test eseguibili, che vengono poi usati da una tool di esecuzione di test (ad esempio, un tool di cattura/riproduzione).

livello di maturità: Il grado di miglioramento dei processi misurato su un insieme predefinito di aree di processo, per le quali vengono raggiunti tutti i rispettivi obiettivi. [TMMi]

livello di rischio: L'importanza di un rischio come definito dalle sue caratteristiche probabilistiche e dal suo impatto. Il livello di rischio può essere usato per determinare l'intensità del testing che deve essere eseguito. Un livello di rischio può essere espresso sia qualitativamente (ad esempio alto, medio, basso) o quantitativamente.

livello di test: Un gruppo di attività di test organizzate e gestite congiuntamente. Un livello di test prevede precise responsabilità progettuali. Esempi di livelli di test sono: il test di componente, il test di integrazione, il test di sistema e il test di accettazione. [In accordo con TMap]

log di esecuzione del test: Vedere *test log*.

10.11. M

macchina a stati finiti: Un modello computazionale consistente di un numero finito di stati e di transizioni tra questi stati, possibilmente con le relative azioni. [IEEE 610]

manutenibilità: La facilità con la quale un prodotto software può essere modificato per correggere difetti, per soddisfare nuovi requisiti, per rendere più semplice la futura manutenzione o può essere adattato ad un ambiente modificato. [ISO 9126]

manutenzione: Modifiche ad un prodotto software dopo il suo rilascio per correggere difetti, per migliorare le prestazioni od altre funzionalità o per adattare il prodotto ad un ambiente modificato. [IEEE 1219]

mappa mentale: Un diagramma utilizzato per rappresentare parole, idee, attività o altri elementi connessi e disposti intorno ad una parola chiave o idea centrale. Le mappe mentali sono utilizzate per generare, visualizzare, strutturare e classificare le idee e come aiuto nelle attività di analisi, organizzazione, risoluzione di problemi, presa di decisioni e documentazione..

mascheramento dei difetti: Una situazione nella quale il verificarsi di un difetto impedisce il rilevamento di un altro difetto. [In accordo con IEEE 610]

mascheramento dei guasti: Vedere *mascheramento dei difetti*.

matrice ortogonale: Una matrice bidimensionale costruita per avere proprietà matematiche particolari, in modo che ad esempio scegliendo due specifiche colonne nella matrice si ottiene una conseguente coppia di combinazione dei valori inseriti nella matrice stessa.

maturità: (1) La capacità di un organizzazione con riferimento all'efficacia e all'efficienza dei suoi processi e delle sue pratiche di lavoro. Vedere anche *Capability Maturity Model, Test Maturity Model*.
(2) La capacità del prodotto software di evitare esiti negativi risultanti da difetti presenti nel software. [ISO 9126] Vedere anche *affidabilità*.

Mean Time Between Failure (MTBF): La media aritmetica (media) del tempointercorso fra i guasti di un sistema. Il MTBF è in genere un indicatore di un modello di crescita dell'affidabilità, che prevede che un guasto del sistema sia immediatamente riparato, come risultato di un processo di riparazione dei difetti. Vedere anche *modello di crescita dell'affidabilità*.

Mean Time To Repair (MTTR): la media aritmetica del tempo necessario al ripristino di un sistema dopo un qualsiasi guasto. Esso comprende il test per assicurarsi che il difetto sia stato risolto.

memorizzazione: Vedere *utilizzo delle risorse*.

memory leak : Un difetto nella logica di allocazione di memoria di un programma che causa il fallimento quando esso rilascia memoria dopo che ha finito di usarla, eventualmente causando il fallimento del programma dovuto alla mancanza di memoria.

metodo dell'albero di classificazione: Una tecnica di progettazione di test black-box nella quale i test case, descritti attraverso un albero di classificazione, sono progettati per eseguire combinazioni di rappresentazioni di domini di input e/o output. [Grochtmann]

metrica: Una scala di misurazione e il metodo usato per la misurazione. [ISO 14598]

metriche di copertura di Chow: Vedere *copertura di N-commutazioni*. [Chow]

miglioramento di processo: Un programma di attività progettato per migliorare le prestazioni e la maturità dei processi delle organizzazioni e il risultato di tale programma. [CMMi]

milestone: Un momento temporale in un progetto in corrispondenza del quale dovrebbero essere effettuati dei rilasci (anche intermedi) e/o dovrebbero essere ottenuti particolari risultati.

misura: Il numero o la categoria assegnata ad un attributo di una entità ottenuto da una misurazione. [ISO 14598]

misurazione: Il processo di assegnazione di un numero o categoria ad una entità per quantificare un suo attributo. [ISO 14598]

mitigazione del rischio: Vedere *controllo del rischio*.

modello a V: Un modello per descrivere le attività del ciclo di sviluppo del software dalle specifiche dei requisiti alla manutenzione. Il modello a V illustra in che modo le attività di testing possono essere integrate in ogni singola fase del ciclo di vita dello sviluppo del software.

modello basato sui contenuti: Un modello di processo che fornisce una descrizione dettagliata di buone pratiche di SW engineering; ad esempio pratiche di test.

modello ciclo-di-vita: La suddivisione della vita di un prodotto o di un progetto in fasi. [CMMi]. Vedere anche: *ciclo-di-vita software*.

modello di crescita dell'affidabilità: Un modello che mostra, durante il testing continuativo, la crescita in termini di affidabilità nel tempo di un componente o sistema, come risultato della rimozione di difetti che hanno causato in esiti negativi di affidabilità.

modello di maturità: una raccolta strutturata di elementi che descrivono alcuni aspetti della maturità di una organizzazione ed un supporto nella definizione e nella comprensione dei processi di una organizzazione. Un modello di maturità fornisce spesso un linguaggio comune, una visione condivisa ed un quadro di priorità per le azioni di miglioramento.

modello di processo: Uno schema in cui processi della stessa natura sono classificati, ad esempio un modello di miglioramento dei test.

modello di sviluppo incrementale: Un ciclo di vita dello sviluppo nel quale un progetto viene suddiviso in una successione di passi incrementali, ognuno dei quali rilascia una parte delle funzionalità relative ai requisiti del progetto complessivo. I requisiti sono ordinati per priorità e sono rilasciati in ordine di priorità nel relativo passo incrementale. In alcune (ma non in tutte) le versioni di questo modello del ciclo di vita, ogni sottoprogetto segue un “mini modello a V” con le sue proprie fasi di progettazione, codifica e testing.

modello di sviluppo iterativo: Un ciclo di vita dello sviluppo nel quale un progetto viene suddiviso in un numero generalmente elevato di iterazioni. Una iterazione è un ciclo di sviluppo completo che termina con un rilascio (interno od esterno) di un prodotto eseguibile (un sottoinsieme del prodotto finale) il quale si accresce ad ogni iterazione per diventare il prodotto finale.

moderatore: Il conduttore ed il principale responsabile di una ispezione o di altri processi di revisione.

modificabilità: La capacità del prodotto software di consentire che specifiche modifiche software possano essere facilmente implementate. [ISO 9126] Vedere anche *mantenibilità*.

modulo: Vedere *componente*.

monitor: Un tool software o un dispositivo hardware che gira in parallelo al componente o sistema sotto test e supervisiona, registra e/o analizza il comportamento del componente o del sistema. [In accordo con IEEE 610]

monitoraggio del test: Una attività di gestione del test che prevede attività di controllo periodico dello stato di un progetto di test. Vengono preparati dei report che confrontano lo stato attuale con quello pianificato. Vedere anche *gestione del test*.

monkey testing: Testing una svolto tramite una selezione casuale da un largo intervallo di input e/o da pulsanti premuti casualmente, ignorando come il prodotto dovrebbe essere effettivamente usato.

10.12. N

non-conformità: Mancato soddisfacimento di un specifico requisito. [ISO 9000]

note di rilascio: Un documento che identifica gli elementi di test, la loro configurazione, lo stato corrente ed altre informazioni rilasciate dallo sviluppo (e possibilmente da altri stakeholders) al testing, , all’inizio di una fase di esecuzione di test. [In accordo con IEEE 829]

numero cicломatico: Vedere *complessità cicломatica*.

10.13. O

obiettivo del test: Un insieme di criteri di uscita.

oggetto del test: Il componente o sistema da essere testato. Vedere anche *elemento di test*.

operabilità: La capacità del prodotto software di consentire agli utenti di operare su di esso e di controllarlo. [ISO 9126] Vedere anche *usabilità*.

oracolo: Vedere *oracolo del test*.

oracolo del test: Una sorgente per determinare i risultati attesi da confrontare con i risultati attuali del software sotto test. Un oracolo può essere il sistema esistente (per un benchmark), un manuale utente o un individuo con una conoscenza specialistica, ma non dovrebbe essere il codice SW sviluppato. [In accordo con Adrion]

output: Una variabile (memorizzata internamente su un supporto od esterna) che è scritta da un componente.

10.14. P

partizionamento di equivalenza: Una tecnica di progettazione di test black-box nella quale i test case sono progettati per eseguire delle istanze delle partizioni di equivalenza. I test case sono, di norma, progettati per coprire ogni singola partizione almeno una volta.

partizione di equivalenza: Una porzione di un dominio di input o di output per il quale il comportamento di un componente o sistema si assume essere sempre lo stesso (in base alle specifiche).

peer review: Una revisione di prodotto software effettuata da parte di colleghi dello sviluppatore con lo scopo di identificare difetti e/o miglioramenti. Esempi di peer review, sono l'ispezione, le review tecniche ed i walkthrough.

percentuale di rilevamento dei difetti: Il numero di difetti trovati da una sessione di test, diviso per il numero "totale", trovato cioè da quella sessione e da tutte le altre successioni successive.

percentuale di rilevamento dei guasti: Vedere *percentuale di rilevamento dei difetti*.

pianificazione del test: La attività di redazione o modifica di un piano di test.

pianificazione dell'esecuzione dei test: Uno schema per l'esecuzione di procedure di test. Le procedure di test sono incluse nella pianificazione dell'esecuzione dei test nel loro contesto e nell'ordine nel quale devono essere eseguite.

piano di miglioramento testing: Un piano per raggiungere gli obiettivi di miglioramento organizzativo del processo di testing, basato su una approfondita comprensione degli attuali punti di forza e di debolezza del processo e delle risorse del testing.

piano di test: Un documento che descrive l'obiettivo, l'approccio, le risorse e la programmazione temporale delle attività di test da eseguire. Esso identifica e specifica: le caratteristiche da testare, le attività di testing, l'esecutore di ogni singola attività, il grado di indipendenza del tester, l'ambiente di test, le tecniche di progettazione dei test, i criteri di ingresso e di uscita, fondamentali motivazioni logiche per la loro scelta e ogni rischio che possa richiedere una pianificazione imprevista. Il piano di test è il registro del processo di pianificazione del test. [In accordo con IEEE 829]

piano di test a livello: Un piano di test che indirizza un tipico livello di test. Vedere anche

piano di test di fase: Un piano di test che tipicamente riguarda una fase di test. Vedere anche *piano di test*.

piano di test di progetto: Vedere *piano di test master*.

piano di test master: Un piano di test che include più livelli di test. Vedere anche *piano di test*.

- politica di test:** Un documento di alto livello che descrive i principi, l'approccio ed i maggiori obiettivi dell'organizzazione relativa al testing.
- portabilità:** La facilità con la quale il prodotto software può essere trasferito da un ambiente hardware o software ad un altro ambiente. [ISO 9126]
- postcondizione:** Condizioni ambientali e di stato che devono essere soddisfatte dopo l'esecuzione di un singolo test o di una procedura di test.
- precondizione:** Condizioni ambientali e di stato che devono essere soddisfatte prima che il componente o sistema possa essere eseguito con un particolare test o procedura di test.
- prestazione:** Il grado con il quale un sistema o componente realizza le funzionalità per cui è stato progettato, con prefissati vincoli relativi al tempo di elaborazione ed al tasso di throughput. [In accordo con IEEE 610] Vedere anche *efficienza*.
- pretest:** Vedere *intake test*.
- priorità:** Il livello di importanza (di business) assegnato ad un elemento, ad esempio un difetto.
- problema:** Vedere *difetto*.
- procedura di test:** Vedere *specificata della procedura di test*.
- processo:** Un insieme di attività correlate tra loro, le quali trasformano un input in output. [ISO 12207]
- processo di assessment:** Una valutazione strutturata (verso uno specifico modello di riferimento) dei processi di sviluppo software di un'organizzazione. [da ISO 15504]
- processo di registrazione del test:** Vedere *test logging*.
- processo di test:** Il processo di test comprende fundamentalmente la pianificazione ed il controllo dei test, l'analisi e la progettazione dei test, l'implementazione ed esecuzione, la valutazione dei criteri di uscita, la reportistica e le attività di chiusura dei test.
- profilo delle prestazioni:** Definizione dei profili utente nel testing di prestazioni, di carico e/o stress. I profili dovrebbero riflettere l'uso previsto o effettivo basato su un profilo operativo di un componente o sistema e quindi il carico di lavoro atteso. Vedere anche *profilo di carico*, *profilo operativo*.
- profilo di carico:** Una specifica dell'attività alla quale un componente o sistema può essere sottoposto in produzione. Un profilo di carico consiste di un numero prefissato di utenti che eseguono un predefinito insieme di transazioni in uno specificato periodo di tempo ed in accordo con un predefinito profilo operativo. Vedere anche *profilo operativo*.
- profilo operativo:** L'insieme delle attività eseguite dal componente o sistema, conseguente al comportamento degli utenti durante la loro interazione col componente o sistema, e le loro probabilità di andare in esecuzione. Ogni attività è da intendersi "logica" piuttosto che "fisica" e può essere eseguita tra diverse macchine o può essere eseguita in segmenti temporali non-consecutivi.
- progettazione basata sul grafo causa-effetto:** Una tecnica di progettazione di test black-box nella quale i test case sono progettati a partire dai grafi causa-effetto. [BS 7925/2]
- progettazione di test:** (1) Vedere *specificata di progettazione dei test*. (2) Il processo di trasformazione di obiettivi generali di testing in condizioni di test e test case che siano tangibili.
- progetto:** Un insieme di attività coordinate e controllate con date di inizio e di fine, intrapreso per realizzare un obiettivo che sia conforme ai requisiti specificati, compresi i vincoli temporali, di costo e di risorse. [ISO 9000]

programmazione parallela: Un approccio di sviluppo del software in base al quale le linee di codice di un componente sono scritte da due programmatori che lavorano sullo stesso computer. Questo implicitamente significa avere una attività continuativa che assicura che le revisioni di codice sono effettuate in tempo reale.

pseudo-casuale: Una serie che sembra essere casuale ma che di fatto è generata sulla base di sequenze programmate in anticipo.

puntatore: Un elemento dati che specifica la locazione di un altro elemento dati; per esempio un elemento dati che specifica l'indirizzo del record successivo che deve essere elaborato. [IEEE 610]

puntatore errato: Un puntatore che referencia una locazione che è al di fuori del campo di visibilità di quel puntatore o che non esiste. Vedere anche *puntatore*.

punto di ingresso: Un'istruzione eseguibile od una fase di processo, che definisce un punto in cui un determinato processo è destinato ad iniziare.

punto di uscita: Un'istruzione eseguibile od una fase di processo, che definisce un punto in cui un determinato processo è destinato a terminare. .

10.15. Q

qualifica: Il processo di dimostrazione della capacità di soddisfare specifici requisiti. Notare che il termine 'qualificato' viene usato per designare lo stato corrispondente. [ISO 9000]

qualità: Il grado con il quale un componente, sistema o processo soddisfa i requisiti specificati e/o le esigenze e le ispettive utente/cliente. [In accordo con IEEE 610]

qualità base fabbricazione: Una vista della qualità, per cui la qualità si misura dal grado di conformità di un prodotto o servizio ai suoi requisiti ed alla sua prevista progettazione. Tale qualità deriva dal processo (i) utilizzato(i). [In accordo con Garvin] Vedere anche *qualità base prodotto, qualità base trascendente, qualità base utente, qualità base valore*.

qualità base prodotto: Una vista della qualità per cui la qualità si basa su un set ben definito di attributi. Questi attributi devono essere misurati in modo oggettivo e quantitativo. Le differenze nella qualità dei prodotti dello stesso tipo si può far risalire al modo in cui gli attributi di qualità specifici sono state realizzati. [In accordo con Garvin] . Vedere anche , *attributi della qualità, qualità base fabbricazione, qualità base trascendente, qualità base utente, qualità base valore*.

qualità base trascendente: Una vista della qualità, in cui la qualità non può essere definita con precisione, ma è percepita quando la si vede o se ne conosce l'assenza quando manca. Tale qualità dipende dalla percezione e da sentimenti affettivi di un individuo o di un gruppo di individui verso un prodotto. [In accordo con Garvin] Vedere anche *qualità di base fabbricazione, qualità base prodotto, qualità base 'utente, qualità base valore*.

qualità base utente: Una vista della qualità, in cui la qualità è la capacità di soddisfare bisogni, necessità e desideri degli utenti . Un prodotto o un servizio che non soddisfa le esigenze degli utenti è improbabile che possa trovare nuovi utenti. Si tratta di un approccio alla qualità contingente e dipendente dal contesto, poichè differenti caratteristiche di business richiedono differenti qualità di prodotto. [In accordo con Garvin] Vedere anche *qualità base fabbricazione, qualità base prodotto, qualità base trascendente, qualità base valore*.

qualità base valore: Una vista della qualità, in cui la qualità è definita dal prezzo. La qualità di un prodotto o servizio è quella che fornisce le prestazioni desiderate ad un costo accettabile. Tale

qualità viene determinata per mezzo di un processo decisionale con le parti interessate sul bilanciamento tra tempo, sforzo e costi. Vedere anche *qualità base fabbricazione, qualità base prodotto, qualità base trascendente*.

qualità del software: L'insieme di funzionalità e caratteristiche di un prodotto software che si riferisce alla sua capacità di soddisfare esigenze predefinite o implicite. [In accordo con ISO 9126]

quality assurance: Parte della gestione della qualità focalizzata sul fatto di fornire confidenza che i requisiti di qualità saranno soddisfatti. [ISO 9000]

10.16. R

ramo: Un blocco base che può essere selezionato per l'esecuzione, basato su un costrutto di un programma nel quale due o più cammini di programma alternativi sono disponibili; ad esempio, case, jump, go to, if then-else.

rappresentazione continua: Una struttura del modello di maturità. in cui i livelli di capacità forniscono una sequenza raccomandata per le attività di miglioramento dei processi all'interno delle aree di processo specificate. [CMMi]

Rational Unified Process: Un processo iterativo di sviluppo software (proprietario e adattabile) composto da quattro fasi del ciclo di vita di progetto: inizio, elaborazione, costruzione e transizione.

recuperabilità: La capacità del prodotto software di ristabilire uno specificato livello di prestazioni e recuperare i dati inquinati nel caso di esito negativo. [ISO 9126] Vedere anche *affidabilità*.

registratore: Vedere *documentatore*.

registrazione degli incidenti: Registrazione dei dettagli di ogni incidente che si è verificato, ad esempio durante il testing.

registrazione del test: Vedere *test log*.

report degli incidenti: Un documento che riporta ogni evento verificatosi, ad esempio durante il testing, che richieda delle indagini e/o diagnosi. [In accordo con IEEE 829]

report degli scostamenti: Vedere *report degli incidenti*.

report dei banchi: Vedere *report dei difetti*.

report dei difetti: Un documento che riporta tutti i difetti di un componente o sistema che possono causare un esito negativo durante l'esecuzione delle funzionalità del componente o sistema. [In accordo con IEEE 829]

report dei problemi: Vedere *report dei difetti*.

report del test: Vedere *report riassuntivo del test*.

report di avanzamento del test: Un documento, prodotto ad intervalli regolari, che riassume le attività di testing ed i relativi risultati, volto a riportare l'avanzamento delle attività di testing in linea con una baseline (come ad esempio il piano di test originale) ed a comunicare al management i rischi identificati e le azioni alternative suggerite richiedenti una decisione.

report di assessment: Un documento che riporta i risultati di un'attività di assessment, evidenze, conclusioni e raccomandazioni. Vedere *processo di assessment*

report di incidente del test: Vedere *report di incidente*.

report di incidente di test del software: Vedere *report degli incidenti*.

report di trasferimento: Vedere *note di rilascio*.

report di trasferimento del test: Vedere *note di rilascio*.

report di valutazione dei test: Un documento prodotto alla fine del processo di test che riassume tutte le attività di testing ed i risultati. Esso contiene anche una valutazione del processo di test e di conoscenza acquisita.

report riassuntivo del test: Un documento che riassume le attività di testing e i risultati. Esso contiene anche una valutazione dei corrispondenti elementi di test items rispetto ai criteri di uscita. [In accordo con IEEE 829]

requisiti testabili: Il livello di chiarezza ed esaustività di un requisito, che consente la progettazione del testing, la redazione di test case e la loro esecuzione volta a dimostrare il soddisfacimento del requisito stesso. [In accordo con IEEE 610]

requisito: Una esigenza di un utente per risolvere un problema o per raggiungere un obiettivo, che deve essere esaudita o posseduta da un sistema (o da una sua componente) per soddisfare un contratto, uno standard, una specifica od altri documenti formalmente imposti. [In accordo con IEEE 610]

requisito del test: Vedere *condizione del test*.

requisito funzionale: Un requisito che specifica una funzionalità che un componente o sistema deve eseguire. [IEEE 610]

requisito non-funzionale: Un requisito che non è associato a funzionalità, ma ad altri attributi, come ad esempio affidabilità, efficienza, usabilità, mantenibilità e portabilità.

responsabile del test: La persona responsabile della gestione e progettuale delle attività di testing e delle risorse correlate e della valutazione di un oggetto del test. L'individuo che dirige, controlla, amministra, pianifica e regola la valutazione di un oggetto del test.

re-testing: Testing che esegue i test case che non sono stati passati durante l'ultima esecuzione, in modo da controllare il successo delle azioni correttive.

retrospettiva progettuale: Una modalità strutturata per acquisire esperienza e per definire specifici piani d'azione volti a migliorare il prossimo progetto od una sua prossima fase.

revisione: Una valutazione di un prodotto o stato di un progetto per constatare discrepanze rispetto ai risultati pianificati e per raccomandare miglioramenti. Esempi includono la revisione della gestione, la revisione informale, la revisione tecnica, l'ispezione e il walkthrough. [In accordo con IEEE 1028]

revisione ad hoc: Vedere *revisione informale*.

revisione di testabilità: Una verifica dettagliata della base di test per determinare se essa ha un adeguato livello di qualità per agire come documento di input nel processo di test. [In accordo con TMap]

revisione formale: Una revisione caratterizzata da requisiti e procedure documentati, ad esempio per condurre un'ispezione.

revisione informale: Una revisione non basata su una procedura formale (documentata).

revisione manageriale: Una valutazione sistematica dei processi di acquisizione, fornitura, sviluppo, manutenzione o erogazione SW, svolto da (o per conto di) una funzione manageriale, che:

- conferma i requisiti del sistema,
- controlla lo stato di avanzamento del processo,

- determina lo stato dei piani e delle schedulazioni,
- valuta l'efficacia dell'operato rispetto agli obiettivi del processo stesso.

[In accordo con IEEE 610, IEEE 1028]

revisione tecnica: Una attività di discussione in un gruppo di colleghi che si concentra sul raggiungimento del consenso per l'approccio tecnico che deve essere seguito. [Gilb e Graham, IEEE 1028] Vedere anche *peer review*.

revisore: La persona coinvolta nella revisione, che identifica e descrive le anomalie nel prodotto o nel progetto sotto revisione. I revisori possono essere scelti per rappresentare differenti punti di vista e ruoli nel processo di revisione.

rilascio: Ogni prodotto (o semi-lavorato) che deve essere consegnato ad un soggetto diverso dall'autore del prodotto stesso.

riproducibilità del test: Un attributo di un test che indica se vengono prodotti gli stessi risultati ogni volta che il test viene eseguito.

rischio: Un fattore che potrebbe avere negative conseguenze future; generalmente espresso come impatto e probabilità.

rischio di prodotto: Un rischio collegato direttamente all'oggetto del test. Vedere anche *rischio*.

rischio di progetto: Rischio correlato alla gestione ed al controllo del progetto (di test), come ad esempio mancanza di personale, scadenze stringenti, modifiche ai requisiti, etc. Vedere anche *rischio*.

risultato: L'esito dell'esecuzione di un test. Esso include gli output, le modifiche ai dati, i report ed i messaggi di comunicazione esterni. Vedere anche *risultato attuale*, *risultato atteso*.

risultato atteso: Il comportamento previsto dalle specifiche, o da altra fonte, del componente o sistema sotto specificate condizioni.

risultato attuale: Il comportamento espresso od osservato quando un sistema od un suo componente viene testato.

risultato del test: Vedere *risultato*.

risultato falso fallimento: Un risultato di un test nel quale viene riportato un difetto nonostante non esista nessun difetto nell'oggetto del test.

risultato falso negativo: Vedere *risultato falso successo*.

risultato falso positivo: Vedere *risultato falso fallimento*.

risultato falso successo: Un risultato di un test nel quale il test non riesce ad identificare la presenza di un difetto che è presente nell'oggetto del test.

riunione di fine progetto: Vedere *riunione retrospettiva*

riunione retrospettiva: Una riunione alla fine del progetto durante il quale i membri del team valutano lo svolgimento ed i risultati del progetto e ne deducono degli insegnamenti che possono essere applicati ai progetti successivi.

robustezza: Il grado col quale un componente o sistema può funzionare correttamente in presenza di input invalidi o condizioni ambientali di stress. [IEEE 610] Vedere anche *tolleranza agli errori*, *tolleranza ai guasti*.

RUP: Vedere *Rational Unified Process*.

10.17. S

safety : La capacità del prodotto software di raggiungere dei livelli di rischio accettabili per quanto riguarda la prevenzione del danno a persone, al business, al software, alle proprietà o all'ambiente in uno specifico contesto di utilizzo. [ISO 9126]

sanity test: Vedere *smoke test*.

sbaglio: Vedere *errore*.

scala di misurazione: Una scala che vincola il tipo di analisi dei dati che possono essere eseguiti. [ISO 14598]

scalabilità: La capacità del prodotto software di essere esteso per adattarsi a carichi crescenti. [In accordo con Gerrard]

scenario di test: Vedere *specifica della procedura di test*.

schedulazione del test: Una lista di attività, compiti od eventi del processo di test, che identifica le loro date e/o tempi di inizio e fine previsti e le loro interdipendenze.

schema di riferimento: Una struttura di modello in cui il raggiungimento degli obiettivi di un insieme di aree di processo stabilisce un livello di maturità; ogni livello costruisce un fondamento per il livelli successivi. [CMMi]

scorecard: Vedere *balanced scorecard, cruscotto aziendale, dashboard*.

scostamento: Vedere *incidente*.

script di test: Una specifica della procedura di test, frequentemente automatizzata.

SCRUM: Una metodologia incrementale ed iterativa per la gestione di progetti comunemente utilizzata nello sviluppo software "agile". Vedere anche *sviluppo software agile*.

sessione di test: Un periodo di tempo ininterrotto trascorso nell'esecuzione di test. Nel testing esplorativo, ogni sessione di test è focalizzata su una dichiarazione di test, ma i tester possono anche esplorare nuove opportunità durante una sessione. Il tester crea ed esegue i test case al volo e registra il loro avanzamento. Vedere anche *testing esplorativo*.

severità: Il grado di impatto che un difetto ha sullo sviluppo o l'operazione di un componente o sistema. [In accordo con IEEE 610]

sicurezza: Attributo di prodotti software che si riferisce alla sua capacità di prevenire accessi non autorizzati, sia accidentali che intenzionali, a programmi e dati. [ISO 9126] Vedere anche *funzionalità*.

simulatore: Un dispositivo, programma o sistema usato durante il testing, il quale si comporta od opera come un specificato sistema quando stimolato con un insieme di input controllati. [In accordo con IEEE 610, DO178b] Vedere anche *emulatore*.

simulazione: La rappresentazione delle caratteristiche comportamentali selezionate di un sistema astratto o fisico da parte di un altro sistema. [ISO 2382/1]

sistema: Una collezione di componenti organizzata per ottenere una specifica funzione o insieme di funzioni. [IEEE 610]

sistema di sistemi: Sistemi multipli, eterogenei e distribuiti, che sono inseriti in reti a livelli multipli ed in domini multipli interconnessi per affrontare problemi (e/o perseguire obiettivi) comuni ed interdisciplinari su larga scala, di solito senza una struttura di gestione comune.

sistema safety critical: Un sistema nel quale un esito negativo od un malfunzionamento può avere come conseguenza la morte o seri danni a persone, perdita o danni severi ad equipaggiamenti o danni ambientali.

situazione di test: Vedere *condizione di test*.

smoke test: Un sottoinsieme di tutti i test case definiti/pianificati che coprono la funzionalità principale di un componente o sistema, per accertarsi che le funzioni più cruciali di un sistema funzionino, ma senza preoccuparsi dei dettagli. La build giornaliera unita allo smoke test è tra le best practice industriali. Vedere anche *intake test*.

software: Programmi, procedure, possibilmente con associata documentazione e dati pertinenti all'operazione del sistema informatico. [IEEE 610]

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

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

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

Software Fault Tree Analysis (SFTA): Vedere *Fault Tree Analysis (FTA)*.

software off-the-shelf: Un prodotto software che viene sviluppato per il mercato, quindi per un elevato numero di clienti e che può essere consegnato a più clienti nello stesso identico formato.

software personalizzato: Software sviluppato specificatamente per un insieme di utenti o clienti. È l'opposto di software off-the-shelf.

Software Process Improvement: Un programma di attività volte a migliorare le prestazioni e la maturità dei processi software di una organizzazione ed i risultati di tale programma. [In accordo con CMMi]

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

Software Usability Measurement Inventory: Vedere *SUMI* [Veenendaal]

sostituibilità: La capacità del prodotto software di essere usato al posto di un altro prodotto software specificato, per lo stesso scopo e nello stesso ambiente. [ISO 9126] Vedere anche *portabilità*.

sottocammino: Una sequenza di istruzioni eseguibili in un componente.

specificata: Un documento che specifica (auspicabilmente in un maniera completa, precisa e verificabile) i requisiti, la progettazione, il comportamento o altre caratteristiche di un componente o sistema e, spesso, le relative procedure, per determinare se queste clausole sono state soddisfatte. [In accordo con IEEE 610]

specificata dei test: Un documento che include una specifica di progettazione dei test, una specifica dei test case e/o una specifica della procedura di test.

specificata della procedura di test: Un documento che specifica una sequenza di azioni per l'esecuzione di un test. È noto anche come script di test o script di test manuale. [In accordo con IEEE 829]

specificata di componente: Una descrizione di un componente funzionale esprimente i suoi valori di output per predeterminati valori di input sotto specifiche condizioni, e di comportamento non-funzionali richiesti. (ad esempio, utilizzazione di risorse).

specifica di progettazione dei test: Un documento che specifica le condizioni di test (gli elementi di copertura) per un elemento di test, l'approccio di test dettagliato e l'identificazione dei test case di alto livello associati. [In accordo con IEEE 829]

specifica di test case: Un documento che specifica un insieme di test case (obiettivo, input, azioni del test, risultati attesi e precondizioni di esecuzione) per un elemento di test. [In accordo con IEEE 829]

SPI: Vedere *Software Process Improvement*.

stabilità: La capacità del prodotto software di evitare effetti inattesi a fronte di modifiche nel software. [ISO 9126] Vedere anche *mantenibilità*.

stadio di test: Vedere *livello di test*.

standard: Un insieme di requisiti, formali ed a volte obbligatori, sviluppati e utilizzati per prescrivere approcci coerenti al modo di lavorare o per fornire linee guida (ad esempio, norme ISO / IEC, standard IEEE e standard organizzativi). [In accordo con CMMi]

STEP: Vedere *Systematic Test and Evaluation Process*.

stima del test: L'approssimazione calcolata di un risultato relativo ai vari aspetti del testing (ad esempio sforzo speso, data di completamento, costi sostenuti, numero di test case etc.), che è utilizzabile anche se i dati di input possono essere incompleti, incerti o instabili.

stimolato: Una componente di un programma (ad esempio un'istruzione, una decisione o altri elementi strutturali) si dice essere stimolata da un test case quando un valore di input causa la sua esecuzione.

strategia di test: Una descrizione dei livelli di test che devono essere eseguiti e delle attività di testing da svolgere per ogni livello nell'ambito di una organizzazione o di uno o più progetti.

strumentazione di programma: Vedere *strumentazione*.

strumento di test di carico: Vedere *strumento di test di prestazioni*.

strumento di test di prestazioni: Uno strumento per supportare il test delle prestazioni che di solito ha due principali componenti: la generazione di carico e la misurazione di transazionale. La generazione del carico è in grado di simulare più utenti o elevati volumi di dati in ingresso. Durante l'esecuzione, si rileva e si registra il tempo di risposta per un certo numero di operazioni selezionate. Tali strumenti di test forniscono normalmente rapporti basati sulle registrazioni dei test e grafici di carico verso i tempi di risposta.

strumento di test ipertestuale: Uno strumento utilizzato per verificare che in un sito web non vi siano falsi collegamenti ipertestuali.

stub: Una implementazione semplificata (o con ridotti obiettivi) di un componente software, usata per sviluppare o testare un componente chiamante o che è comunque dipendente da esso. Lo stub sostituisce il componente chiamato. [In accordo con IEEE 610]

SUMI (Software Usability Measurement Inventory): Una tecnica di test di usabilità, basata su questionari, per la misurazione della qualità del software da parte dell'utente finale.

superamento: Un test è considerato superato se il suo risulta attuale corrisponde al suo risultato atteso.

superamento del test: Vedere *superamento*.

sviluppo guidato dal test: Una modalità di sviluppare software nel quale i test case sono progettati, e spesso automatizzati, prima che il software stesso sia sviluppato.

sviluppo software agile: un gruppo di metodologie di sviluppo software basate sullo sviluppo incrementale iterativo, in cui i requisiti e le soluzioni evolvono tramite una stretta collaborazione tra team interfunzionali che si auto-organizzano.

Systematic Test and Evaluation Process: Un metodo di analisi strutturato, usato anche come un modello content-based per migliorare il processo di test. STEP non richiede che i miglioramenti avvengano in un ordine specifico. Vedere anche *modello basato sui contenuti*.

10.18. T

tabella delle decisioni: Una tabella che evidenzia tutte le combinazioni di input e/o di stimoli (cause) con i relativi output e/o azioni (effetti) previsti; essa può essere usata per progettare test case.

tabella delle decisioni causa-effetto: Vedere *tabella delle decisioni*.

tabella di stato: Una tabella che mostra le transizioni risultanti dalla combinazione di ogni singolo stato con ogni singolo possibile evento, evidenziante sia le transizioni valide che quelle invalide.

tasso di esiti negativi: Il rapporto del numero di esiti negativi di una certa categoria rispetto ad una data unità di misura; ad esempio, esiti negativi per unità di tempo, esiti negativi per numero di transazioni, esiti negativi per numero di calcolatori attivi. [IEEE 610]

tassonomia dei banchi: Vedere *tassonomia dei difetti*.

tassonomia dei difetti: Un sistema di strutturazione (gerarchico) dei difetti che ne facilita la classificazione in modo riproducibile.

tecnica basata sui difetti: Vedere *tecnica di progettazione basata sui difetti*.

tecnica basata sull'esperienza: Vedere *tecnica di progettazione di test basata sull'esperienza*.

tecnica basata sulla struttura: Vedere *tecnica di progettazione di test white-box*.

tecnica basata sulle specifiche: Vedere *tecnica di progettazione di test black-box*.

tecnica black-box: Vedere *tecnica di progettazione black-box*.

tecnica di esecuzione dei test: Il metodo usato per svolgere l'attuale esecuzione dei test, sia manuale che automatico.

tecnica di progettazione basata sui difetti: Una procedura per creare e/o selezionare i test case sulla base di una o più categorie di difetti, in cui i test siano progettati da ciò che è noto relativamente alla specifica categoria di difetti.

tecnica di progettazione dei test: Procedura usata per derivare e/o selezionare i test case.

tecnica di progettazione di test basata sulla struttura: Vedere *tecnica di progettazione di test white-box*.

tecnica di progettazione di test basata sull'esperienza: Procedura per derivare e/o selezionare test case basati sull'esperienza, la conoscenza e l'intuizione del tester.

tecnica di progettazione di test basata sulle specifiche: Vedere *tecnica di progettazione di test black-box*.

tecnica di progettazione di test black-box: Procedura per derivare e/o scegliere i test case basata su una analisi delle specifiche, sia funzionale che non-funzionale, senza nessun riferimento alla struttura interna del componente o del sistema.

tecnica di progettazione di test case: Vedere *tecnica di progettazione di test*.

tecnica di progettazione di test funzionali: Procedura per derivare e/o selezionare test case sulla base di un'analisi delle specifiche della funzionalità di un componente o sistema senza nessun riferimento alla sua struttura interna. Vedere anche *tecnica di progettazione di test black-box*.

tecnica di progettazione di test strutturali: Vedere *tecnica di progettazione di test white-box*.

tecnica di progettazione di test white-box: Procedura per derivare e/o selezionare i test case sulla base di un'analisi della struttura interna di un componente o sistema.

tecnica di specifica dei test: Vedere *tecnica di progettazione dei test*.

tecnica di test: Vedere *tecnica di progettazione dei test*.

tecniche di progettazione di test non-funzionali: Procedure per derivare e/o selezionare dei test case di testing non-funzionale, basate su una analisi delle specifiche di un componente o sistema senza riferimento alla sua struttura interna. Vedere anche *tecniche di progettazione di test black-box*.

tecniche white-box: Vedere *tecniche di progettazione di test white-box*.

test: Un insieme di uno o più test case. [IEEE 829]

test bed: Vedere *ambiente di test*.

test case: Un insieme di valori di input, precondizioni di esecuzione, risultati attesi e postcondizioni di esecuzione, sviluppate per un particolare obiettivo o per una condizione di test, come ad esempio per attivare un particolare cammino del programma o per verificare la conformità con uno specifico requisito. [In accordo con IEEE 610]

test case astratto: Vedere *test case di alto livello*.

test case bloccato: Un test case che non può essere eseguito in quanto le precondizioni per la sua esecuzione non sono soddisfatte.

test case concreto: Vedere *test case di basso livello*.

test case di alto livello: Un test case senza dei concreti valori di dati di input e di risultati attesi (ovvero senza un livello di implementazione). Vengono usati operatori logici; a questo livello le istanze dei valori effettivi, non sono ancora state definite e/o disponibili. Vedere anche *test case di basso livello*.

test case di basso livello: Un test case con concreti valori dei dati di input e dei risultati attesi. Gli operatori logici provenienti dai test case di alto livello sono sostituiti dai valori effettivi, corrispondenti agli obiettivi degli operatori logici. Vedere anche *test case di alto livello*.

test case logico: Vedere *test case di alto livello*.

test del ciclo di processo: Una tecnica di progettazione di test black-box nella quale i test case sono progettati per eseguire le procedure di business ed i relativi processi. [TMap] Vedere anche *testing di procedure*.

test deliverable: Ogni prodotto (o semilavorato) del test che deve essere consegnato ad altri.

test dell'utente: Un test in base a cui gli utenti della vita reale sono coinvolti nella valutazione dell'usabilità di un componente o sistema.

test di algoritmo [TMap]: Vedere *testing dei rami*.

test di copertura: Vedere *copertura*.

test di idoneità: Il processo di testing per determinare l'idoneità di un prodotto SW.

test di riferimento: (1) Uno standard nei confronti del quale possono essere svolte misurazioni o confronti. (2) Un test che viene usato per confrontare reciprocamente componenti o sistemi o confrontarli rispetto ad uno standard, come in (1). [In accordo con IEEE 610]

test harness : Un ambiente di test compreso di stub e driver necessari per eseguire un test.

test leader: Vedere *responsabile del test*.

test log: Una registrazione cronologica dei dettagli rilevanti relativi all'esecuzione dei test. [IEEE 829]

test logging: Il processo di registrazione di informazioni sui test eseguiti all'interno di un test log.

Test Maturity Model (TMM): Un modello a cinque livelli per il miglioramento del processo di test, legato al Capability Maturity Model (CMM), che descrive gli elementi chiave di un processo di test efficace.

Test Maturity Model Integrated (TMMi): Un modello a cinque livelli per il miglioramento del processo di test, legato al Capability Maturity Model Integration (CMMi), che descrive gli elementi chiave di un processo di test efficace.

Test Point Analysis (TPA): [TMap] Una metodologia di stima delle attività di test, basata sull'analisi dei Function Point.

Test Process Group: Un insieme di specialisti (di test) che favoriscono la definizione, la manutenzione ed il miglioramento dei processi di test utilizzati dall'organizzazione [CMMi]

Test Process Improvement (TPI): Uno schema di miglioramento che descrive gli elementi chiave per un processo di test efficace, indirizzato principalmente al testing di sistema e di accettazione.

Test Process Improvement manifesto: Una dichiarazione derivata dal manifesto "agile", che definisce i valori per migliorare il processo di test. I valori sono:

- Flessibilità dei processi
- Best Practices vs modelli
- Orientamento di distribuzione vs. processo di orientamento
- Peer Reviews vs Quality Assurance
- Business Driven vs Model Driven. [Veenendaal08]

Test Process Improver: La persona che realizza i miglioramenti nel processo di test basato su un piano di miglioramento del testing.

test rig: Vedere *ambiente di test*.

testabilità: La capacità del prodotto software (realizzato o modificato) di essere testato. [ISO 9126]
Vedere anche *mantenibilità*.

tester: Un professionista con le qualità/capacità necessarie per eseguire il testing di un componente o sistema.

testing: Il processo che include tutte le attività, sia statiche che dinamiche, del ciclo di vita, riguardanti la pianificazione, preparazione e valutazione di prodotti software e dei prodotti di lavoro correlati, per

evidenziare che i prodotti software soddisfano i requisiti specificati, per dimostrare che essi sono adatti al loro scopo e per rilevare gli eventuali difetti.

testing a sessioni: Un approccio al testing in cui sono pianificate le attività di test come sessioni ininterrotte di progettazione ed esecuzione dei test, spesso utilizzato in combinazione con il testing sperimentale.

testing a stati finiti: Vedere *testing delle transizioni di stato*.

testing ad hoc: Testing condotto informalmente; non si preparano test formali, nessuna tecnica di progettazione riconosciuta viene usata, non ci sono aspettative per i risultati e la attività di esecuzione dei test è condotta in modo soggettivo.

testing back-to-back : Testing nel quale due o più varianti di un componente o sistema sono eseguite con gli stessi input, in cui quindi gli output vengono confrontati ed analizzati in casi di discrepanze. [IEEE 610]

testing basato su checklist: Una tecnica prova di disegno dei test basata sulla esperienza, in base alla quale l'esperto tester utilizza un elenco (di alto livello) degli elementi da osservare, controllare o ricordare, od un insieme di regole o criteri in base ai quali un prodotto SW deve essere verificato. Vedere anche *testing basato sull'esperienza*.

testing basato sui processi di business: Un approccio al testing nel quale i test case sono progettati basandosi sulle descrizioni e/o sulla conoscenza di processi di business.

testing basato sui requisiti: Un approccio al testing nel quale i test case sono progettati sulla base delle finalità di test e di condizioni di test derivate dai requisiti, come ad esempio i test che attivano specifiche funzioni o esaminano attributi non-funzionali come affidabilità o usabilità.

testing basato sul codice: Vedere *testing white-box*.

testing basato sul rischio: Un approccio al testing per ridurre il livello di rischio ed informare gli stakeholders sui loro stati, partendo dallo stadio iniziale di un progetto. Esso coinvolge l'identificazione dei rischi di prodotto e l'uso dei livelli di rischio nella conduzione del processo di test.

testing basato sulla progettazione: Un approccio al testing nel quale i test case sono progettati sulla base dell'architettura e/o della progettazione di dettaglio di un componente o sistema (ad esempio, i test delle interfacce tra componenti o sistemi).

testing basato sulla struttura: Vedere *testing white-box*.

testing basato sulle specifiche: Vedere *testing black-box*.

testing big-bang: Una tipologia di testing di integrazione nella quale gli elementi software, gli elementi hardware, oppure entrambi, vengono utilizzati assieme all'interno di un componente o del sistema complessivo. [In accordo con IEEE 610] Vedere anche *testing di integrazione*.

testing black-box: Testing, sia funzionale che non-funzionale, senza nessun riferimento alla struttura interna del componente o del sistema.

testing bottom-up: Un approccio incrementale al testing di integrazione nel quale vengono testati prima i componenti di livello più basso per facilitare poi il testing dei componenti di più livello più alto. Questo processo viene ripetuto finché il componente in cima alla gerarchia viene testato. Vedere anche *testing di integrazione*.

testing casuale: Una tecnica di progettazione di test black-box nella quale i test case vengono selezionati, possibilmente usando un algoritmo di generazione pseudo-casuale, per corrispondere ad un profilo

operativo. Questa tecnica può essere usata per il testing di attributi non-funzionali, come ad esempio affidabilità e prestazioni.

testing clear-box: Vedere *testing white-box*

testing combinatorio: Una tecnica di progettazione di test black-box nella quale i test case sono progettati per eseguire tutte le possibili combinazioni discrete di ogni singola coppia di parametri. Vedere anche *testing di matrici ortogonali*.

testing completo: Vedere *testing esaustivo*.

testing concorrente: Testing atto a determinare in che modo il verificarsi di due o più attività nello stesso intervallo di tempo, ottenuto interallacciando le attività o con esecuzioni simultanee, viene gestito dal componente o sistema. [In accordo con IEEE 610]

testing confermativo: Vedere *re-testing*.

testing degli scenari: Vedere *testing dei casi d'uso*.

testing dei cammini: Una tecnica di progettazione di test white-box nella quale i test case sono progettati per eseguire cammini.

testing dei casi d'uso: Una tecnica di progettazione di test black-box nella quale i test case sono progettati per eseguire gli scenari dei casi d'uso.

testing dei rami: Una tecnica di progettazione white-box nella quale i test case sono progettati per stimolare l'esecuzione dei rami.

testing dei valori limite: Vedere *analisi ai valori limite*.

testing del flusso dati: Una tecnica di progettazione di test white-box nella quale i test case sono progettati per attivare variabili per le quali è stato specificato l'abbinamento definizione/utilizzo.

testing della combinazione di condizioni: Vedere *testing delle condizioni multiple*.

testing della documentazione: Testing della qualità della documentazione; ad esempio, di un manuale utente o di una guida di installazione.

testing della tabella delle decisioni: Una tecnica di progettazione di test black-box nella quale i test case sono progettati per eseguire le diverse combinazioni di input e/o stimoli (cause) evidenziate in una tabella delle decisioni. [Veenendaal] Vedere *anche tabella delle decisioni*.

testing delle combinazioni delle condizioni dei rami : Vedere *testing delle condizioni multiple*.

testing delle condizioni: Una tecnica di progettazione di test white-box nella quale i test case vengono progettati per attivare gli esiti delle condizioni.

testing delle condizioni multiple: Una tecnica di progettazione di test white-box nella quale i test case sono progettati per eseguire combinazioni di singoli esiti condizionali (nell'ambito di una singola istruzione).

testing delle condizioni multiple modificate: Vedere *testing di determinazione delle condizioni*.

testing delle decisioni: Una tecnica di progettazione di test white-box nella quale i test case sono progettati per eseguire gli esiti delle decisioni.

testing delle decisioni condizionate: Una tecnica di progettazione di test white-box nella quale i test case sono progettati per eseguire gli esiti delle condizioni e gli esiti delle decisioni.

testing delle decisioni in condizioni modificate: Vedere *testing di determinazione delle condizioni*.

- testing delle istruzioni:** Una tecnica di progettazione di test white-box test nella quale i test case sono progettati per eseguire istruzioni.
- testing delle mutazioni:** Vedere *testing back-to-back*.
- testing delle partizioni:** Vedere *partizionamento di equivalenza*. [Beizer]
- testing delle prestazioni:** Il processo di testing per determinare le prestazioni di un prodotto software. Vedere anche *testing di efficienza*.
- testing delle transizioni di stato:** Una tecnica di progettazione di test black-box test nella quale i test case sono progettati per eseguire transizioni di stato valide ed invalide. Vedere anche *testing di N-commutazioni*.
- testing di accessibilità:** Testing atto a determinare la facilità attraverso la quale utenti con disabilità possono utilizzare un sistema od un suo componente . [Gerrard]
- testing di accettazione:** Testing formale, basato sulle esigenze dell'utente, sui requisiti e sui processi di business, condotto per determinare se un sistema soddisfa o meno i criteri di accettazione e per consentire all'utente, al cliente o ad un'altra entità autorizzata di decidere se accettare oppure non accettare il sistema. [In accordo con IEEE 610]
- testing di accettazione in produzione:** Vedere *testing di accettazione operativo*.
- testing di accettazione on-site:** Testing di accettazione da parte di utenti/clienti presso il loro sito, per determinare se un componente o sistema soddisfa o meno le esigenze utente/cliente ed è adeguato ai processi di business.
- testing di accettazione operativo:** Testing condotto nella fase di test di accettazione, normalmente eseguito in un ambiente operativo simulato (simile a quello reale) dall'operatore e/o amministratore concentrandosi su aspetti operativi, come ad esempio la ripristinabilità, l'impegno delle risorse, l'installabilità e la conformità tecnica. Vedere anche *testing operativo*.
- testing di accettazione utente:** Vedere *testing di accettazione*.
- testing di accuratezza:** Il processo di testing per determinare l'accuratezza di un prodotto SW.
- testing di affidabilità:** Il processo di testing per determinare la affidabilità di un prodotto software.
- testing di carico:** Un tipo di testing prestazionale condotto per valutare il comportamento di un componente o sistema sottoposto a carico crescente, come ad esempio il numero di utenti in parallelo e/o il numero delle transazioni, per determinare quale carico massimo può essere gestito dal componente o sistema stesso. Vedere anche *testing delle prestazioni, testing di stress*.
- testing di collegamento:** Vedere *testing di integrazione di componenti*.
- testing di compatibilità:** Vedere *testing di interoperabilità*.
- testing di componente:** Il testing dei singoli componenti software. [In accordo con IEEE 610]
- testing di configurazione:** Vedere *testing di portabilità*.
- testing di conformità:** Il processo di testing per determinare la conformità del componente o sistema.
- testing di conversione:** Testing del software usato per convertire dati da sistemi esistenti a sistemi di rimpiazzo.
- testing di copertura logica:** Vedere *testing white-box*. [Myers]

testing di determinazione delle condizioni: Una tecnica di progettazione di test white-box nella quale i test case vengono progettati per eseguire singoli esiti di una condizione che indipendentemente interessano una l'esito di una decisione.

testing di efficienza: Il processo di testing per determinare l'efficienza di un prodotto software.

testing di funzionalità: Il processo di testing per determinare la funzionalità di un prodotto software.

testing di gestibilità: Vedere *testing di mantenibilità*.

testing di idoneità: Il processo di testing per determinare l'idoneità di un prodotto software

testing di installabilità: Il processo di testing dell'installabilità di un prodotto software. Vedere anche *testing di portabilità*.

testing di integrazione: Testing eseguito per rilevare difetti nelle interfacce e nelle interazioni tra i componenti o sistemi integrati. Vedere anche *testing di integrazione di componenti*, *testing di integrazione di sistemi*.

testing di integrazione di componenti: Testing eseguito per identificare eventuali difetti nelle interfacce e nelle interazioni tra i componenti integrati.

testing di integrazione di sistemi: Testing della integrazione di sistemi e componenti, nonché delle interfacce verso organizzazioni esterne (ad esempio, Electronic Data Interchange, Internet).

testing di integrazione in grande: Vedere *testing di integrazione di sistemi*.

testing di integrazione in piccolo: Vedere *testing di integrazione di componenti*.

testing di integrità dei dati: Vedere *testing di integrità del database*.

testing di integrità del database: Testing dei metodi e dei processi usati per accedere e gestire il database, per assicurare che i metodi di accesso, i processi e le regole sui dati funzionino come atteso e che, durante l'accesso al database, i dati non siano corrotti o cancellati, modificati o creati in modo inatteso.

testing di interfaccia: Un tipo di test di integrazione che riguarda il testing delle sole interface tra componenti o sistemi.

testing di interoperabilità: Il processo di testing per determinare l'interoperabilità di un prodotto software. Vedere anche *testing di funzionalità*.

testing di mantenibilità: Il processo di testing per determinare la mantenibilità di un prodotto software.

testing di manutenzione: Testing delle modifiche ad un sistema in produzione o dell'impatto di un ambiente modificato ad un sistema in produzione.

testing di matrici ortogonali: Un modo sistematico di testare tutte le coppie di combinazioni di variabili, usando delle matrici ortogonali. Riduce significativamente il numero di combinazioni di variabili per testare tutte le coppie di combinazioni. Vedere anche *testing combinatorio*.

testing di memorizzazione: Vedere *testing di utilizzazione delle risorse*.

testing di migrazione: Vedere *testing di conversione*.

testing di modulo: Vedere *testing di componente*.

testing di N-commutazioni: Una forma di testing delle transizioni di stato nella quale i test case sono progettati per eseguire tutte le sequenze valide di N+1 transizioni. [Chow] Vedere anche *testing delle transizioni di stato*.

- testing di portabilità:** Il processo di testing atto a determinare la portabilità di un prodotto software.
- testing di procedure:** Testing finalizzato ad assicurare che il componente o sistema può operare in congiunzione con procedure (nuove o esistenti) di business o procedure operative.
- testing di profilo operativo:** Testing campionario che usa un insieme di attività (di breve durata) derivate del profilo operativo del sistema [Musa]
- testing di programma:** Vedere *testing di componente*.
- testing di recuperabilità:** Il processo di testing per determinare la recuperabilità di un prodotto software. Vedere anche *testing di affidabilità*.
- testing di recupero:** Vedere *testing di recuperabilità*.
- testing di regressione:** Testing delle modifiche di un programma precedentemente testato, per assicurare che non siano stati introdotti o non scoperti dei difetti nelle aree non modificate del software, come risultato delle modifiche effettuate. Esso viene eseguito quando il software od il suo ambiente viene modificato.
- testing di robustezza:** Testing per determinare la robustezza del prodotto software.
- testing di safety:** Testing atto a determinare la safety di un prodotto software.
- testing di scalabilità:** Testing atto a determinare la scalabilità del prodotto software.
- testing di scenari utente:** Vedere *testing dei casi d'uso*.
- testing di sicurezza:** Testing atto a determinare la sicurezza del prodotto software. Vedere anche *testing di funzionalità*.
- testing di sistema:** Il processo di testing di un sistema integrato per verificare che esso soddisfi i suoi requisiti specificati. [Hetzel]
- testing di standard:** Vedere *testing di conformità*.
- testing di stress:** Un tipo di testing delle prestazioni condotto per valutare un sistema o componente ai limiti od oltre i limiti dei suoi carichi di lavoro (stimati o specificati) o condotto con una ridotta disponibilità di risorse, [In accordo con IEEE 610] Vedere anche *testing delle prestazioni, testing di carico*.
- testing di sviluppo:** Testing formale o informale condotto durante la realizzazione di un componente o sistema, generalmente nell'ambiente di sviluppo dagli sviluppatori. [In accordo con IEEE 610]
- testing di unità:** Vedere *testing di componente*.
- testing di usabilità:** Testing per determinare il grado con il quale il prodotto software viene compreso, risulta facile da imparare, facile da usare e di piacevole utilizzo per gli utenti sotto specifiche condizioni. [In accordo con ISO 9126]
- testing di utilizzazione delle risorse:** Il processo di testing per determinare l'utilizzazione delle risorse di un prodotto software. Vedere anche *testing di efficienza*.
- testing di volume:** Testing nel quale il sistema è alimentato con grandi volumi di dati. Vedere anche *testing di utilizzazione delle risorse*.
- testing dinamico:** Testing che comporta l'esecuzione del software di un componente o sistema.
- testing esaustivo:** Un approccio al test nel quale l'insieme di test comprende tutte le combinazioni di valori di input e di precondizioni.

testing esplorativo: Una tecnica informale di progettazione di test nella quale il tester controlla direttamente la progettazione dei test, come questi test sono eseguiti ed usa le informazioni che ha tratto durante il testing per progettare nuovi e migliori test. [In accordo con Bach]

testing funzionale: Testing basato su un'analisi delle specifiche della funzionalità di un componente o sistema. Vedere anche *testing black-box*.

testing guidato: Esecuzione di test condotta seguendo una sequenza di test precedentemente documentata.

testing guidato dai dati: Una tecnica di scripting che memorizza gli input dei test ed i risultati attesi in una tabella o in un database, in modo che un singolo script di controllo possa eseguire tutti i test nella tabella. Il testing guidato dai dati viene spesso usato per supportare l'applicazione di tool di esecuzione di test, come i tool di cattura/riesecuzione. [Fewster e Graham] Vedere anche *testing guidato dalle parole chiave*.

testing guidato dalla logica: Vedere *testing white-box*.

testing guidato dalle parole chiave: Una tecnica di scripting che usa dei file di dati per contenere non solo i dati di test ed i risultati attesi, ma anche le parole chiave legate all'applicazione di cui effettuare il testing. Le parole chiave sono interpretate da speciali script di supporto che sono chiamati dallo script di controllo per il test. Vedere anche *testing guidato dai dati*.

testing in isolamento: Testing di singoli componenti isolati dai componenti circostanti, nel quale i componenti circostanti vengono simulati, se necessario, da stub e driver.

testing incrementale: Testing nel quali i componenti o sistemi sono integrati e testati uno alla volta (od a gruppi), fino a che tutti i componenti o sistemi sono integrati e testati.

testing invalido: Testing che utilizza valori di input che dovrebbero essere rifiutati/rigettati dal componente o sistema. Vedere anche *testing negativo*, *tolleranza agli errori*.

testing LCSAJ: Una tecnica di progettazione di test white-box nella quale i test case sono progettati per eseguire dei LCSAJ.

testing negativo: Test mirati a dimostrare che un componente o sistema non funziona. Il testing negativo è legato all'attitudine del tester piuttosto che ad uno specifico approccio di test o tecnica di progettazione di test, come ad esempio il testing con valori di input invalidi o eccezioni. [In accordo con Beizer]

testing non-funzionale: Testing degli attributi di un componente/sistema che non sono legati a funzionalità, come ad esempio affidabilità, efficienza, usabilità, mantenibilità e portabilità.

testing operativo: Testing condotto per valutare un componente o sistema nel suo ambiente operativo. [IEEE 610]

testing parallelo: Una coppia di persone (ad esempio due tester, uno sviluppatore e un tester, oppure un utente finale ed un tester), che lavorano assieme per trovare difetti. Tipicamente, essi condividono un computer e prendono il controllo di esso durante il testing.

testing progressivo: Una versione del testing di integrazione di componenti nella quale la progressiva integrazione dei componenti è successiva all'implementazione di sottoinsiemi dei requisiti, in alternativa all'integrazione dei componenti per livelli di una gerarchia.

testing regolamentativo: Vedere *testing di conformità*.

testing sintattico: Una tecnica di progettazione di test black-box nella quale i test case sono progettati basandosi sulla definizione del dominio di input e/o di output.

testing statico: Testing di un componente o sistema a livello di specifica o di implementazione senza l'esecuzione di questo software, come ad esempio revisioni o analisi statica del codice.

testing statistico: Una tecnica di progettazione di test nella quale viene usato un modello di distribuzione statistica degli input per costruire dei test case rappresentativi. Vedere anche *testing di profilo operativo*.

testing strutturale: Vedere *testing white-box*.

testing sul campo: Vedere *beta testing*.

testing top-down: Un approccio incrementale al testing di integrazione, nel quale viene testato per primo il componente in cima alla gerarchia dei componenti, con i componenti al livello inferiore simulati da opportuni stub. I componenti testati sono quindi usati per testare i componenti del livello inferiore. Il processo viene ripetuto finchè tutti i componenti di livello più basso sono stati testati. Vedere anche *testing di integrazione*.

testing white-box: Testing basato su un'analisi della struttura interno del componente o sistema.

testware: Prodotti sviluppati durante il processo di test per pianificare, progettare ed eseguire test, come ad esempio documentazione, script, input, risultati attesi, procedure di configurazione e di ripristino, file, database, ambiente ed ogni software di utilità aggiuntivo usato durante il testing. [In accordo con Fewster e Graham]

testware automatico: Testware utilizzato nel testing automatico, come ad esempio scripts (o sequenza di comandi) di un tool.

tipo di rischio: Un insieme di rischi raggruppati per uno o più fattori comuni, come ad esempio un attributo di qualità, un loro causa, la loro posizione o l'effetto potenziale del rischio. Uno specifico insieme di tipi di rischio di prodotto viene correlato al tipo di test in grado di attenuare (o controllare) quel tipo di rischio. Per esempio il rischio che una interazione utente possa essere fraintesa può essere attenuato dal test di usabilità.

tipo di test: Un gruppo di attività di test che hanno la finalità di testare un componente o sistema, focalizzandosi su uno specifico obiettivo di test (test funzionale, test di usabilità, test di regressione etc). Un tipo di test può essere condotto nell'ambito di uno o più livelli di test o in una o più fasi. [In accordo con Tmap]

tolleranza agli errori: La capacità di un sistema o componente di continuare le normali operazioni anche in presenza di input errati. [In accordo con IEEE 610].

tolleranza ai guasti: La capacità del prodotto software di mantenere uno specificato livello di prestazioni in casi di guasti software (difetti) o di violazione delle sue specifiche interfacce. [ISO 9126] Vedere anche *affidabilità, robustezza*.

tool di analisi dinamica: Un tool che fornisce informazioni sullo stato del codice del software durante la sua esecuzione. Questi tool sono più comunemente usati per identificare puntatori non assegnati, controllare l'aritmetica dei puntatori e monitorare l'allocazione, l'utilizzo ed il rilascio della memoria e per segnalare insufficienze di memoria.

tool di analisi statica: Vedere *analizzatore statico*.

tool di cattura/riesecuzione: Vedere tool di cattura/riproduzione.

tool di cattura/riproduzione: Un tipo di tool di esecuzione di test nel quale gli input sono registrati durante il testing manuale con lo scopo di generare script di test automatici, che possono essere eseguiti successivamente (quindi rieseguiti). Questi tool sono spesso usati per supportare testing di regressione automatico.

tool di copertura: Un tool che fornisce misure oggettive di quali elementi strutturali, ad esempio istruzioni o rami, sono stati attivati da un insieme di test.

tool di debugging: Un tool usato dai programmatori per riprodurre esiti negativi, esaminare lo stato dei programmi e trovare i corrispondenti difetti. I tool di debugging consentono ai programmatori di eseguire i programmi passo per passo, per fermare un programma ad ogni sua istruzione e per impostare ed esaminare le variabili del programma.

tool di disseminazione degli errori: Vedere *tool di disseminazione dei guasti*.

tool di disseminazione dei guasti: Un tool per la disseminazione (e quindi l'inserimento intenzionale) di guasti in un componente o sistema.

tool di esecuzione dei test: Un tipo di tool di test che è in grado di eseguire altro software, usando uno script di test automatico, come ad esempio un tool di cattura/riproduzione. [Fewster e Graham]

tool di gestione degli incidenti: Un tool che supporta la registrazione ed il tracciamento dello stato degli incidenti. Essi hanno spesso delle funzionalità di workflow per tenere traccia e controllare la allocazione, la correzione ed il re-testing degli incidenti e la fornitura di report. Vedere anche *tool di gestione dei difetti*.

tool di gestione dei difetti: Un tool che agevola la registrazione ed il tracciamento dello stato dei difetti e delle conseguenti modifiche correttive. Questi tool spesso hanno delle funzionalità di workflow per tracciare e controllare l'assegnazione, la diagnosi, la correzione ed il re-testing dei difetti e delle funzionalità di reportistica. Vedere anche *tool di gestione degli incidenti*.

tool di gestione dei requisiti: Un tool che supporta la registrazione dei requisiti, degli attributi dei requisiti (ad esempio, priorità, responsabile del processo etc.) e di altre annotazioni, e che agevola la tracciabilità e la gestione delle modifiche dei requisiti attraverso le varie fasi del ciclo di sviluppo/modifica. Alcuni tool di gestione dei requisiti forniscono anche supporto per l'analisi statica, come ad esempio controlli di consistenza e violazioni a predefinite regole di specifica dei requisiti.

tool di gestione del test: Un tool che fornisce supporto alla gestione del test e controlla parte di un processo di test. Esso spesso offre diverse funzionalità come la gestione del testware, la pianificazione dell'esecuzione dei test, il logging dei risultati, il tracciamento dell'avanzamento, la gestione degli incidenti e la reportistica dei test.

tool di gestione della configurazione: Un tool che fornisce supporto per l'identificazione ed il controllo degli elementi della configurazione, dello stato delle loro modifiche e del rilascio di baseline includente elementi di configurazione.

tool di misura della copertura: Vedere *tool di copertura*.

tool di modellazione: Un tool che supporta la verifica di modelli di software o di sistemi. [Graham].

tool di monitoraggio: Vedere *monitor*.

tool di preparazione dei dati di test: Una tipologia di tool di test, che consente ai dati di essere creati o selezionati da database esistenti di essere manipolati ed editati per l'utilizzo nel testing.

tool di progettazione di test: Un tool che supporta l'attività di progettazione di test generando input di test a partire da:

- una specifica, che può essere gestita in un repository di un CASE tool, come ad esempio un tool di gestione dei requisiti,
- condizioni di test specificate nel tool stesso,
- dal codice sorgente.

tool di registrazione/riproduzione: Vedere *tool di cattura/riproduzione*.

tool di revisione: Un tool che fornisce supporto al processo di revisione. Caratteristiche tipiche includono la pianificazione della revisione e il supporto dei tracciamenti, il supporto della comunicazione, le revisioni collaborative ed un repository per la raccolta e la reportistica di metriche.

tool di sicurezza: Un tool che supporta la sicurezza operativa.

tool di test: Un prodotto software che supporta una o più attività di test, come la pianificazione ed il controllo, la specifica, la predisposizione dei file di input, la generazione dei dati, l'esecuzione e l'analisi dei test. [TMap] Vedere anche *CAST*.

tool di testing delle prestazioni: Un tool per supportare il testing delle prestazioni, che generalmente ha due funzionalità principali: la generazione del carico e la misurazione delle transazioni di test. La generazione di carico può simulare sia utenti multipli sia elevati volumi di dati di input. Durante l'esecuzione vengono rilevate le misure dei tempi di risposta dalle transazioni selezionate e tali misure vengono poi registrate. I tool di testing delle prestazioni normalmente forniscono dei report basati sulle registrazioni dei test e grafici del carico correlati ai tempi di risposta.

tool di testing di sicurezza: Un tool che fornisce supporto per definire le caratteristiche e le vulnerabilità del testing di sicurezza.

tool di testing di stress: Un tool che supporta il testing di stress.

tool di tracciamento dei banchi: Vedere *tool di gestione dei difetti*.

tool di tracciamento dei difetti: Vedere *tool di gestione dei difetti*.

TPG: Vedere *Test Process Group*

TQM: Vedere *Test Quality Management*

tracciabilità: La capacità di identificare elementi correlati nella documentazione e nel software, come ad esempio i requisiti con i test associati. Vedere anche *tracciabilità orizzontale*, *tracciabilità verticale*.

tracciabilità orizzontale: Il tracciamento dei requisiti di un certo livello di test attraverso gli strati della documentazione disponibile (ad esempio, piano di test, specifica di progettazione dei test, specifica dei test case, specifica della procedura di test o script di test).

tracciabilità verticale: Il tracciamento dei requisiti attraverso i livelli della documentazione di sviluppo verso i componenti.

transizione di stato: Una transizione tra due stati di un componente o sistema.

10.19. U

unità: Vedere *componente*.

usabilità: La capacità del software di essere compreso, imparato, utilizzato e di essere di piacevole utilizzo per l'utente quando usato sotto specifiche condizioni. [ISO 9126]

utilizzo delle risorse: La capacità del prodotto software di utilizzare (quando il software esegue le sue funzionalità sotto prefissate condizioni) le appropriate quantità e tipologia di risorse, per esempio l'impegno di memoria principale e secondaria, le dimensioni dei file temporanei o di overflow, etc. [In accordo con ISO 9126] Vedere anche *efficienza*.

10.20. V

validazione: Conferma per esame (e attraverso la fornitura di evidenze obiettive) che i requisiti per un uso specifico e predefinito o per una applicazione sono stati soddisfatti. [ISO 9000]

valore di input: Una istanza di un input.

valore di output: Una istanza di un output.

valore limite: Un valore di input o di output che è sul margine di una partizione di equivalenza o alla più piccola distanza incrementale su qualunque lato del margine; per esempio se la partizione di equivalenza è compresa fra 1 e 10 i valori limite sono : 1 = valore minimo, 0 = 1 -1 e 10 = valore massimo e 11 = 10+1.

valutazione: Vedere *testing*.

valutazione euristica: Una tecnica statica di test di usabilità per determinare la conformità di una interfaccia utente ai principi di usabilità comunemente accettati e riconosciuti (i cosiddetti "euristici").

variabile: Un elemento di memorizzazione in un computer che è accessibile da un programma software tramite il riferimento ad esso per un nome.

verifica: Conferma per esame (e attraverso la fornitura di evidenze obiettive) che specifici requisiti sono stati soddisfatti. [ISO 9000]

verificatore: Vedere *revisore*.

10.21. W

walkthrough : Una presentazione passo per passo dell'autore del contenuto di un documento con l'obiettivo di raccogliere suggerimenti di miglioramento e di pervenire ad una comprensione comune del suo contenuto. [Freedman e Weinberg, IEEE 1028] Vedere anche *peer review*.

walkthrough strutturato: Vedere *walkthrough*.

Wide Band Delphi : Una tecnica di stima basata sul contributo collettivo di un gruppo di esperti.

wizard di installazione: Software fornito su qualsiasi supporto adeguato, che conduce l'installatore durante il processo di installazione. Esso normalmente lancia il processo di installazione, fornisce dei riscontri sui risultati dell'installazione e assiste nella scelta delle opzioni disponibili.

11. ANNESSO A (INFORMATIVO)

Indice delle fonti; le seguenti fonti non-normative sono state usate nella costruzione di questo glossario:

- [Abbott] J. Abbot (1986), *Software Testing Techniques*, NCC Publications.
- [Adrion] W. Adrion, M. Branstad and J. Cherniabsky (1982), Validation, Verification and Testing of Computer Software, in: *Computing Surveys*, Vol. 14, No 2, June 1982.
- [Bach] J. Bach (2004), Exploratory Testing, in: E. van Veenendaal, *The Testing Practitioner – 2nd edition*, UTN Publishing, ISBN 90-72194-65-9.
- [Beizer] B. Beizer (1990), *Software Testing Techniques*, van Nostrand Reinhold, ISBN 0-442-20672-0
- [Chow] T. Chow (1978), Testing Software Design Modelled by Finite-Sate Machines, in: *IEEE Transactions on Software Engineering*, Vol. 4, No 3, May 1978.
- [CMM] M. Paulk, C. Weber, B. Curtis and M.B. Chrissis (1995), *The Capability Maturity Model, Guidelines for Improving the Software Process*, Addison-Wesley, ISBN 0-201-54664-7
- [CMMi] M.B. Chrissis, M. Konrad and S. Shrum (2004), *CMMi, Guidelines for Process Integration and Product Improvement*, Addison Wesley, ISBN 0-321-15496-7
- [Fenton] N. Fenton (1991), *Software Metrics: a Rigorous Approach*, Chapman & Hall, ISBN 0-53249-425-1
- [Fewster and Graham] M. Fewster and D. Graham (1999), *Software Test Automation, Effective use of test execution tools*, Addison-Wesley, ISBN 0-201-33140-3.
- [Freedman and Weinberg] D. Freedman and G. Weinberg (1990), *Walkthroughs, Inspections, and Technical Reviews*, Dorset House Publishing, ISBN 0-932633-19-6.
- [Gerrard] P. Gerrard and N. Thompson (2002), *Risk-Based E-Business Testing*, Artech House Publishers, ISBN 1-58053-314-0.
- [Gilb and Graham] T. Gilb and D. Graham (1993), *Software Inspection*, Addison-Wesley, ISBN 0-201-63181-4.
- [Graham] D. Graham, E. van Veenendaal, I. Evans and R. Black (2007), *Foundations of Software Testing*, Thomson Learning, ISBN 978-1-84480-355-2
- [Grochtmann] M. Grochtmann (1994), Test Case Design Using Classification Trees, in: *Conference Proceedings STAR 1994*.
- [Hetzel] W. Hetzel (1988), *The complete guide to software testing – 2nd edition*, QED Information Sciences, ISBN 0-89435-242-3.
- [McCabe] T. McCabe (1976), A complexity measure, in: *IEEE Transactions on Software Engineering*, Vol. 2, pp. 308-320.
- [Musa] J. Musa (1998), *Software Reliability Engineering Testing*, McGraw-Hill Education, ISBN 0-07913-271-5.
- [Myers] G. Myers (1979), *The Art of Software Testing*, Wiley, ISBN 0-471-04328-1.
- [TMap] M. Pol, R. Teunissen, E. van Veenendaal (2002), *Software Testing, A guide to the TMap Approach*, Addison Wesley, ISBN 0-201-745712.

[Veenendaal] E. van Veenendaal (2004), *The Testing Practitioner – 2nd edition*, UTN Publishing, ISBN 90-72194-65-9.

12. ANNESSO B (METODO DI COMMENTO AL GLOSSARIO ORIGINALE IN LINGUA INGLESE)

Commenti al presente documento sono graditi così che esso possa essere ulteriormente migliorato per soddisfare le esigenze della comunità di testing.

Nel fornire un commento, assicurarsi di includere le seguenti informazioni:

- Il vostro nome e riferimenti di contatto;
- Il numero di versione del glossario (attualmente 2.0);
- Parte esatta del glossario;
- Supportare le informazioni del commento, come la ragione per la proposta di modifica, o il riferimento all'uso di un termine.

Potete inviare commenti in diversi modi, i quali sono in ordine di preferenza i seguenti :

1. Per E-mail a: eve@improveqs.nl;
2. Per posta a: Improve Quality Services BV, attn. Mr. E. van Veenendaal, Waalreweg 39, 5554 HA, Valkenswaard, The Netherlands;
3. Per FAX a: +31 40 20 21450, all'attenzione di Mr. E. van Veenendaal.

13. ANNESSO C (METODO DI COMMENTO AL PRESENTE GLOSSARIO)

Come descritto nel §1 del documento di supporto e completamento a questo Glossario, ovvero il già citato :

“DOCUMENTO DI TRACCIABILITÀ dei termini tra GLOSSARIO ISTQB in LINGUA ORIGINALE e GLOSSARIO ITA-STQB tradotto in LINGUA ITALIANA”

la traduzione è stata effettuata senza seguire il mero approccio di tradurre tutti i termini in ogni caso, prescindendo magari addirittura a volte dalla semantica del termine da tradurre.

Invece l'approccio seguito è stato improntato a cercare di esaminare la semantica del termine, la sintassi e di cercare di trovare un compromesso tra le due che rendesse anche in Lingua Italiana il significato del termine inglese. Questo talvolta è stato molto naturale a volte molto meno ed in alcuni casi addirittura impossibile.

Tale approccio sicuramente è opinabile, ma ogni tentativo di traduzione di glossari di terminologia tecnica che si sviluppano prevalentemente in una lingua presta il fianco a commenti e opinioni diverse.

Quindi commenti e suggerimenti sono sicuramente graditi e la loro approvazione è sempre soggetta alla approvazione del Comitato Scientifico di ITA-STQB. Qualora approvate, le modifiche alla traduzione del presente Glossario, daranno luogo all'emissione di una nuova revisione del presente documento e del documento di supporto sopra citato, che sarà identificata con le modifiche nel Registro delle Revisioni di entrambi i documenti.

Per inviare commenti e suggerimenti si prega di mandare una e-mail ai seguenti indirizzi: info@ita-stqb.org