

# **Glossario standard dei termini utilizzati in Software Engineering**



**Versione 1.0**

Nota di Copyright

Questo documento può essere copiato nella sua interezza, o in parte, se la fonte è riconosciuta.

## Change History

Versione	Data	Note
1.0	01.02.2019	Prima versione italiana.

## Sommario

Change History .....	2
1. Introduzione .....	4
2. Ambito .....	4
3. Struttura .....	4
4. Riferimenti normativi .....	4
5. Marchi registrati .....	5
6. Definizioni.....	6
5.....	6
A.....	6
B.....	7
C.....	9
D .....	12
E.....	13
F.....	14
G .....	16
H .....	16
I.....	16
L .....	17
M .....	18
N .....	19
O .....	20
P.....	20
Q.....	22
R.....	22
S.....	25
T.....	28
U .....	28
V.....	29
W .....	30
7. Annex A (Informativa) .....	31

## 1. Introduzione

Lo scopo di questo documento è di fornire un glossario standardizzato utilizzabile da parte dei professionisti IT in Business Analysis e Ingegneria dei requisiti per garantire una comprensione comune dei termini e delle attività basilari.

## 2. Ambito

Questo documento presenta concetti, termini e definizioni relativi alla business e system analysis, all'ingegneria del software in generale e alle discipline correlate

## 3. Struttura

Il glossario è stato organizzato in un'unica sezione di definizioni ordinate alfabeticamente. Alcuni termini sono preferiti ad altri sinonimi, nel qual caso appare la definizione del termine preferito, con i sinonimi che si riferiscono a quel termine.

*Nota relativa alla versione italiana: ogni termine viene mantenuto nella versione inglese e a fianco, dove si ritiene applicabile perchè comunemente in uso, viene indicato il termine tradotto in italiano. Dove questa traduzione manca, si è ritenuto più comunemente utilizzato il termine originale inglese. Anche nel syllabus si è adottato questo criterio nella traduzione dei termini.*

## 4. Riferimenti normativi

Al momento della pubblicazione di questo documento, era valida l'edizione indicata. Tutti gli standard sono soggetti a revisione e le parti concordi all'utilizzo di questo documento sono incoraggiate a valutare la possibilità di applicare l'edizione più recente degli standard sotto elencati. A tale proposito si ricorda che i membri di IEC e ISO mantengono un registro che riporta gli standard internazionali attualmente 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 830: 1998: Recommended Practice for Software Requirements Specifications
- 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.
- IEEE 1233:1998: Guide for Developing of System Requirements Specifications
- IEEE 1362:1998: Guide for Information Technology – System Definition
- ISO/IEC 2382-1:1993. Data processing - Vocabulary - Part 1: Fundamental terms.
- ISO 9000:2005. Quality Management Systems – Fundamentals and Vocabulary.
- ISO/IEC 12207:1995. Information Technology – Software Lifecycle 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
- ISO/IEC 25000:2005 Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE
- ISO 31000: Risk Management - Principles and Guidelines on Implementation
- IEC 31010: Risk Management - Risk Assessment Techniques
- ISO/IEC 73: Risk Management – Vocabulary
- ISTQB Glossary of testing terms ver. 2.1

## 5. Marchi registrati

In questo documento sono citati i seguenti marchi registrati:

- CMM, CMMI e IDEAL sono marchi registrati di Carnegie Mellon University
- BABOK è un marchio registrato di Business Analysis Body of Knowledge by IIBA
- BPMN è un marchio registrato di Business Process Management Initiative (BPMI), attualmente incorporato in Object Management Group
- RUP è un marchio registrato di Rational Software Corporation
- SysML è un marchio registrato di Object Management Group
- TMMI è un marchio registrato di TMMi Foundation
- UML è un marchio registrato di Object Management Group

## 6. Definizioni

### 5

**5 x Why:** vedi *Five Why's*.

### A

**Acceptance (Accettazione):** Vedi *acceptance testing*.

**Acceptance criteria (Criteri di accettazione):** I criteri che un componente o sistema deve soddisfare per poter essere accettato da un utente, un cliente o un'altra entità autorizzata. [IEEE 610].

**Acceptance testing (Test 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. [IEEE 610].

**Accuracy (Accuratezza):** La capacità del prodotto software di fornire il risultato corretto o previsto, oppure con il necessario grado di precisione. [ISO/IEC 25000].

**Accuracy testing (Test di accuratezza):** Il processo di testing per determinare l'accuratezza di un prodotto SW.

**Activity diagram:** Una rappresentazione grafica di flussi di lavoro di attività sequenziali e di azioni conseguenti supportate da scelte, iterazioni e parallelismi.

**Ad hoc review (Review ad-hoc):** Vedi *informal review*.

**Adaptability (Adattabilità):** La capacità del prodotto software di adattarsi a diversi ambienti specificati, senza utilizzare azioni o mezzi diversi da quelli forniti a tale scopo dal software stesso. [ISO/IEC 25000]. Vedi anche *portability*.

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

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

**Agile software development (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 inter-funzionali che si auto-organizzano.

**Agreeing on requirements (Accordo sui requisiti):** vedi *Requirements acceptance*.

**Apprenticing:** Un processo di apprendimento dal cliente sul suo lavoro. Il cliente insegna al Requirement Engineer, con un rapporto maestro - studente.

**Artefact (Artefatto):** Uno dei risultati ottenuti durante lo sviluppo del software. Alcuni artefatti (ad es. use case, class diagram e altri modelli UML, requisiti e documenti di progettazione) aiutano a descrivere le funzionalità, l'architettura e la progettazione del software. Altri artefatti riguardano il processo di sviluppo stesso - come piani di progetto, business case e valutazioni del rischio.

**Assessment:** Attività di determinazione del valore quantitativo o qualitativo di un prodotto, servizio, attività, processo in relazione a criteri di qualità o di accettazione.

**Attractiveness (Attrattività):** La capacità del prodotto software di essere attraente per l'utente. [ISO/IEC 25000]. Vedi anche *usability*.

**Audit:** Una valutazione indipendente di un prodotto software, di un processo per accertare l'aderenza a standard, linee guida, specifiche e/o procedure basati su criteri oggettivi, comprendente documentazione che specifica [IEEE 1028]:

- (1) La struttura e il contenuto dei prodotti da sviluppare.
- (2) Il processo con cui i prodotti devono essere sviluppati.
- (3) Come l'aderenza a standard o linee guida debba essere misurata.

**Availability:** Il grado col quale un sistema od un componente è operativo ed accessibile, quando ne viene richiesto l'uso. Spesso espresso in percentuale [IEEE 610].

## B

**BA:** vedi *Business Analysis, Business Analyst*.

**Baseline:** Una specifica o un prodotto software che è stato formalmente revisionato o concordato, e che successivamente funge da base per ulteriori sviluppi e che può essere modificato solo attraverso un processo formale di controllo delle modifiche [IEEE 610].

**Behavioral diagram:** Un tipo di diagramma UML che descrive le caratteristiche comportamentali di un sistema o di un processo aziendale. Esso include l'attività, la macchina a stati e gli use case diagram, nonché i quattro interaction diagram. Vedi: *Interaction diagrams*

**Benefit (Beneficio):** Il valore fornito agli stakeholder [TGilb].

**Best practice:** Un metodo o una pratica innovativa che contribuisce a migliorare le performance di una organizzazione in un dato contesto, normalmente considerati i 'migliori' da altre organizzazioni che operano nel medesimo contesto.

**Bug (Baco):** Vedi *defect*.

**Business Analysis:** L'insieme di attività, conoscenze, strumenti e tecniche necessarie per identificare le esigenze aziendali e determinare soluzioni ai problemi aziendali [BABOK]. Vedi anche: *System*

## Analysis

**Business Analyst:** Una persona responsabile dell'identificazione delle esigenze dei clienti e stakeholder aziendali, volta a determinare le appropriate soluzioni che li soddisfino [BABOK]. Vedi anche: *System Analyst*

**Business Case:** Giustificazione delle motivazioni per l'avvio di un progetto o di un'attività, in termini di valore aggiunto al business conseguente ai risultati del progetto o dell'attività confrontato al costo di sviluppo.

**Business domain (Dominio di business):** (1) L'insieme di classi che rappresentano gli oggetti nel modello di business in fase di implementazione. (2) In generale un'area del business che è oggetto o impattato della soluzione pianificata.

**Business Goal (Obiettivo di business):** Obiettivo a breve o lungo termine di un'organizzazione.

**Business Need (Esigenza di business):** Definisce il problema o l'opportunità di business che i BA devono comprendere per raccomandare soluzioni appropriate.

**Business Process (Processo di business):** Una raccolta di attività progettate per produrre uno specifico output per un particolare cliente o mercato

**BPM:** vedi *Business Process Management*.

**BPMN:** vedi *Business Process Modeling Notation*.

**BPS:** vedi *Business Process Simulation*.

**Business Process Management (BPM):** Approccio gestionale focalizzato all'allineamento di tutti gli aspetti di un'organizzazione verso i desideri e le esigenze dei clienti. La gestione dei processi aziendali cerca di migliorare continuamente i processi e può quindi essere descritta come un "processo di ottimizzazione dei processi". Tali attività aziendali possono essere raggruppate in cinque categorie: progettazione, modellazione, esecuzione, monitoraggio e ottimizzazione.

**Business Process Modeling Notation (BPMN):** Una notazione grafica che descrive i passi di un processo aziendale. BPMN descrive il flusso end to end di un processo aziendale. La notazione è stata specificamente progettata per coordinare la sequenza di processi e i messaggi che fluiscono tra i diversi partecipanti al processo in un gruppo correlato di attività [BPMN.ORG].

**Business Process Simulation (BPS):** Una tecnica che consente di simulare dinamicamente l'esecuzione dei processi aziendali e dei loro parametri in base a modelli di processo.

**Business Sponsor (Sponsor di business):** Una persona che propone il nuovo progetto al gruppo di governance per consentire loro di selezionare e dare priorità al portafoglio dei progetti aziendali. [BABOK].



**Business Strategy (Strategia di business):** Un documento o una dichiarazione formale che descrive la direzione di un'organizzazione e le azioni che intraprenderà per raggiungere i propri obiettivi. La strategia aziendale può derivare da obiettivi definiti per supportare la dichiarata missione dell'organizzazione. Una tipica strategia aziendale è sviluppata in tre fasi:

- (1) analisi,
- (2) integrazione,
- (3) implementazione.

## C

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

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

**Certification (Certificazione):** Il processo di conferma che un componente, sistema o persona soddisfi i suoi requisiti specifici; ad esempio, passando un esame.

**Change Control:** Vedi *Configuration Control*.

**Change Control Board:** Vedi *Configuration Control Board*.

**Change List:** vedi *Change Log*.

**Change Log:** Un documento ufficiale contenente l'elenco di tutte le Change Request presentate per l'analisi al Change Control Board. Esso contiene la raccolta di tutte le modifiche (ad esempio per un progetto) e contiene i seguenti elementi: ID e/o titolo della Change Request, valutazione della fattibilità tecnica, costi e benefici della modifica, analisi dell'impatto e pianificazione della modifica, rapporto di test e di verifica efficacia della modifica. Non tutti questi elementi devono essere inclusi se il processo è terminato prima (ad es. se la modifica non è implementata).

**Change Management:** (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. Vedi anche *configuration management*.

**Change Request:** Un documento ufficiale che richiede la modifica (o la nuova implementazione) di funzionalità, requisiti o funzioni di un prodotto software. La richiesta di modifica dovrebbe contenere la descrizione della soluzione corrente, la giustificazione della modifica e la soluzione suggerita (o desiderata).

**Changeability (Modificabilità):** La capacità del prodotto software di consentire che specifiche modifiche software possano essere facilmente implementate [ISO/IEC 25000]. Vedi anche *maintainability*.

**Class (Classe):** Una classe descrive un insieme di oggetti che condividono le stesse specifiche di funzioni, vincoli e semantica. La classe è un tipo di classificatore le cui caratteristiche sono attributi e operazioni.

**Class diagram:** Una tipologia di diagramma di struttura statico che descrive la struttura di un sistema, mostrandone le classi, i loro attributi, le relative operazioni (o metodi) e le relazioni tra le diverse classi.

**Client (Cliente):** vedi *Customer*

**Commitment:** Il grado di impegno a soddisfare il requisito.

**Completeness of a requirement (Completezza di un requisito):** Il grado con cui un requisito contiene tutte le informazioni necessarie.

**Commercial off-the-shelf software:** Vedi *off-the-shelf software*.

**Communication diagram:** Un diagramma UML che mostra le istanze di classi, le loro interrelazioni e il flusso di messaggi che intercorre tra di loro.

**Complexity (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.

**Compliance (Conformità):** La capacità del prodotto software di aderire a standard, convenzioni o norme legali e prescrizioni similari. [ISO/IEC 25000].

**Component (Componente):** (1) Una parte minimale di un sistema che ad. es. può essere testata da sola. (2) Una parte potenziale o effettiva di un sistema più grande o di una sua specifica [TGilb].

**Component specification (Specificazione di componente):** Una descrizione di un componente funzionale in base ai suoi valori di output per predeterminati valori di input sotto specifiche condizioni, e a comportamenti non-funzionali richiesti. (ad es. utilizzazione di risorse).

**Component diagram:** Un diagramma UML che descrive gli elementi che compongono un'applicazione, un sistema o un'azienda.

**Composite Structure diagram:** Un diagramma UML che descrive la struttura interna di un elemento classificabile (come una classe, un componente o un use case), inclusi i suoi punti di interazione con altre parti del sistema.

**Conceptual model (Modello concettuale):** Un modello che descrive le specifiche tecnologiche di software/hardware.

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

**Configuration Auditing (Auditing di configurazione):** La funzionalità che controlla il contenuto delle librerie di elementi di configurazione, ad es. per conformità agli standard [IEEE 610].

**Configuration Control (Controllo di configurazione):** Un elemento della gestione della configurazione, consistente nella valutazione, coordinamento, approvazione o rifiuto, e implementazione delle modifiche agli elementi di configurazione, dopo la verifica formale della loro corretta identificazione [IEEE 610].

**Configuration Control Board (CCB):** Un gruppo di persone responsabili di valutare, approvare o rifiutare le modifiche proposte agli elementi di configurazione e di garantire l'implementazione delle modifiche approvate [IEEE 610].

**Configuration Identification:** Una parte della gestione della configurazione, che consiste nella selezione dei configuration item di un sistema e nella registrazione delle loro caratteristiche funzionali e fisiche nella documentazione tecnica [IEEE 610].

**Configuration Item:** Un insieme di prodotti di lavoro che viene progettato e trattato come una singola entità nel processo di gestione della configurazione [IEEE 610].

**Configuration Management:** Una disciplina che si avvale di compiti amministrativi, tecnici e di sorveglianza per: identificare e documentare le caratteristiche fisiche e funzionali di un configuration item, 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].

**Configuration Management tool (Strumento di Configuration Management):** Uno strumento che fornisce supporto per l'identificazione ed il controllo dei configuration item, del loro stato rispetto alle modifiche e alle versioni, e del rilascio di baseline degli appropriati configuration item.

**Consistency (Consistenza):** Il grado di uniformità, standardizzazione e coerenza tra documenti o parti di un componente o di un sistema [IEEE 610].

**Constraint (Vincolo):** Una dichiarazione di restrizione che modifica un requisito o una serie di requisiti limitando la gamma di soluzioni accettabili.

**Context (Contesto):** Vista di un sistema da ogni utile prospettiva [TGilb].

**Contractor:** vedi *Vendor*

**COTS:** Acronimo di software Commercial Off-The-Shelf. Vedi *off-the-shelf software*.

**Coverage (Copertura):** Il grado, espresso come percentuale, con il quale uno specifico coverage item è stato determinato o attivato da un insieme di test.

**Coverage item:** Un attributo o combinazione di attributi che è derivato da una o più condizioni di test, usando una tecnica di testing che consente la misura della completezza dell'esecuzione del test.

**Context diagram:** Un diagramma che rappresenta gli attori esterni al sistema che potrebbero interagire

con esso.

**Criticality of requirements (Criticità dei requisiti):** Valutazione del rischio di un requisito stimando il danno in caso di suo inadempimento.

**Customer (Cliente):** Acquirente o utente, attuale o potenziale, di prodotti o servizi di un'organizzazione.

## D

**Data definition (Definizione dei dati):** Una istruzione eseguibile in cui ad una variabile viene assegnato un valore.

**Data flow (Flusso dei 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].

**Data flow diagram:** Una rappresentazione grafica di un data flow.

**Decision table (Tabella delle decisioni)** Una tabella che mostra combinazioni di input e/o stimoli (cause) con i relativi output e/o azioni (effetti), che possono essere utilizzati per progettare casi di test.

**Data flow analysis:** Una forma di analisi statica basata sulla definizione e l'utilizzo di variabili.

**Defect (Difetto):** Una imperfezione in un componente o sistema che può causare la loro non corretta esecuzione della funzione richiesta, ad es. un'istruzione o definizione dei dati errata. Un difetto, se rilevato durante l'esecuzione, può causare una failure (errore) del componente o del sistema.

**Defect Management:** Il processo of riconoscimento e registrazione dei difetti, la loro classificazione, la loro diagnosi, l'analisi degli impatti, l'esecuzione delle relative azioni correttive e la loro risoluzione [IEEE 1044].

**Defect Management tool (Strumento di Defect Management):** Uno strumento che facilita la registrazione e il monitoraggio dello stato di difetti e modifiche. Spesso dispone di funzioni di workflow per tracciare e controllare l'assegnazione, la correzione e il re-testing dei difetti e di reporting. Vedi anche *incident management tool*.

**Defect Tracking tool (Strumento di Defect Tracking):** Vedi *Defect Management tool*.

**Deliverable:** Qualsiasi prodotto (di lavoro) che deve essere consegnato a qualcuno che non ne sia l'autore.

**Delphi method (Metodo di Delphi):** Una tecnica di comunicazione strutturata utilizzata per condurre previsioni interattive, che coinvolge un gruppo di esperti [Linstone75].

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

**Deviation (Deviazione):** Vedi *incident*.

**Dependency (Dipendenza):** Una subordinazione, di qualsiasi tipo, di un insieme di componenti verso un altro insieme di componenti, o di un insieme di requisiti o altri artefatti verso un altro [TGilb].

**Deployment diagram:** Un diagramma UML che mostra l'architettura di esecuzione di sistemi.

**Design Thinking:** Un processo collaborativo con cui le sensibilità e i metodi del progettista vengono impiegati per soddisfare le esigenze delle persone con ciò che è tecnicamente fattibile e in linea con una valida strategia aziendale. In breve, il design thinking converte l'esigenza in domanda. Il processo è descritto in tre fasi principali: ispirazione, ideazione, implementazione.

**Documentation testing (Testing della documentazione):** Test della qualità della documentazione, ad es. guida per l'utente o guida all'installazione.

**Domain (Dominio):** L'insieme da cui è possibile selezionare valori di input e/o output validi.

## E

**Efficiency (Efficienza):** La capacità del prodotto software di fornire performance appropriate, relativamente alla quantità di risorse utilizzate nell'ambito di condizioni prefissate [ISO/IEC 25000].

**Elicitation (Elicitazione):** L'atto di ottenere informazioni da altre persone. Nel contesto del Requirements Engineering, l'elicitation è il processo di raccolta dei requisiti dagli stakeholder.

**Emotional intelligence:** La competenza, la capacità e l'abilità per identificare, valutare e gestire le emozioni di sé stessi, di altri individui e di gruppi.

**End user (Utente finale):** vedi *User*

**Enterprise Analysis:** Insieme delle attività pre-progettuali che consentono di acquisire la visione futura del business aziendale per fornire un quadro di riferimento nell'identificazione dei requisiti di progetto e nella progettazione di una singola soluzione o nella pianificazione strategica a lungo termine.

**Entity (Entità):** (1) Un elemento o insieme di elementi aventi un'esistenza distinta e separata, benché non debba necessariamente essere un'esistenza materiale. (2) Un'astrazione dalle complessità di alcuni domini.

**Entity-relationship diagram:** vedi *Entity-relationship model*.

**ERD:** vedi *Entity-relationship diagram*.

**Entity-relationship model (Modello entità-relazioni):** Una rappresentazione astratta e concettuale dei dati. Il modello di entità-relazioni è costituito da un insieme di entità (rappresentate da dati), caratterizzate da attributi e collegate da relazioni.

**ERM:** vedi *Entity-relationship model*.

**Error (Errore):** Un'azione umana che produce un risultato scorretto [IEEE 610].

**Estimate (Stima):** Una valutazione numerica del livello (futuro, presente o passato) di un attributo di sistema scalare. Ciò include tutti gli attributi di prestazioni e di costo. Le stime vengono solitamente effettuate laddove la misurazione diretta è: impossibile (futura) o poco pratica (passata) o antieconomica (presente) [TGilb].

**Evaluation (Valutazione):** Vedi *testing*.

**Exception handling (Gestione delle eccezioni):** Comportamento di un componente o di un sistema in risposta a un input errato, proveniente da un utente umano o da un altro componente o sistema) oppure da un errore interno.

**Exit criteria (Criteri di uscita):** L'insieme delle condizioni (generiche o specifiche), concordate con gli stakeholder, per consentire il completamento ufficiale di un processo. Lo scopo dei criteri di uscita è di impedire che un'attività venga considerata completata, quando vi sono ancora parti dell'attività in sospeso che non sono state completate. I criteri di uscita vengono utilizzati per segnalare e pianificare quando interrompere il testing [Gilb and Graham].

**Extreme Programming:** Una metodologia di ingegneria del software utilizzata nell'ambito dello sviluppo software Agile, in cui le pratiche base sono: la programmazione a coppie, estese revisioni del codice, unit test di tutto il codice, nonché semplicità e chiarezza nel codice. Vedi anche *agile software development*.

## F

**Facilitator (Facilitatore):** Una persona o un gruppo che assiste gli altri nella realizzazione di un processo lavorativo, come il controllo della qualità o la definizione di obiettivi; in virtù del loro essere particolarmente preparati, qualificati e ben informati su quel processo [TGilb].

**Failure:** Divergenza del componente o del sistema dal rilascio, dal servizio o dal risultato previsti [Fenton].

**Failure mode:** La manifestazione fisica o funzionale di una failure. Per es. un sistema in tale stato può essere caratterizzato da operazioni lente, output scorretti o dalla interruzione dell'esecuzione [IEEE610].

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

**Failure Mode, Effects, and Criticality Analysis (FMECA):** Una estensione dell'approccio FMEA, che aggiunge una analisi di criticità, usata per tracciare la probabilità di failure rispetto alla severità delle loro conseguenze. Il risultato focalizza l'attenzione sulle failure con alta probabilità di severità delle loro conseguenze, consentendo di indirizzare e concentrare lo sforzo di correzione dove possa produrre il maggior valore. Vedi anche *Failure Mode and Effect Analysis (FMEA)*.

**Fault:** Vedi *defect*.

**Feature:** Un attributo di un componente o di un sistema specificato (o derivante) dalla documentazione dei requisiti. (per es. affidabilità, usabilità o vincoli di progettazione) [IEEE 1008].

**Feasibility Analysis:** vedi *Feasibility Study*.

**Feasibility Study (Studio di fattibilità):** L'analisi e la valutazione di un progetto proposto per determinare se (1) è tecnicamente fattibile, (2) è fattibile nel costo stimato e (3) sarà redditizio. Gli studi di fattibilità sono quasi sempre condotti dove sono in gioco ingenti somme. Vedi anche *Feasibility Analysis*.

**Five Why's (I cinque Why):** Una tecnica utilizzata per arrivare alla radice di ciò che sta realmente accadendo in una singola istanza. Per ogni risposta viene fornito un ulteriore "perché". Ciò può aiutare a identificare ulteriori requisiti e ad analizzare i requisiti in modo più approfondito.

**Free revealing:** La pratica di aziende che rilasciano al pubblico i risultati dei loro processi innovativi, piuttosto che brevettarli o tenerli segreti.

**Formal review (Revisione formale):** Una tipologia di revisione che segue un processo definito con output documentato in modo formale, per es. un'ispezione.

**Function (Funzione):** Una descrizione di "cosa" fa un sistema. Una funzione ha un corrispondente scopo implicito, ed è una parte fondamentale del sistema descritto: un sistema è costituito da attributi di funzione, attributi di prestazione, attributi di risorse (costo) e attributi di progettazione. Tutti gli attributi esistono rispettando le condizioni specificate. Una funzione può spesso essere scomposta in un insieme gerarchico di sotto-funzioni [TGilb].

**Function point:** Un'unità di misura che esprime la quantità di funzionalità aziendali fornite da un sistema informatico a un utente.

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

**Functional integration (Integrazione funzionale):** Un approccio di integrazione che combina componenti o sistemi con l'obiettivo di ottenere anticipatamente una funzionalità base funzionante.

**Functional requirement (Requisito funzionale):** Un requisito che specifica una funzionalità che un componente o sistema deve essere in grado di eseguire [IEEE 610].

**Functionality (Funzionalità):** La capacità del prodotto software di fornire funzioni che soddisfano le esigenze dichiarate e implicite, quando il software viene utilizzato in determinate condizioni [ISO/IEC 25000].

**Fuzzy Front End:** Il periodo iniziale dei processi di sviluppo di nuovi prodotti. È il momento in cui l'organizzazione definisce una struttura del prodotto da sviluppare e decide il possibile investimento [Koen].

## G

**Goal (Obiettivo):** Uno stato o un risultato desiderato di un'attività intrapresa. Gli obiettivi dovrebbero essere misurabili e definiti nel tempo, in modo che il loro avanzamento possa essere monitorato.

## H

**High-level (Alto livello):** Una posizione in una gerarchia di componenti di sistema definiti, che è più vicina alla parte superiore rispetto alla parte inferiore, rispetto all'insieme definito di tali componenti [TGilb].

**Horizontal traceability (Tracciabilità orizzontale):** Tracciamento dei requisiti per un livello di testing, attraverso gli strati della documentazione dei test (ad es. piano di test, specifiche di progettazione del testing, specifiche dei test case e specifiche della procedura di test o degli script di test).

## I

**Idea Generation:** vedi *Fuzzy Front End*.

**IE table:** vedi *Impact Estimation Table*.

**Impact (Impatto):** Valore numerico (stimato o effettivo) di un'ipotesi di progettazione su un attributo di requisito in determinate condizioni.

**Impact Analysis (Analisi d'impatto):** L'identificazione di tutti i prodotti di lavoro impattati da una modifica, inclusa la stima delle risorse necessarie per realizzare la modifica.

**Impact Estimation Table (Tabella di stima degli impatti):** Le principali specifiche di output di un processo di stima dell'impatto. Una tabella IE mostra stime o misurazioni effettive dell'effetto di un qualsiasi insieme di progettazioni (architettura, strategia, soluzione) su qualsiasi insieme di requisiti. L'accento è posto sull'impatto delle progettazioni sugli obiettivi di prestazione e di risorse [TGilb].

**Incremental development model (Modello di sviluppo incrementale):** Un modello di ciclo di vita dello sviluppo, in cui un progetto viene suddiviso in una serie di incrementi, ciascuno dei quali fornisce una parte della funzionalità richieste dai requisiti generali del progetto. I requisiti sono prioritizzati e consegnati in ordine di priorità nell'ambito dell'incremento appropriato. In alcune (ma non tutte) versioni di questo modello di ciclo di vita, ciascun sotto progetto segue un "mini modello V" con le sue fasi di progettazione, codifica e test.

**Indicator (Indicatore):** Una misura che fornisce una stima o una valutazione di specifici attributi [ISO 14598].



**Informal review (Revisione informale):** Una revisione non basata su una procedura formale (documentata).

**Innovation (Innovazione):** Il processo di rinnovamento di qualcosa che esiste. L'innovazione cambia i valori su cui si basa il sistema.

**Input:** Un dato ricevuto da un componente o da un sistema da una sorgente esterna.

**Input value (Valore di input):** Una istanza di un input. Vedi anche *input*.

**Inspection (Ispezione):** Una tipologia di revisione formale, basata spesso su l'esame di documenti, per rilevare difetti in un prodotto di lavoro, E' la revisione più formale e quindi sempre basata su una procedura documentata [IEEE 610, IEEE 1028]. Vedi anche *peer review*.

**Inspection leader:** Vedi *moderator*.

**Inspector:** Vedi *reviewer*.

**Installability (Installabilità):** La capacità del prodotto software di essere installato in uno specifico ambiente [ISO/IEC 25000]. Vedi anche *Portability*.

**Integration (Integrazione):** Il processo di combinazione di componenti o sistemi in aggregati più grandi.

**Interaction diagram:** Un sottoinsieme di behavior diagram che enfatizzano le interazioni tra oggetti. Ciò include comunicazione, panoramica delle interazioni, sequence, and timing diagram.

**Interaction overview diagram:** Una variante di un activity diagram che visualizza il flusso di controllo all'interno di un sistema o di un processo aziendale.

**Interoperability (Interoperabilità):** La capacità del prodotto software di interagire con uno o più componenti o sistemi specificati [ISO/IEC 25000]. Vedi anche *functionality*.

**Interview (Intervista):** Una tecnica di conversazione in cui l'intervistatore chiede al corrispondente informazioni su specifici argomenti.

**Iterative development model (Modello di sviluppo incrementale):** 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.

## L

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

**Learnability (Apprendibilità):** La capacità del prodotto software di consentire all'utente di imparare il suo l'utilizzo [ISO/IEC 25000]. Vedi anche *Usability*.

**Lifecycle model (Modello del ciclo di vita):** Una descrizione dei processi, workflow e attività usati nelle varie fasi di sviluppo, delivery, manutenzione e ritiro di un sistema [CMMI].

## M

**Maintainability (Manutenibilità):** La facilità con la quale un prodotto software può essere modificato per correggere difetti, per soddisfare nuovi requisiti, per facilitare la futura manutenzione o per adattarlo a un ambiente modificato [ISO/IEC 25000].

**Maintenance (Manutenzione):** Il processo di modifica di un componente o di un sistema dopo il suo rilascio per correggere difetti, per migliorare gli attributi di qualità o per adattare il prodotto ad un ambiente modificato. [IEEE 1219].

**Management review (Revisione di management):** 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 e la loro realizzazione nel sistema, controlla lo stato di avanzamento del processo, determina lo stato dei piani e delle schedulazioni, conferma i requisiti e il loro recepimento nel sistema e valuta l'efficacia dell'operato rispetto agli obiettivi del processo stesso. [IEEE 610, IEEE 1028].

**Maturity (Maturità):** (1) La capacità di un'organizzazione con riferimento all'efficacia e all'efficienza dei suoi processi e delle sue pratiche di lavoro. Vedi anche *Capability Maturity Model*. (2) Il grado col quale un componente o un sistema soddisfa i requisiti di affidabilità nella normale operatività [ISO 9126] Vedi anche *reliability*.

**Maturity level (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].

**Maturity model (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.

**Measure (Misura - valore):** Il valore o la categoria assegnata ad un attributo di una entità ottenuto da una misurazione [ISO 14598].

**Measurement (Misura - processo):** Il processo di assegnazione di un valore o categoria ad una entità per quantificare un suo attributo [ISO 14598].

**Measurement scale (Scala di misura):** Una scala che vincola il tipo di analisi dei dati che possono essere eseguiti [ISO 14598].

**Metric (Metrica):** Una scala di misurazione e il metodo usato per la misurazione [ISO 14598].

**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.

**Mind-map (Mappa mentale):** Un diagramma utilizzato per rappresentare parole, idee, attività o altri elementi connessi e disposti intorno ad una parola chiave o un'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 relativa documentazione.

**Modeling tool (Strumento di modeling):** Uno strumento che supporta la verifica di modelli di software o di sistemi [Graham].

**Moderator (Moderatore):** Una persona neutrale responsabile di un'ispezione o di un processo di revisione.

**Module (Modulo):** Vedi *component*.

**MoSCoW:** Una tecnica che consente di prioritizzare i requisiti, allocando una priorità appropriata espressa nei seguenti termini: Deve avere, Dovrebbe avere, Potrebbe avere e Non avrà (desiderebbe avere in futuro).

**MOST:** Una tecnica che consente di eseguire un'analisi ambientale interna, definendone gli attributi per garantire che il progetto sia allineato a ciascuno dei seguenti quattro attributi: Missione, Obiettivi, Strategie, Tattiche.

## N

**Need (Necessità):** Qualcosa desiderato da uno stakeholder. Soddisfare questa necessità avrebbe un certo valore per alcuni stakeholder. Una necessità potrebbe non essere accettata come requisito formale e potrebbe non essere definita come prioritaria in modo da essere effettivamente realizzata (progettata e implementata). Necessità è un termine spesso usato come punto di vista di un problema dello stakeholder prima che venga eseguita la specifica dei requisiti [TGilb].

**New Product Development (Sviluppo di un nuovo prodotto):** Un processo completo per portare sul mercato un nuovo prodotto o servizio.

**Non-conformity (Non-conformità):** Mancato soddisfacimento di uno specifico requisito [ISO 9000].

**Non-functional requirement (Requisito non funzionale):** Un requisito che non si riferisce alla funzionalità, ma ad attributi quali affidabilità, efficienza, usabilità, manutenibilità e portabilità.

**NPD:** vedi *New Product Development*

## O

**Object (Oggetto):** Un'istanza di una classe in OOAD.

**Object diagram:** Un diagramma UML che descrive gli oggetti e le loro relazioni in un determinato momento, in genere un caso speciale di un class diagram di un communication diagram.

**Object-oriented analysis and design (Analisi e disegno object-oriented):** Un approccio di ingegneria del software che modella un sistema come un gruppo di oggetti interagenti. Ogni oggetto rappresenta una certa entità di interesse nel sistema che viene modellato ed è caratterizzato dalla sua classe, dal suo stato (data elements) e dal suo comportamento. OOAD comprende Object-oriented analysis (OOA) e Object-oriented design (OOD). OOA applica tecniche di modellazione di oggetti per analizzare i requisiti funzionali di un sistema. OOD elabora i modelli di analisi per produrre specifiche di implementazione.

**Off-the-shelf software:** Un prodotto software sviluppato per il mercato generale, (ad es. per un numero elevato di clienti) e consegnato a molti clienti nello stesso formato.

**OOA:** vedi *Object-oriented analysis and design*.

**OOAD:** vedi *Object-oriented analysis and design*.

**OOD:** vedi *Object-oriented analysis and design*.

**Operability (Operabilità):** La capacità del prodotto software di consentire agli utenti di operare su di esso e di controllarlo [ISO/IEC 25000]. Vedi anche *Usability*.

**Output:** Dati trasmessi da un componente o da un sistema ad una destinazione.

## P

**Pair Programming:** Un approccio di sviluppo del software (specialmente in Agile) in base al quale due programmatori codificano insieme sulla stessa postazione di lavoro. Il che implica che vengono eseguite revisioni del codice in tempo reale.

**Pareto analysis (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%).

**Path (Cammino):** Una sequenza di eventi (ad es. istruzioni eseguibili) di un componente o sistema, da un punto di ingresso ad un punto di uscita.

**PBS:** vedi *Product Breakdown Structure*

**Peer review:** Una revisione di un prodotto di lavoro del software da parte di colleghi allo scopo di

identificare difetti e miglioramenti. Esempi sono ispezioni, revisioni tecniche e walkthrough.

**Performance (Prestazione):** Il grado in cui un sistema o un componente realizza le funzioni a lui assegnate, entro determinati limiti di tempo di elaborazione e alla velocità di throughput [IEEE 610]. Vedi anche *efficiency*.

**Persona:** Un personaggio fittizio, una descrizione di un archetipo, che rappresenta le diverse tipologie di utenti che utilizzeranno il prodotto o la soluzione finale. Essa dovrebbe rappresentare un gruppo di persone con gli stessi bisogni, attitudini, comportamenti o aspettative nei confronti del prodotto.

**PESTLE:** Una tecnica utilizzata per eseguire un'analisi ambientale esterna, esaminando fattori esterni che hanno influenza su un'organizzazione. Essa analizza i seguenti sei attributi: Politico, Economico, Sociologico, Tecnologico, Legale, Ambientale.

**Point of view (Punto di vista):** Una certa prospettiva sul sistema o sui requisiti.

**Portability (Portabilità):** La facilità con la quale il prodotto software può essere trasferito da un ambiente hardware o software ad un altro ambiente [ISO/IEC 25000].

**Priority (Priorità):** Il livello di importanza (di business) assegnato ad un elemento, ad es. un difetto.

**Process (Processo):** Un insieme di attività correlate tra loro, le quali trasformano un input in output. [ISO 12207].

**Process Assessment (Assessment di processo):** Una valutazione disciplinata dei processi software di un'organizzazione con riferimento ad un modello [ISO 15504].

**Process Improvement (Miglioramento del processo):** Una lista di attività progettato per migliorare le performance e la maturità dei processi delle organizzazioni e il risultato di tale programma [CMMI].

**Process model (Modello di processo):** (1) Uno schema in cui processi della stessa natura sono classificati, ad es. un modello di miglioramento dei test. (2) Una descrizione dei processi di sviluppo SW indipendente dal loro metodo.

**Process requirement (Requisito di processo):** Un requisito legato al processo di sviluppo.

**Product (Prodotto):** Un output di un processo.

**Product Breakdown Structure:** Una scomposizione dei componenti di un prodotto.

**Product requirement (Requisito di prodotto):** Un requisito relativo al prodotto del processo di sviluppo, che influenza la qualità del prodotto stesso.

**Product risk (Rischio di prodotto):** Un rischio direttamente correlato alla qualità del prodotto. Vedi anche *Risk*.

**Project (Progetto):** Un insieme di attività coordinate e controllate con date di inizio e di fine, intraprese per realizzare un obiettivo che sia conforme ai requisiti specificati, compresi i vincoli temporali, di costo e di risorse [ISO 9000].

**Project risk (Rischio di progetto):** Un rischio legato alla gestione e al controllo del progetto. Vedi anche *risk*.

**Prototype (Prototipo):** Un primo esempio o modello sviluppato per testare un concetto o un processo o una cosa da replicare o da cui imparare. I prototipi nel Requirements Engineering possono essere utilizzati per l'elicitazione e la convalida dei requisiti.

## Q

**Quality (Qualità):** Il grado con il quale un componente, un sistema od un processo soddisfa i requisiti specificati e/o le esigenze e le aspettative dell'utente/cliente [IEEE 610].

**Quality Assurance:** Componente della gestione della qualità focalizzata sul fornire fiducia che i requisiti di qualità saranno soddisfatti [ISO 9000].

**Quality attribute (Attributo di qualità):** Un aspetto o una caratteristica che attiene alla qualità di un elemento [IEEE 610].

**Quality characteristic (Caratteristica di qualità):** Vedi *Quality attribute*.

**Quality Management:** Un insieme di attività coordinate per gestire e controllare gli aspetti della qualità in una organizzazione. La gestione ed il controllo della qualità generalmente include la creazione della politica e degli obiettivi di qualità, della pianificazione e del controllo della qualità, dell'assicurazione e del miglioramento della qualità [ISO 9000].

## R

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

**Recoverability (Recuperabilità):** La capacità del prodotto software di ristabilire uno specificato livello di performance e recuperare i dati inquinati nel caso di failure [ISO/IEC 25000]. Vedi anche *Reliability*.

**Redundancy (Ridondanza):** Ricorrenza multipla delle stesse informazioni in luoghi diversi.

**Release (Rilascio):** Una versione della soluzione consegnata per l'installazione e l'utilizzo da parte degli utenti.

**Reliability (Affidabilità):** La capacità di un prodotto o un sistema di eseguire le funzionalità richieste sotto prefissate condizioni, per uno specifico periodo di tempo o per un numero specificato di transazioni [ISO/IEC 25000].

**Replaceability (Sostituibilità):** La capacità del prodotto software di essere utilizzato al posto di un altro specificato prodotto software, per lo stesso scopo e nello stesso ambiente [ISO/IEC 25000]. Vedi anche *Portability*.

**Requirement (Requisito):** (1) Una condizione o capacità richiesta da un utente per risolvere un problema o raggiungere un obiettivo. (2) Una condizione o capacità che deve essere soddisfatta o posseduta da un sistema o sua componente per soddisfare un contratto, uno standard, una specifica o altri documenti formalmente richiesti. (3) Una rappresentazione documentata di una condizione o capacità come in (1) o (2) [IEEE 610].

**RD:** vedi *Requirements Development*.

**RE:** vedi *Requirements Engineering*.

**RM:** vedi *Requirements Management*

**Requirements acceptance (Accettazione dei requisiti):** Un processo di formalizzazione dell'accordo che il contenuto e l'ambito dei requisiti sono accurati e completi tra tutte gli stakeholder [BABOK].

**Requirements analysis (Analisi dei requisiti):** Una serie di compiti, attività e strumenti per determinare se i requisiti elicitati non sono chiari, incompleti, ambigui o contraddittori e successivamente il documentare i requisiti in forma di modello consistente.

**Requirement attribute (Attributo del requisito):** Informazioni descrittive su un requisito che arricchisce la sua definizione oltre la dichiarazione della funzionalità richiesta. Ad es. origine, motivazione, priorità, proprietario, numero di release e di versione [Wieggers].

**Requirements Development:** Insieme di attività, compiti, tecniche e strumenti per identificare, analizzare e validare i requisiti. Includono il processo di trasformazione delle esigenze in requisiti. Nel modello CMMI il Requirements Development è un'area del processo di ingegneria del SW a livello di maturità 3

**Requirements elicitation:** vedi *Elicitation*

**Requirements Engineering:** Una sottodisciplina dell'ingegneria dei sistemi e del software che si occupa della definizione degli obiettivi, delle funzionalità e dei vincoli dei sistemi hardware e software [Laplanche]. Essa coinvolge i seguenti sottoprocessi: elicitazione, analisi e negoziazione, specifiche, modellazione del sistema, validazione e gestione dei requisiti.

**Requirements Management:** Un processo continuo di documentazione, analisi, tracciamento, definizione delle priorità, comunicazione, accettazione dei requisiti e di gestione delle modifiche dei requisiti. Nel modello CMMI, il Requirements Management è un'area di processo di Project Management a livello di maturità 2.

**Requirements Management tool (Strumento di Requirements Management):** Uno strumento che supporta la registrazione dei requisiti, dei loro attributi (ad es. priorità, responsabilità etc.) e di altre annotazioni, e che agevola la tracciabilità e la gestione delle modifiche attraverso le varie fasi della gestione dei requisiti e delle loro modifiche. Alcuni strumenti 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.

**Requirements model (Modello dei requisiti):** Una rappresentazione dei requisiti degli utenti, utilizzando testo e diagrammi. I modelli dei requisiti possono essere chiamati anche modelli di requisiti utente o modelli di analisi e possono integrare le specifiche testuali dei requisiti.

**Requirements phase (Fase 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].

**Requirements source (Fonte dei requisiti):** La fonte da cui sono derivati i requisiti. Le fonti possono essere stakeholder, documenti, processi aziendali, sistemi esistenti, mercato, ecc.

**Requirements specification (customer) (Specifiche dei requisiti):** Una specifica che descrive l'area del problema da risolvere. (La specifica dei requisiti del cliente viene solitamente da lui fornita e contiene una descrizione delle capacità richieste ad una soluzione dal punto di vista del cliente.)

**Requirements traceability (Tracciabilità dei requisiti):** La capacità di definire, catturare e seguire le tracce lasciate dai requisiti su altri elementi dell'ambiente di sviluppo del software e la traccia lasciata da tali elementi sui requisiti stessi [Pinheiro F.A.C. and Goguen J.A].

**Requirements Traceability Matrix (Matrice di tracciabilità dei requisiti):** Un documento, solitamente sotto forma di tabella, che mette in relazione due documenti di base, che richiedono una relazione multi-molti per determinare la completezza della relazione

**Review (Revisione):** Una valutazione di uno stato di prodotto o progetto per accertare discrepanze dai risultati pianificati e raccomandare miglioramenti. Ad es. revisione manageriale, revisione informale, revisione tecnica, ispezione e walkthrough [IEEE 1028].

**Review tool (Strumento di revisione):** Uno strumento che fornisce il supporto al processo di revisione. Caratteristiche tipiche includono la pianificazione ed il monitoraggio, la comunicazione, le revisioni di collaborazione ed un archivio per la raccolta e la registrazione di metriche.

**Reviewer (Revisore):** +

**Risk (Rischio):** Un fattore che potrebbe avere future conseguenze negative, generalmente espresso come impatto e probabilità [ISTQB].

**Risk analysis (Analisi dei rischi):** Il processo di identificazione e valutazione dei rischi per stimare il loro impatto e la probabilità di accadimento.

**Risk category (Categoria di rischio):** Vedi *risk type*.

**Risk control (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.

**Risk identification (Identificazione dei rischi):** Il processo di ricerca, riconoscimento e descrizione dei rischi, usando tecniche come il brainstorming, checklist e storia delle failure.

**Risk level (Livello di rischio):** L'importanza di un rischio, definita dalle sue caratteristiche di impatto e probabilità. Il livello di rischio può essere utilizzato per determinare l'intensità dei test da eseguire.



Un livello di rischio può essere espresso qualitativamente (ad esempio alto, medio, basso) o quantitativamente.

**Risk Management:** Le attività sistematiche (applicando procedure e pratiche) per identificare, analizzare, prioritizzare e controllare i rischi.

**Risk mitigation (Riduzione del rischio):** Vedi *Risk control*.

**Risk type (Tipo di rischio):** Un insieme di rischi raggruppati per uno o più fattori comuni, come ad.es. attributi di qualità, cause, localizzazione ed effetti potenziali.

**RTM:** Vedi *Requirements Traceability Matrix*.

**RUP:** Vedi *Rational Unified Process*.

## S

**Safety (Protezione):** La capacità di un sistema di raggiungere livelli accettabili di rischio di danni a persone, aziende, software, proprietà o ambiente in uno specifico contesto di utilizzo [ISO/IEC 25000].

**Scalability (Scalabilità):** La capacità del prodotto software di essere esteso per adattarsi a carichi crescenti [Gerrard].

**Scenario:** (1) Una linea di azione, di eventi o situazioni che portano al risultato specificato. (2) Una sequenza ordinata di interazioni tra entità specificate (ad es. un sistema e un attore). (3) In UML: una traccia di esecuzione di uno use case.

**Scope (Ambito):** La capacità ed il grado di influenzare qualcosa. L'ambito può applicarsi a qualsiasi cosa, come una specifica, un sistema o un progetto determinati [TGilb].

**Scribe (Scriba):** La persona che registra, su un modulo, ogni difetto rilevato e qualsiasi suggerimento per il miglioramento del processo durante una riunione di revisione. Lo scriba dovrebbe assicurarsi che il modulo di registrazione sia leggibile e comprensibile.

**Scrum:** Una metodologia incrementale ed iterativa per la gestione di progetti comunemente utilizzata nello sviluppo software "Agile". Vedi anche *Agile software development*.

**Security (Sicurezza):** Attributi di prodotti software che hanno la capacità di impedire l'accesso non autorizzato, accidentale o intenzionale, a programmi e dati [ISO/IEC 25000]. Vedi anche *Functionality*.

**Sequence diagram:** Una rappresentazione strutturata UML di un comportamento, espresso come una serie di passaggi sequenziali nel tempo. Il sequence diagram è una sorta di interaction diagram che mostra come i processi operano l'uno con l'altro e in quale ordine.

**Signoff:** vedi *Requirements acceptance*.

**SME:** vedi *Subject Matter Expert*

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

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

**Software Failure Mode, Effects, and Criticality Analysis (SFMECA):** Vedi *Failure Mode, Effects, and Criticality Analysis (FMECA)*.

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

**Software quality (Qualità del software):** L'insieme di funzionalità e caratteristiche di un prodotto software che identifica la sua capacità di soddisfare esigenze predefinite o implicite. [ISO/IEC 25000].

**Software quality characteristic (Caratteristica di qualità del software):** Vedi *Quality attribute*.

**Software Process Improvement (Miglioramento del processo software):** Un insieme di attività volte a migliorare le performance e la maturità dei processi software di una organizzazione ed i risultati ottenuti [CMMI].

**Solution (Soluzione):** (1) La soluzione è l'implementazione del requisito. (2) Un'idea progettuale che, se implementata, dovrebbe portare alla soddisfazione parziale o totale di un insieme di requisiti per risolvere un problema (definito) [TGilb].

**Solution model (Modello della soluzione):** Un modello che descrive l'area della soluzione da diversi punti di vista sul sistema.

**Solution validation (Validazione della soluzione):** Attività che spiega l'appropriatezza della soluzione agli stakeholder e agli sponsor

**Subject Matter Expert:** Una persona che esibisce il più alto livello di competenza nell'esecuzione di un lavoro, di un compito o di un'attività [Pyzdek, Thomas and Paul A. Keller].

**Solution Specification (Specificazione della soluzione):** Specifica dei requisiti del sistema o specifica dei requisiti del software. Descrive l'area della soluzione. Vedi anche *Functional Specification*

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

**Stability (Stabilità):** La capacità di un componente o un sistema di essere modificato in modo efficace ed efficiente senza introdurre difetti o degradare la qualità esistente [ISO/IEC 25000]. Vedi anche

*Maintainability.*

**Stakeholder (Soggetto interessato):** Qualsiasi persona che abbia un interesse in un progetto IT. Gli stakeholder del progetto sono individui e organizzazioni che sono attivamente coinvolti nel progetto o i cui interessi possono essere influenzati dall'esecuzione o dal completamento del progetto. Gli stakeholder possono esercitare un controllo sia sulle caratteristiche operative del sistema, sia su considerazioni sulla vita del sistema a lungo termine (come portabilità, costi del ciclo di vita, considerazioni ambientali e dismissione del sistema) [TGilb].

**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 es. norme ISO/IEC, standard IEEE e standard organizzativi) [CMMI].

**State machine (Macchina a stati):** Un modello di comportamento composto da un numero finito di stati, transizioni tra questi stati e azioni, simile a un diagramma di flusso.

**State transition (Transizione di stato):** Una transizione tra due stati di un componente o di un sistema.

**State machine diagram:** vedi *State machine*

**Storytelling:** Una tecnica di persuasione utilizzata per convincere l'interlocutore sugli argomenti del narratore. Le storie si basano su ipotesi o situazioni reali vissute durante la fase di ricerca. Le storie possono riguardare il prodotto, l'utente e l'esperienza dell'utente.

**Structure diagram:** Un tipo di diagramma che descrive gli elementi di una specifica indipendentemente dal tempo. Esso include classi, strutture composite, componenti, rilasci, oggetti.

**Suitability (Adeguatezza):** La capacità del prodotto software di fornire un appropriato insieme di funzioni per specifiche attività e obiettivi dell'utente [ISO/IEC 25000]. Vedi anche *Functionality*.

**SWOT:** Un'analisi utilizzata per determinare i punti di forza e di debolezza delle prestazioni di un'organizzazione e per identificare opportunità e pericoli (per es. punti deboli e minacce interne ed esterne). I quattro attributi di SWOT sono: Punti di forza, Punti di debolezza, Opportunità, Minacce.

**SysML:** Un linguaggio di modellazione generico per applicazioni di ingegneria dei sistemi. Supporta le specifiche, l'analisi, la progettazione, la verifica e la validazione di un'ampia gamma di sistemi e di sistemi di sistemi.

**System (Sistema):** Un insieme di elementi interagenti, organizzato per ottenere una specifica funzione od un gruppo di funzioni [IEEE 610].

**System Analysis (Analisi di sistema):** Un insieme di attività, metodi, tecniche e strumenti incentrati sulla traduzione dei requisiti aziendali in requisiti di sistema.

**System Analyst (Analista di sistema):** Una persona tecnicamente istruita, che si interessa di problemi di business, pianifica soluzioni software, raccomanda software e sistemi e coordina lo sviluppo per soddisfare i requisiti aziendali o di altro tipo. Il compito di System Analyst è di trasformare i requisiti aziendali in specifiche tecniche.

**System boundary (Confini del sistema):** Il confine tra un sistema e il suo contesto.

**System Interface Analysis (Analisi dell'interfaccia di sistema):** Uno studio dell'ambito del lavoro richiesto per integrare la nuova soluzione negli ambienti aziendali e tecnici.

**Systems Modeling Language:** vedi *SysML*.

## T

**Technical review (Revisione tecnica):** Una tipologia di revisione formale condotta da un gruppo di persone tecnicamente qualificate, che esamina l'idoneità di un prodotto di lavoro all'uso previsto e identifica discrepanze rispetto alle specifiche e agli standard [Gilb and Graham, IEEE 1028]. Vedi anche *peer review*.

**Testability (Testabilità):** La capacità di un prodotto software di essere testato efficacemente ed efficientemente [ISO/IEC 25000]. Vedi anche *Maintainability*.

**Testable requirements (Requisiti testabili):** Il livello di chiarezza e completezza di un requisito, che consente la progettazione del testing, la redazione di casi di test e la loro esecuzione volta a dimostrare il soddisfacimento del requisito stesso [IEEE 610].

**Timing diagram:** Un diagramma UML che illustra il cambiamento di stato o la condizione di un'istanza di classe o un ruolo nel tempo.

**Traceability (Tracciabilità):** Il grado col quale si può stabilire una relazione fra due o più prodotti di lavoro. Vedi anche *horizontal traceability*, *vertical traceability*.

**Trend (Tendenza):** Tendenza di un mercato (o di un prodotto o servizio) a spostarsi in una determinata direzione nel tempo [G. Fontanills and T. Gentile].

**Trial and Error:** Un processo per acquisire conoscenza generando/prototipando soluzioni, testandole e imparando dai propri errori. Il test della soluzione viene normalmente eseguito utilizzando i prototipi, pertanto i prototipi dovrebbero essere facili ed economici da creare.

## U

**UML:** vedi *Unified Modeling Language*.

**Unified Modeling Language:** Un linguaggio di modellazione generalizzato nel campo dell'ingegneria del software. Esso include un insieme di tecniche di notazione grafica per creare modelli visivi di sistemi di software, come use case diagram, activity diagram, class diagram e molti altri.

**Understandability (Comprensibilità):** La capacità del prodotto software di consentire all'utente di comprendere facilmente se il software è adatto alle sue esigenze e come possa essere utilizzato per particolari compiti e condizioni d'uso [ISO/IEC 25000]. Vedi anche *Usability*.

**Usability (Usabilità):** La capacità del software di essere facilmente compreso, imparato, utilizzato e di essere attraente per l'utente, quando usato in determinate condizioni [ISO/IEC 25000].

**Use case:** 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.

**Use Case diagram:** Un diagramma UML che mostra use case, attori e loro interrelazioni.

**User (Utente):** Una persona che utilizza un prodotto software.

**User Innovation (Innovazione d'utente):** Un'innovazione svolta dagli utenti consumer (singoli o gruppi di utenti finali). Il creatore dell'innovazione è l'utente finale che sviluppa o perfeziona prodotti e servizi disponibili nel sito di utilizzo.

## V

**Value (Valore):** Beneficio percepito, da uno o più stakeholder, come conseguenza degli attributi di sistema realizzati o modificati. Il valore non è linearmente correlato a un miglioramento del sistema: ad es. un piccolo cambiamento a livello di attributo potrebbe aggiungere immenso valore percepito a un gruppo di stakeholder per un costo relativamente basso. Il valore è l'utilità, la validità, la convenienza o l'importanza di un componente di sistema o di uno stato del sistema, percepiti dagli stakeholder, in condizioni specificate. Il valore è relativo a un singolo stakeholder: non è assoluto [TGilb].

**Validation (Validazione):** Conferma tramite un 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].

**Vendor:** Una persona, un gruppo o un'organizzazione che fornisce la soluzione.

**Verification (Verifica):** Conferma tramite esame (e attraverso la fornitura di evidenze obiettive) che specifici requisiti sono stati soddisfatti [ISO 9000].

**Version (Versione):** Una specifica forma, struttura, contenuto di un prodotto di lavoro.

**Version Control (Controllo di versione):** Vedi *Configuration Control*.

**Vertical traceability (Tracciabilità verticale):** Il tracciamento dei requisiti attraverso i livelli della documentazione di sviluppo verso i componenti.

**Vision (Visione):** Un'immagine del futuro risultato di un progetto come soluzione all'esigenza o al problema dichiarato.

**V-model:** Una struttura per descrivere le attività del ciclo di vita di sviluppo del software, dalla specifica dei requisiti alla manutenzione. Esso illustra come le attività di test possano essere integrate in ogni fase del ciclo di vita.

## W

**Walkthrough:** Un tipo di revisione in cui un autore guida passo a passo i membri della revisione all'esame di un prodotto di lavoro per favorire una conoscenza collegiale del suo contenuto.

[Freedman and Weinberg, IEEE 1028]. Vedi anche *Peer review*.

**WBS:** Vedi *Work Breakdown Structure*.

**Workshop:** Una sorta di incontro, per un periodo breve e intenso, focalizzato su un argomento specifico (precedentemente definito e annunciato ai partecipanti), che solitamente coinvolge gli stakeholder che rappresentano aree e/o domini diversi.

**Work Breakdown Structure:** Decomposizione gerarchica di un prodotto di lavoro in componenti più elementari e le loro relazioni reciproche [CMMI].

## 7. Annex A (Informativa)

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

[BABOK] International Institute of Business Analysis (2006), *A guide to the Business Analysis Body of Knowledge*, version 1.6

[Beizer] B. Beizer (1990), *Software Testing Techniques*, van Nostrand Reinhold, ISBN 0-442- 20672-0

[BPMN.ORG] Object Management Group/Business Process Management Initiative

[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

[Deming] D. W. Edwards (1986), *Out of the Crisis*, MIT Center for Advanced Engineering Study, ISBN 0-911379-01-0

[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.

[G. Fontanills and T. Gentile] G. Fontanills and T. Gentile (2001), Start Market Course, George Fontanills, Tom Gentile, John Wiley and Sons Inc.

[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.

[Gilb and Brodie RQNG] T. Gilb and L. Brodie (2010), *What's fundamentally wrong? Improving our approach towards capturing value in requirements specification*,

[Graham] D. Graham, E. van Veenendaal, I. Evans and R. Black (2007), *Foundations of Software Testing*, Thomson Learning, ISBN 978-1-84480-355-2

[Koen] Koen et al. (2001), *Providing clarity and a common language to the 'fuzzy front end'*. Research Technology Management, 44 (2), pp.46-55

[Laplante] Laplante, Phil (2009). *Requirements Engineering for Software and Systems (1st ed.)*. Redmond, WA: CRC Press. ISBN 1-42006-467-3.

[Pinheiro F.A.C. and Goguen J.A] Pinheiro F.A.C. and Goguen J.A., An object-oriented tool for tracing requirements, in: *IEEE Software* 1996, 13(2), pp. 52-64

[Pyzdek, Thomas and Paul A. Keller] Pyzdek, Thomas and Paul A. Keller (2009). *The Six Sigma Handbook*, Third Edition. New York, NY: McGraw-Hill. ISBN 0071623388.

[TMap] M. Pol, R. Teunissen, E. van Veenendaal (2002), *Software Testing, A guide to the TMap Approach*, Addison Wesley, ISBN 0-201-745712.

[TGilb] vedi: <http://gilb.com>, Planguage Concept Glossary