

ISO/IEC/IEEE 12207:2026 Systems and software engineering, Software life cycle processes (2ª edizione, aprile 2026)

Quadro di processi per il ciclo di vita del software

63 requisiti. Per questa norma non esiste un certificato accreditato. Versione di 07/2026.

ISO/IEC/IEEE 12207 definisce un quadro di 30 processi che copre l'intero ciclo di vita del software, dall'accordo allo sviluppo e all'esercizio fino alla dismissione. Non esiste uno schema di certificazione accreditato: la norma tratta esplicitamente la conformità come un'autodichiarazione in tre varianti (conformità completa ai risultati, conformità completa ai compiti, oppure conformità adattata con una registrazione documentata dell'adattamento), per cui il risultato di una valutazione è una dichiarazione di conformità e non un certificato. Le organizzazioni che necessitano di un giudizio esterno commissionano una valutazione dei processi basata sulla serie ISO/IEC 33000, che fornisce un profilo di capacità sotto forma di rapporto.

Azienda	Data	Redatto da

6.1 Processi di accordo (Agreement Processes)					6
6.1.1	Il processo di Acquisizione è implementato: viene identificata l'esigenza di software o servizi approvvigionati esternamente, i fornitori vengono selezionati, e un accordo viene concluso e monitorato.				Documento
Come si dimostra: Le evidenze comprendono documenti di gara o di requisiti, confronti tra offerte, contratti o ordini di acquisto firmati e verbali di accettazione. Nei team piccoli è sufficiente un breve ticket di acquisizione per ogni acquisto nel sistema di gestione delle segnalazioni, che copra l'esigenza, il confronto tra alternative e l'approvazione.					
ApertoIn corsoSoddisfattoNon applicabile					
Evidenza / motivazione					
6.1.1.2	I risultati dell'acquisizione sono raggiunti: i requisiti dell'elemento acquisito sono definiti, l'accordo è concluso e l'adempimento è confermato.				
Come si dimostra: Criteri di accettazione nel contratto o nel ticket, oltre al test di accettazione o al verbale di accettazione. Per le dipendenze open source, il file di blocco (lockfile) con un criterio di selezione documentato e la scansione delle licenze fungono da evidenza di accettazione.					
ApertoIn corsoSoddisfattoNon applicabile					
Evidenza / motivazione					
6.1.1.3	Le attività di acquisizione sono svolte: l'identificazione dell'esigenza, la selezione dei fornitori, la conclusione dell'accordo, il monitoraggio dell'avanzamento e l'accettazione sono eseguiti in modo tracciabile.				
Come si dimostra: Registrazioni provenienti dal sistema di gestione delle segnalazioni e dalla cartella dei contratti, integrate da report di stato dei fornitori. Per le piccole organizzazioni è sufficiente un elenco fornitori con la data dell'ultima valutazione.					
ApertoIn corsoSoddisfattoNon applicabile					
Evidenza / motivazione					
6.1.2	Il processo di Fornitura è implementato: l'organizzazione esamina le richieste, presenta un'offerta, conclude un accordo, consegna il prodotto o servizio e chiude formalmente l'accordo.				Documento
Come si dimostra: Offerte, conferme d'ordine, note di consegna e di rilascio, verbali di consegna e fatture. Le note di rilascio per ogni versione sono l'evidenza di consegna più pratica per i team software.					
ApertoIn corsoSoddisfattoNon applicabile					
Evidenza / motivazione					
6.1.2.2	I risultati della fornitura sono raggiunti: l'ambito della consegna è concordato, il prodotto viene consegnato e la responsabilità su di esso è definita.				
Come si dimostra: Artefatti di rilascio firmati con numero di versione, elenco delle modifiche (changelog) e un messaggio di consegna al cliente. Inoltre, gli accordi di supporto e garanzia nelle condizioni contrattuali.					
ApertoIn corsoSoddisfattoNon applicabile					
Evidenza / motivazione					

6.1.2.3 Le attività di fornitura sono svolte: la fattibilità viene valutata, la consegna viene eseguita e l'accordo viene chiuso una volta adempiuto.

Come si dimostra: Una registrazione della stima di fattibilità o di sforzo nell'offerta, il registro di distribuzione (deployment) della pipeline di integrazione continua come evidenza di consegna, e un breve verbale di chiusura per ogni ordine.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.2 Processi organizzativi di abilitazione dei progetti (Organizational Project-Enabling Processes)

9

6.2.1 La Gestione del Modello di Ciclo di Vita è implementata: l'organizzazione definisce, mantiene e rende disponibili i modelli di ciclo di vita, i processi e le procedure utilizzati dai progetti.

[Documento](#)

Come si dimostra: Una descrizione snella del processo nel repository, ad esempio un file CONTRIBUTING.md più una pagina che descrive il flusso di lavoro di sviluppo con il modello di ramificazione (branching), la definizione di completato e la cadenza dei rilasci. Le sue modifiche sono databili tramite la cronologia di Git.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.2.1.2 I risultati sono raggiunti: i modelli di ciclo di vita in uso sono definiti, i progetti ne sono a conoscenza e vengono riesaminati regolarmente per verificarne l'efficacia.

Come si dimostra: Note di retrospettiva o una voce di riesame annuale nel repository che indichi il motivo dell'adattamento del flusso di lavoro. Le richieste di integrazione delle modifiche (pull request) sui documenti di processo costituiscono l'evidenza dell'efficacia.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.2.2 La Gestione dell'Infrastruttura è implementata: l'infrastruttura necessaria per lo sviluppo, il collaudo e l'esercizio è definita, fornita e mantenuta.

Come si dimostra: Infrastruttura come codice nel repository (Terraform, Docker Compose, configurazione degli esecutori di integrazione continua), più un elenco dei servizi in uso con i relativi responsabili. Per i team piccoli è sufficiente una singola pagina con ambienti, accessi e regole di backup.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.2.2.3 Le attività infrastrutturali sono svolte: le esigenze vengono identificate, le risorse vengono acquisite, gestite operativamente e dismesse quando non sono più necessarie.

Come si dimostra: Cronologia delle modifiche della configurazione dell'infrastruttura in Git, cruscotti di monitoraggio e ticket per le modifiche agli ambienti. Gli ambienti dismessi sono documentati nel ticket con una data.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.2.3 La Gestione del Portafoglio è implementata: i progetti vengono avviati, prioritizzati, dotati di risorse, valutati in modo continuativo e conclusi quando non generano più valore.

Come si dimostra: Una roadmap di prodotto o una bacheca di portafoglio con priorità e date di decisione, più le registrazioni delle iniziative interrotte. Per le piccole organizzazioni è sufficiente un documento di roadmap con riesame trimestrale.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.2.4 La Gestione delle Risorse Umane è implementata: le competenze richieste vengono determinate, il personale viene qualificato e le conoscenze vengono costruite e conservate.

Come si dimostra: Una matrice delle competenze per ruolo, registrazioni di formazione e certificati, una checklist di inserimento. Nei team piccoli è sufficiente una tabella delle competenze che mostri le lacune e la formazione pianificata, insieme all'evidenza della formazione completata.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.2.5	La Gestione della Qualità è implementata: gli obiettivi di qualità, la politica per la qualità e le responsabilità sono definiti, e il raggiungimento degli obiettivi viene valutato.				Documento
	Come si dimostra: Una breve politica per la qualità nel repository, obiettivi di qualità definiti (copertura dei test, densità dei difetti, tempo di risposta agli incidenti) e le metriche corrispondenti provenienti dall'integrazione continua e dal sistema di gestione delle segnalazioni.				
	Aperto	In corso	Soddisfatto	Non applicabile	
Evidenza / motivazione					
6.2.5.2	I risultati della gestione della qualità sono raggiunti: gli obiettivi di qualità sono misurabili, gli scostamenti vengono rilevati e le azioni correttive vengono monitorate.				
	Come si dimostra: Report di tendenza dalla pipeline di integrazione continua (copertura, build falliti), statistiche sui difetti dal sistema di gestione delle segnalazioni, e ticket per le azioni conseguenti con una data di completamento.				
	Aperto	In corso	Soddisfatto	Non applicabile	
Evidenza / motivazione					
6.2.6	La Gestione della Conoscenza è implementata: la conoscenza, le competenze e gli asset riutilizzabili vengono raccolti, condivisi e mantenuti disponibili per i progetti successivi.				
	Come si dimostra: Una base di conoscenza interna o wiki, registrazioni delle decisioni architetturali nel repository, librerie riutilizzabili con documentazione. Le note di analisi post incidente (postmortem) sono un'evidenza solida.				
	Aperto	In corso	Soddisfatto	Non applicabile	
Evidenza / motivazione					
6.3 Processi tecnici di gestione (Technical Management Processes)					16
6.3.1	La Pianificazione del Progetto è implementata: ambito, deliverable, cronoprogramma, impegno, ruoli e interfacce sono pianificati, e il piano viene tenuto aggiornato.				Documento
	Come si dimostra: Un piano di progetto o un backlog con traguardi, stime e responsabili. Una bacheca di sprint o di traguardi mantenuta nel sistema di gestione delle segnalazioni soddisfa il requisito purché ambito e date siano visibili.				
	Aperto	In corso	Soddisfatto	Non applicabile	
Evidenza / motivazione					
6.3.1.2	I risultati della pianificazione sono raggiunti: i compiti e le risorse sono assegnati, i ruoli sono nominati e i piani sono concordati con le parti coinvolte.				
	Come si dimostra: Segnalazioni assegnate con un responsabile e una versione target, più la panoramica dei traguardi. Le ipotesi di capacità sono annotate brevemente nel ticket di pianificazione.				
	Aperto	In corso	Soddisfatto	Non applicabile	
Evidenza / motivazione					
6.3.1.3	Le attività di pianificazione sono svolte: l'ambito del progetto è definito, i piani vengono prodotti e vengono aggiornati ogni volta che si verificano cambiamenti.				
	Come si dimostra: Cronologia delle versioni del documento di piano o di roadmap in Git, e commenti di modifica sui traguardi. Questo rende databile ogni aggiornamento.				
	Aperto	In corso	Soddisfatto	Non applicabile	
Evidenza / motivazione					
6.3.2	La Valutazione e il Controllo del Progetto sono implementati: l'avanzamento rispetto al piano viene valutato, gli scostamenti vengono rilevati e vengono adottate azioni di controllo.				Documento
	Come si dimostra: Report di stato, riesami di sprint, metriche di burndown o di throughput dal sistema di gestione delle segnalazioni, e registrazioni delle decisioni prese quando si sono verificati scostamenti. È sufficiente una breve nota di stato settimanale.				
	Aperto	In corso	Soddisfatto	Non applicabile	
Evidenza / motivazione					

6.3.3	La Gestione delle Decisioni è implementata: le decisioni vengono preparate in modo strutturato, le alternative vengono valutate, e la scelta viene motivata e registrata.	Documento
Come si dimostra: Registrazioni delle decisioni architetturali nel repository, che coprano contesto, opzioni, decisione e conseguenze. Per decisioni minori è sufficiente un commento motivato nella richiesta di integrazione delle modifiche (pull request).		
Aperto In corso Soddisfatto Non applicabile		
Evidenza / motivazione		
6.3.3.2	I risultati della gestione delle decisioni sono raggiunti: la necessità di una decisione viene riconosciuta, le alternative vengono valutate, e la decisione presa è tracciabile insieme alla sua motivazione.	
Come si dimostra: La cartella delle registrazioni delle decisioni architetturali (ADR) con numerazione sequenziale e stato (proposta, accettata, superata). La cronologia di Git documenta la data e l'autore.		
Aperto In corso Soddisfatto Non applicabile		
Evidenza / motivazione		
6.3.4	La Gestione dei Rischi è implementata: i rischi vengono identificati, analizzati, trattati e monitorati in modo continuativo.	Documento
Come si dimostra: Un registro dei rischi con probabilità, impatto, trattamento e responsabile. I rischi tecnici possono essere monitorati come segnalazioni con un'etichetta di rischio, insieme ai risultati delle scansioni delle dipendenze e delle vulnerabilità.		
Aperto In corso Soddisfatto Non applicabile		
Evidenza / motivazione		
6.3.4.2	I risultati della gestione dei rischi sono raggiunti: i rischi vengono prioritizzati, vengono definite soglie di intervento, e l'efficacia del trattamento viene monitorata.	
Come si dimostra: Cronologia degli aggiornamenti del registro dei rischi, ticket di rischio chiusi con una motivazione, e report della scansione automatizzata delle dipendenze che mostrano la data di risoluzione.		
Aperto In corso Soddisfatto Non applicabile		
Evidenza / motivazione		
6.3.5	La Gestione della Configurazione è implementata: gli elementi di configurazione vengono identificati, le baseline vengono definite, le modifiche vengono controllate, e lo stato di ogni elemento può essere determinato in qualsiasi momento.	Documento
Come si dimostra: Git come controllo di versione con tag per i rilasci, rami principali protetti, file di blocco (lockfile) per le dipendenze e un elenco dei componenti software (SBOM) per ogni rilascio. La cronologia dei tag costituisce la registrazione delle baseline.		
Aperto In corso Soddisfatto Non applicabile		
Evidenza / motivazione		
6.3.5.2	I risultati della gestione della configurazione sono raggiunti: le baseline sono stabilite, le modifiche sono tracciabili e i rilasci sono riproducibili.	
Come si dimostra: Una build riproducibile a partire da un tag Git tramite la pipeline di integrazione continua, più i log di build e un registro degli artefatti con i codici di controllo (checksum).		
Aperto In corso Soddisfatto Non applicabile		
Evidenza / motivazione		
6.3.5.3	Le attività di gestione della configurazione sono svolte: le richieste di modifica vengono valutate, approvate o respinte, e la loro incorporazione viene verificata.	
Come si dimostra: Cronologia delle richieste di integrazione delle modifiche (pull request) con revisione obbligatoria e una segnalazione collegata, e regole di unione (merge) nel repository (protezione dei rami) come evidenza del controllo delle approvazioni.		
Aperto In corso Soddisfatto Non applicabile		
Evidenza / motivazione		

6.3.6	La Gestione delle Informazioni è implementata: le informazioni da creare e conservare vengono determinate, e vengono mantenute, protette e rese disponibili a chi ne ha diritto.			Documento
	Come si dimostra: Un concetto di gestione delle registrazioni che copra il luogo di conservazione, il responsabile, il periodo di conservazione e i diritti di accesso. I permessi del repository e i log di backup documentano la protezione e la disponibilità.			

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.3.7	Il processo di Misurazione è implementato: le esigenze informative vengono determinate, misure adeguate vengono definite, raccolte, analizzate e utilizzate per le decisioni.		
	Come si dimostra: Metriche dall'integrazione continua e dal sistema di gestione delle segnalazioni, come il tempo di attraversamento (lead time), il tasso di difetti, la copertura dei test, la frequenza dei rilasci. Ciò che conta è il percorso documentato dalla misura alla decisione, ad esempio nel verbale della retrospettiva.		

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.3.7.2	I risultati della misurazione sono raggiunti: i dati raccolti sono validi, l'analisi è documentata e i risultati vengono comunicati.		
	Come si dimostra: Report della pipeline generati automaticamente con marca temporale, e una sintesi periodica delle metriche pubblicata nel canale del team o nella wiki.		

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.3.8	L'Assicurazione della Qualità è implementata: prodotti e processi vengono verificati in modo indipendente rispetto ai criteri definiti, e gli scostamenti vengono registrati e risolti.			Documento
	Come si dimostra: Revisione obbligatoria a quattro occhi nella richiesta di integrazione delle modifiche (pull request), esecuzioni di linter e di test come condizione per l'unione (merge), e registrazione degli scostamenti riscontrati come segnalazioni. Un team piccolo ottiene l'indipendenza tramite revisori a rotazione invece di un reparto qualità dedicato.			

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.3.8.2	I risultati dell'assicurazione della qualità sono raggiunti: la conformità del processo e la conformità del prodotto vengono valutate, e le azioni correttive vengono monitorate fino alla chiusura.		
	Come si dimostra: Commenti di revisione con uno stato di risoluzione, ticket di errore chiusi che fanno riferimento al commit di correzione, e un'esecuzione della pipeline positiva (verde) per il tag di rilascio.		

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4 Processi tecnici (Technical Processes)

26

6.4.1	L'Analisi di Business o di Missione viene svolta: il problema da risolvere, lo spazio delle soluzioni e i vincoli dell'iniziativa vengono determinati.		
	Come si dimostra: Una visione di prodotto, una descrizione del problema o un business case nella wiki, integrati da analisi di mercato o interviste agli utenti. Per iniziative piccole è sufficiente un'epica con una descrizione del problema e un obiettivo nel sistema di gestione delle segnalazioni.		

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4.2	La Definizione delle Esigenze e dei Requisiti delle Parti Interessate viene svolta: le parti interessate vengono identificate, le loro esigenze vengono raccolte e trasformate in requisiti verificabili delle parti interessate.	Documento
Come si dimostra: Un elenco delle parti interessate, note di intervista, storie utente (user story) con criteri di accettazione. I criteri di accettazione nel ticket costituiscono la forma verificabile del requisito.		
ApertoIn corsoSoddisfattoNon applicabile		
Evidenza / motivazione		
6.4.2.2	I risultati sono raggiunti: i requisiti delle parti interessate sono concordati, prioritizzati e tracciabili fino all'esigenza sottostante.	
Come si dimostra: Un backlog prioritizzato che collega la storia all'epica, più il commento di approvazione del cliente nel ticket.		
ApertoIn corsoSoddisfattoNon applicabile		
Evidenza / motivazione		
6.4.3	La Definizione dei Requisiti di Sistema viene svolta: i requisiti tecnici del sistema software vengono derivati dai requisiti delle parti interessate, incluse le caratteristiche di qualità e i vincoli.	Documento
Come si dimostra: Un documento dei requisiti o segnalazioni strutturate che coprano i requisiti funzionali e non funzionali (prestazioni, sicurezza, accessibilità). Collegare ogni requisito a un caso di test rende esplicita la verificabilità.		
ApertoIn corsoSoddisfattoNon applicabile		
Evidenza / motivazione		
6.4.3.2	I risultati della definizione dei requisiti sono raggiunti: i requisiti sono univoci, privi di contraddizioni, verificabili e tracciabili fino ai requisiti delle parti interessate.	
Come si dimostra: Una matrice di tracciabilità, oppure in alternativa collegamenti tra segnalazioni dalla storia al test e al commit. I test di accettazione automatizzati costituiscono al tempo stesso l'evidenza della verificabilità.		
ApertoIn corsoSoddisfattoNon applicabile		
Evidenza / motivazione		
6.4.4	La Definizione dell'Architettura di Sistema viene svolta: viene definita e valutata rispetto ai requisiti un'architettura con viste, blocchi costitutivi e interfacce.	Documento
Come si dimostra: Una panoramica architetturale nel repository (ad esempio diagrammi arc42 o C4 come immagine più testo), insieme a registrazioni delle decisioni architetturali che contengano le alternative valutate.		
ApertoIn corsoSoddisfattoNon applicabile		
Evidenza / motivazione		
6.4.4.2	I risultati della definizione dell'architettura di sistema sono raggiunti: l'architettura viene valutata rispetto ai requisiti, le alternative vengono ponderate e la scelta viene motivata.	
Come si dimostra: Una registrazione di decisione architetturale (ADR) che ponderi opzioni e conseguenze, più i risultati di un riesame architetturale nella richiesta di integrazione delle modifiche (pull request) o in un verbale di riunione.		
ApertoIn corsoSoddisfattoNon applicabile		
Evidenza / motivazione		
6.4.5	La Definizione della Progettazione viene svolta: esistono progetti dei blocchi costitutivi architetturali sufficienti per l'implementazione, incluso il modello dei dati e i contratti di interfaccia.	
Come si dimostra: Specifiche di interfaccia (OpenAPI, schema GraphQL, definizioni di tipo), script di migrazione del database e note di progettazione nella richiesta di integrazione delle modifiche (pull request). Il codice stesso, con i suoi tipi e i confini dei moduli, è un'evidenza di progettazione accettabile purché le decisioni siano documentate.		
ApertoIn corsoSoddisfattoNon applicabile		
Evidenza / motivazione		

6.4.5.2 I risultati della definizione della progettazione sono raggiunti: le caratteristiche di progettazione sono tracciabili fino all'architettura e la fattibilità è confermata.

Come si dimostra: Riferimenti dalla richiesta di integrazione delle modifiche (pull request) alla registrazione di decisione architeturale (ADR) e al requisito, e prototipi o esperimenti esplorativi (spike) con una nota sul risultato come evidenza di fattibilità.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4.6 L'Analisi di Sistema viene svolta: le questioni tecniche vengono indagate con metodi analitici al fine di motivare le decisioni su architettura, progettazione o rischi.

Come si dimostra: Misurazioni di carico e risultati di prove comparative (benchmark), analisi di costo o di compromesso (trade off), note di modellazione delle minacce. I risultati vanno allegati alla relativa registrazione di decisione architeturale (ADR) come breve nota di analisi.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4.7 L'Implementazione viene svolta: gli elementi software vengono realizzati in conformità alla progettazione e alle regole di codifica, e i risultati vengono posti sotto controllo di configurazione.

Come si dimostra: Cronologia Git con commit che fanno riferimento alle segnalazioni, regole di linter e di formattazione come condizione per l'unione (merge), e build riuscite nella pipeline di integrazione continua.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4.7.3 Le attività di implementazione sono svolte: gli standard di codifica sono definiti, la realizzazione viene revisionata e gli elementi vengono resi disponibili per l'integrazione.

Come si dimostra: File di configurazione di linter e formattatore nel repository, revisioni delle richieste di integrazione delle modifiche (pull request) superate, e artefatti di build pubblicati nel registro.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4.8 L'Integrazione viene svolta: gli elementi software vengono assemblati progressivamente in un sistema funzionante e le interfacce vengono verificate.

Come si dimostra: Esecuzioni di integrazione continua ad ogni unione (merge), test di integrazione e test di contratto sulle interfacce, più i log di build con marca temporale.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4.8.2 I risultati dell'integrazione sono raggiunti: la strategia di integrazione è definita, le interfacce sono coerenti, e i difetti di integrazione vengono registrati e risolti.

Come si dimostra: La definizione della pipeline nel repository come strategia di integrazione, esecuzioni fallite seguite da esecuzioni positive (verdi) come evidenza della risoluzione dei difetti, e ticket di errore con il commit di correzione.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4.9 La Verifica viene svolta: per ogni requisito specificato esiste un'evidenza verificabile e tracciabile che il software realizzato lo soddisfi effettivamente.[Documento](#)

Come si dimostra: Test automatizzati unitari, di integrazione e di regressione con un report di test dalla pipeline di integrazione continua, analisi statica e revisioni del codice. Il report di test per ogni tag di rilascio costituisce l'evidenza centrale.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4.9.2 I risultati della verifica sono raggiunti: i criteri di verifica sono definiti, i controlli vengono eseguiti e gli scostamenti vengono registrati.

Come si dimostra: Soglie di accettazione definite (ad esempio una copertura minima dei test, assenza di rilievi critici), report di test archiviati e una segnalazione per ogni scostamento.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4.9.3 Le attività di verifica sono svolte: l'ambito e i metodi della verifica vengono pianificati, eseguiti, e i risultati vengono analizzati.

Come si dimostra: Un piano di test o una strategia di test come breve pagina nel repository, la configurazione della pipeline con le sue fasi di verifica, e l'analisi nell'approvazione del rilascio.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4.10 Il processo di Transizione viene svolto: il sistema viene installato nel suo ambiente di destinazione, consegnato pronto per l'uso, e la sua capacità operativa viene confermata.[Documento](#)

Come si dimostra: Un manuale operativo di distribuzione (runbook), log di distribuzione dalla pipeline, test di verifica rapida (smoke test) dopo il rilascio, e un piano di ripristino (rollback) documentato. Le note di rilascio documentano la consegna agli utenti e all'esercizio.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4.11 La Validazione viene svolta: nelle condizioni d'uso previste viene dimostrato che il software soddisfa effettivamente le esigenze originarie delle parti interessate.[Documento](#)

Come si dimostra: Test di accettazione con il cliente, una fase beta o pilota con riscontri degli utenti, e criteri di accettazione nel ticket contrassegnati come soddisfatti. I dati di utilizzo provenienti dall'esercizio forniscono un ulteriore supporto alla validazione.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4.11.2 I risultati della validazione sono raggiunti: i criteri di validazione sono definiti, la validazione avviene in un ambiente rappresentativo e gli scostamenti vengono registrati.

Come si dimostra: Un verbale di accettazione con data e partecipanti, risultati dall'ambiente di preproduzione o pilota, e ticket di riscontro con il relativo stato di risoluzione.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4.12 Il processo di Esercizio è implementato: il sistema viene gestito come previsto, viene monitorato e gli utenti vengono supportati.

Come si dimostra: Monitoraggio e allarmistica con soglie definite, ticket di incidente con tempi di risposta, e una casella di supporto o un canale di ticket. Per le piccole organizzazioni è sufficiente un manuale operativo di poche pagine.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4.12.2 I risultati dell'esercizio sono raggiunti: la strategia di esercizio e le sue misure sono definite, gli incidenti vengono registrati, e le prestazioni vengono valutate rispetto alle attese.

Come si dimostra: Metriche di disponibilità e di errore dal monitoraggio, registri degli incidenti, e analisi post incidente (postmortem) con le azioni da esse derivate.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4.13

Il processo di Manutenzione è implementato: le correzioni dei difetti, gli adattamenti e i miglioramenti vengono registrati, valutati, implementati e rilasciati senza compromettere l'integrità del sistema.

[Documento](#)

Come si dimostra: Ticket di errore e di modifica con priorità e causa radice, rami di correzione urgente (hotfix), test di regressione prima del rilascio, e note di rilascio per ogni versione di manutenzione. Gli aggiornamenti regolari delle dipendenze sono documentati tramite la cronologia del file di blocco (lockfile).

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4.13.2

I risultati della manutenzione sono raggiunti: le modifiche vengono analizzate, il loro impatto viene valutato, e l'efficacia della correzione viene confermata.

Come si dimostra: Un'analisi di impatto nel ticket, un'esecuzione di test di regressione dopo la correzione, e la conferma da parte di chi ha segnalato il problema. Gli aggiornamenti di sicurezza sono documentati tramite report di scansione delle vulnerabilità che mostrano la data di risoluzione.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4.14

Il processo di Dismissione è implementato: il ritiro di sistemi o di parti di sistemi viene pianificato, i dati vengono migrati o cancellati, e le parti interessate vengono informate.

[Documento](#)

Come si dimostra: Un piano di ritiro con date, evidenza della migrazione o cancellazione dei dati (inclusi i backup), un verbale di spegnimento per l'infrastruttura, e la comunicazione agli utenti o ai clienti.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4.14.2

I risultati della dismissione sono raggiunti: gli obblighi di conservazione vengono rispettati, gli ambienti vengono spenti, e la responsabilità sul sistema si conclude.

Come si dimostra: Un archivio dei dati soggetti a obblighi di conservazione con il periodo di conservazione indicato, un ticket per lo spegnimento delle risorse con una data, e una conferma di cancellazione da parte del fornitore di hosting.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

A Conformità e adattamento (Tailoring)

6

4.1 L'ambito di applicazione è definito (punto 4.1, Generalità): viene stabilito se la norma si applica a un progetto, a un'organizzazione o a un accordo, e quali processi sono coperti.

[Documento](#)

Come si dimostra: Una breve dichiarazione di ambito nella documentazione di processo che nomini il progetto, il prodotto e i gruppi di processi applicati. Una pagina nel repository è sufficiente.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

4.2 La variante di conformità scelta viene dichiarata: conformità completa a tutti i risultati dei processi dichiarati, oppure conformità completa a tutti i compiti dei processi dichiarati.

[Documento](#)

Come si dimostra: Una dichiarazione di conformità come documento separato che indichi la variante e rimandi all'evidenza per ogni processo. Ciò produce un'autodichiarazione, non un certificato.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

4.3 Quando viene dichiarata la conformità adattata, l'adattamento viene documentato: viene indicato quali processi, compiti o risultati sono stati omessi, modificati o aggiunti.

[Documento](#)

Come si dimostra: Una tabella di adattamento con le colonne processo, decisione (adottato, adattato, omesso), motivazione e rischio. Per i team piccoli questo documento costituisce l'evidenza decisiva.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

5 I concetti chiave vengono applicati all'organizzazione: la relazione tra il modello di ciclo di vita, la struttura del progetto e i processi in uso viene presentata in modo comprensibile.

Come si dimostra: Una pagina di panoramica che mette in relazione l'approccio scelto (ad esempio iterativo, consegna continua) con i 30 processi. Un diagramma più una tabella sono sufficienti.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

A.1 L'adattamento viene trattato come una decisione deliberata: l'adattamento è consentito, ma viene applicato solo dove i vincoli dell'iniziativa lo giustificano, ed è sempre motivato.[Documento](#)

Come si dimostra: Una colonna di motivazione nella tabella di adattamento che faccia riferimento alla dimensione del progetto, alla classe di rischio, alla situazione contrattuale o ai requisiti legali. Le eliminazioni generalizzate senza motivazione non costituiscono un adattamento valido.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

A.2 Il processo di adattamento viene svolto: i fattori di influenza vengono identificati, le parti interessate vengono coinvolte, e le decisioni di adattamento vengono prese e documentate.[Documento](#)

Come si dimostra: Un verbale o ticket per la decisione di adattamento che nomini i ruoli coinvolti, più la tabella di adattamento versionata in Git. Gli effetti sui rischi vengono annotati nel registro dei rischi.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

ISO/IEC 25010:2023, Ingegneria dei sistemi e del software, SQuaRE, Modello di qualità del prodotto

Modello di qualità per i prodotti software

40 requisiti. Per questa norma non esiste un certificato accreditato. Versione di 07/2026.

ISO/IEC 25010:2023 è un modello di qualità, non una norma di requisiti con clausole verificabili, e pertanto non può essere certificato tramite accreditamento. Il modello funge da griglia di riferimento per definire in modo completo i requisiti di qualità; la misurazione segue la norma ISO/IEC 25020 e la valutazione segue la norma ISO/IEC 25040, e il risultato è un rapporto di valutazione tracciabile, non un certificato. Non è consentito pubblicizzare una presunta certificazione ISO 25010.

Azienda	Data	Redatto da

1 Idoneità funzionale3

1.1 L'insieme delle funzioni copre tutti i compiti specificati e gli obiettivi degli utenti per cui il prodotto viene utilizzato (Completezza funzionale). Gli scostamenti tra i requisiti documentati e le funzioni implementate sono noti e valutati.

Come si dimostra: L'evidenza deriva da una matrice di tracciabilità che collega ogni requisito a un caso di test, misurabile come percentuale di copertura dei requisiti. Nei team piccoli è sufficiente un sistema di gestione dei ticket con un'etichetta per requisito e un test di accettazione per obiettivo utente.

Aperto	In corso	Soddisfatto	Non applicabile
--------	----------	-------------	-----------------

Evidenza / motivazione

1.2 Per gli input definiti, il prodotto fornisce risultati con il grado di precisione richiesto (Correttezza funzionale). Le tolleranze e gli scostamenti ammissibili sono specificati.

Come si dimostra: Il dato misurabile è la densità dei difetti per mille righe di codice o per funzione, supportata da test unitari e di integrazione che confrontano i risultati attesi con quelli effettivi, oltre alle statistiche degli errori tratte dal sistema di ticket. Per la logica di calcolo, conservare set di dati di riferimento con i risultati attesi.

Aperto	In corso	Soddisfatto	Non applicabile
--------	----------	-------------	-----------------

Evidenza / motivazione

1.3 Le funzioni fornite supportano i compiti degli utenti senza passaggi superflui e si adattano all'uso previsto (Appropriatezza funzionale).

Come si dimostra: L'evidenza deriva dal tasso di successo dei compiti e dal numero di passaggi per compito utente in un test di usabilità ridotto con cinque partecipanti, registrato per ciascun compito. I dati di utilizzo o le richieste di assistenza che rivelano percorsi non ottimali completano il quadro.

Aperto	In corso	Soddisfatto	Non applicabile
--------	----------	-------------	-----------------

Evidenza / motivazione

2 Efficienza delle prestazioni3

2.1 I tempi di risposta, elaborazione e throughput restano al di sotto delle soglie definite (Comportamento temporale). Le soglie sono definite per gli scenari di utilizzo tipici.

Come si dimostra: Misurabile tramite il tempo di caricamento e la latenza di interazione rilevati con Lighthouse o Web Vitals (LCP, INP), oltre ai tempi di risposta del server ricavati dai log al 95esimo percentile. Registrare le soglie per scenario in un breve performance budget.

Aperto	In corso	Soddisfatto	Non applicabile
--------	----------	-------------	-----------------

Evidenza / motivazione

2.2 Il consumo di tempo di processore, memoria, spazio di archiviazione e larghezza di banda di rete rimane entro i limiti definiti (Utilizzo delle risorse).

Come si dimostra: L'evidenza deriva dai valori di monitoraggio di CPU, RAM e volume di dati sotto carico, oltre alle dimensioni del bundle distribuito ricavate dal report di build. Per le piccole imprese è sufficiente un registro di misurazione di un test di carico confrontato con i limiti dell'hardware target.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

2.3 Il numero massimo di utenti, transazioni o record concorrenti è noto e viene rispettato (Capacità).

Come si dimostra: L'evidenza deriva da un test di carico (per esempio con k6 o JMeter) con un punto di saturazione documentato e il punto in cui il tasso di errore inizia ad aumentare. Riassumere il risultato in un breve report di capacità che indichi il limite effettivo e la domanda prevista.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

3 Compatibilità

2

3.1 Il prodotto funziona in un ambiente condiviso senza compromettere altri prodotti (Coesistenza).

Come si dimostra: L'evidenza deriva da un test eseguito nell'ambiente target con altro software in esecuzione in parallelo, registrando conflitti di porte, utilizzo della memoria e accessi ai file. La containerizzazione con limiti di risorse definiti è la via pratica per i team piccoli.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

3.2 Lo scambio di dati con altri sistemi avviene tramite interfacce e formati definiti (Interoperabilità).

Come si dimostra: L'evidenza deriva da una descrizione dell'interfaccia sottoposta a versionamento (OpenAPI o file di schema) e da contract test rispetto a tale descrizione. Verificare i formati di importazione ed esportazione con file di esempio reali e conservare i risultati.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

4 Capacità di interazione

8

4.1 Gli utenti sono in grado di riconoscere senza aiuto esterno se il prodotto si adatta alle loro esigenze (Riconoscibilità dell'idoneità). Lo scopo e il beneficio risultano evidenti dall'interfaccia e dai test di primo livello.

Come si dimostra: L'evidenza deriva da un breve test di prima impressione: i partecipanti descrivono lo scopo del prodotto dopo 30 secondi e viene registrato il tasso di risposte corrette. Gli screenshot della pagina iniziale e delle schermate di onboarding rafforzano la documentazione.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

4.2 I nuovi utenti raggiungono i compiti principali entro un tempo definito e con un numero limitato di errori (Apprendibilità).

Come si dimostra: Misurabile come tempo necessario per completare con successo il primo compito principale e come tentativi falliti per compito, rilevato in un test con cinque nuovi utenti. Documentare l'onboarding, il tour guidato o i dati dimostrativi come misure di supporto.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

4.3 L'utilizzo è efficiente e controllabile, e i compiti ricorrenti possono essere svolti senza percorsi superflui (Operabilità).

Come si dimostra: L'evidenza deriva da un'analisi del percorso di click per ogni compito standard (passaggi previsti confrontati con quelli effettivi) e dall'utilizzo esclusivamente da tastiera senza mouse. Il registro del test di usabilità con il tasso di abbandono per compito completa la documentazione.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

4.4 Il prodotto previene gli errori operativi o li rende reversibili (Protezione dagli errori dell'utente). Le azioni critiche richiedono una conferma o possono essere annullate.

Come si dimostra: L'evidenza deriva da casi di test con input non validi e valori limite, oltre alla prova della validazione, della finestra di conferma e della funzione di annullamento. Valutare gli input errati dal log di produzione e monitorarne la frequenza come metrica.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

4.5 L'interazione è progettata per risultare coinvolgente e mantiene gli utenti motivati a portare a termine il compito (Coinvolgimento dell'utente).

Come si dimostra: Misurabile tramite il tasso di ritorno, la durata della sessione e il tasso di completamento dei processi avviati, ricavati da statistiche di utilizzo rispettose della privacy. Un breve sondaggio di soddisfazione con una scala di valutazione è sufficiente come integrazione, anche per le piccole imprese.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

4.6 Il prodotto è utilizzabile da persone con abilità, lingue e contesti culturali diversi (Inclusività).

Come si dimostra: L'evidenza deriva da una verifica rispetto al WCAG 2.2 livello AA: rapporti di contrasto, navigazione da tastiera, etichette per screen reader, testati con axe o con l'audit di accessibilità di Lighthouse più un campione manuale. Dimostrare la copertura linguistica e la localizzazione tramite la completezza dei file di traduzione.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

4.7 Gli utenti ricevono un aiuto adeguato al compito quando ne hanno bisogno, anche nelle situazioni di errore e nei casi limite (Assistenza all'utente).

Come si dimostra: L'evidenza deriva dal manuale, dai testi di aiuto contestuale e dai messaggi di errore comprensibili che suggeriscono un'azione successiva. Metrica: la percentuale di richieste di assistenza che l'aiuto esistente avrebbe potuto risolvere, ricavata dal sistema di ticket.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

4.8 L'interfaccia si spiega da sola: gli stati, gli effetti delle azioni e gli input richiesti sono riconoscibili senza conoscenze pregresse (Autodescrittività).

Come si dimostra: L'evidenza deriva da una revisione dell'interfaccia rispetto a una checklist: etichette significative, stato del sistema visibile, indicatori di avanzamento e caricamento, unità di misura nei campi numerici. Un registro di osservazione del test con gli utenti rafforza la documentazione.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

5 Affidabilità**4****5.1 In condizioni operative normali il prodotto funziona senza guasti causati da difetti del software (Assenza di guasti).**

Come si dimostra: Misurabile tramite la densità dei difetti e il tempo medio tra i guasti (MTBF) ricavati dai log di produzione, e tramite la copertura dei test come metrica preventiva. Un sistema di monitoraggio degli errori come Sentry fornisce i dati grezzi con poco sforzo.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

5.2 Il prodotto è raggiungibile e utilizzabile nelle fasce orarie concordate (Disponibilità). L'obiettivo di disponibilità è definito.

Come si dimostra: L'evidenza deriva dalla misurazione dell'uptime tramite monitoraggio esterno, espressa come percentuale di disponibilità mensile e confrontata con l'obiettivo concordato. Registrare le interruzioni con durata e causa in un semplice registro degli incidenti.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

5.3 In presenza di guasti hardware o software, il prodotto rimane in uno stato definito e utilizzabile (Tolleranza ai guasti).

Come si dimostra: L'evidenza deriva da casi di test con guasti simulati (rete non disponibile, database non disponibile, sistema terzo che non risponde) e dal comportamento documentato, per esempio modalità offline, nuovo tentativo o visualizzazione di riserva. Archiviare i risultati in un registro di test.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

5.4 Dopo un guasto, i dati e lo stato operativo possono essere ripristinati entro il tempo definito (Recuperabilità).

Come si dimostra: I dati misurabili sono il recovery time objective (RTO) e la perdita massima di dati tollerabile (RPO), comprovati da un test di ripristino effettivamente eseguito con data e durata. Per le piccole imprese è sufficiente un test di ripristino semestrale con un breve registro.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6 Sicurezza**6****6.1 I dati sono accessibili solo a persone e sistemi autorizzati (Riservatezza). I diritti di accesso sono concessi secondo il principio della necessità di conoscere.**[Documento](#)

Come si dimostra: L'evidenza deriva da un concetto di ruoli e permessi, dalla cifratura in transito e a riposo, e da test che confermano che l'accesso viene negato senza autorizzazione. Una verifica degli header e del TLS dell'applicazione completa il quadro.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.2 Le modifiche non autorizzate a programmi e dati vengono impedito o rilevate (Integrità).

Come si dimostra: L'evidenza deriva da checksum o firme degli artefatti, dalla validazione degli input e da una scansione delle dipendenze (npm audit, Dependabot) che dimostri la correzione dei risultati rilevati. I vincoli del database e i log delle migrazioni rafforzano ulteriormente questo punto.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.3 Le azioni eseguite non possono essere negate successivamente (Non ripudio).[Documento](#)

Come si dimostra: L'evidenza deriva da log a prova di manomissione con timestamp e autore, e da firme quando la transazione ha rilevanza legale. Una destinazione di log centrale e protetta da scrittura è sufficiente per le piccole imprese.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.4 Le azioni nel sistema possono essere ricondotte in modo inequivocabile a una persona o a un servizio responsabile (Responsabilità).[Documento](#)

Come si dimostra: L'evidenza deriva da log di audit che riportano un identificativo utente per ogni azione rilevante ai fini della sicurezza, unita al divieto di account condivisi. Controllo a campione: seguire una transazione dall'inizio alla fine nel log e documentare il risultato.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.5 L'identità di utenti e sistemi viene verificata in modo affidabile (Autenticità).

Come si dimostra: L'evidenza deriva dal metodo di autenticazione, dalla policy delle password e dall'autenticazione a due fattori per l'accesso amministrativo, nonché dalla validazione di token e certificati. I casi di test con token scaduti o contraffatti dimostrano l'efficacia.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

6.6 Il prodotto resiste agli attacchi e rimane operativo mentre è sotto attacco (Resistenza).

Come si dimostra: L'evidenza deriva da una scansione delle vulnerabilità, da un'analisi statica del codice (SAST) e, ove proporzionato, da un penetration test con un elenco dei risultati e lo stato di correzione. Per le piccole imprese, una scansione automatizzata nella pipeline più un breve test esterno annuale è la via pratica.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

7 Manutenibilità**5****7.1 Il prodotto è costituito da componenti separati la cui modifica interessa il minor numero possibile di altri componenti (Modularità).**

Come si dimostra: Misurabile tramite l'accoppiamento e la coesione ricavati da un'analisi statica e tramite grafi delle dipendenze (madge, dependency-cruiser) che dimostrino l'assenza di cicli. Documentare i confini dei moduli in uno schizzo architetturale sintetico.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

7.2 I componenti sono progettati in modo da poter essere riutilizzati in più di un contesto (Riutilizzabilità).

Come si dimostra: L'evidenza deriva da librerie o componenti condivisi con un proprio versionamento e un'interfaccia documentata. Metrica: la percentuale di componenti riutilizzati e il livello di duplicazione del codice ricavato dall'analisi statica.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

7.3 Le cause dei guasti e l'impatto delle modifiche pianificate possono essere determinati in modo efficiente (Analizzabilità).

Come si dimostra: L'evidenza deriva dal logging strutturato, dal monitoraggio degli errori e da metriche del codice leggibili come la complessità ciclomatica ricavata con SonarQube o ESLint. Metrica: il tempo medio per individuare la causa radice, ricavato dal sistema di ticket.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

7.4 Le modifiche possono essere effettuate senza introdurre nuovi difetti e senza degradare la qualità (Modificabilità).

Come si dimostra: L'evidenza deriva dai test di regressione nella pipeline e dal change failure rate, cioè la percentuale di difetti introdotti dalle modifiche. La revisione del codice obbligatoria e il controllo di versione con commit tracciabili rafforzano questo punto.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

7.5 Per il prodotto e i suoi componenti possono essere definiti ed eseguiti criteri di test e test efficaci (Testabilità).

Come si dimostra: Misurabile tramite la copertura dei test in percentuale (righe e branch) ricavata dal report di copertura e tramite il tempo di esecuzione della suite di test. Un'esecuzione automatica dei test a ogni commit costituisce l'evidenza del funzionamento.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

8 Flessibilità**4****8.1 Il prodotto può essere adattato ad ambienti diversi o mutevoli senza dover essere ricostruito (Adattabilità).**

Come si dimostra: L'evidenza deriva dalla configurazione mantenuta al di fuori del codice (variabili d'ambiente, file di configurazione) e da test su più ambienti target, browser o sistemi operativi all'interno di una matrice di test. Archiviare il risultato come tabella di compatibilità.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

8.2 Il prodotto gestisce volumi crescenti o decrescenti di utenti e dati senza un calo inaccettabile delle prestazioni (Scalabilità).

Come si dimostra: L'evidenza deriva da test di carico con un numero crescente di utenti, monitorando il tempo di risposta e il tasso di errore, integrati dalla strategia di scalabilità (orizzontale o verticale). Metrica: tempo di risposta con carico singolo, doppio e quintuplo.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

8.3 Il prodotto può essere installato, aggiornato e disinstallato in modo affidabile nell'ambiente target (Installabilità).

Come si dimostra: L'evidenza deriva da una guida di installazione riproducibile e da un test di installazione automatizzato su un sistema pulito, misurato come tasso di successo e tempo richiesto. Eseguire una volta anche la disinstallazione e il rollback, registrando l'esito.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

8.4 Il prodotto può sostituire un altro prodotto utilizzato per lo stesso scopo mantenendo invariati i dati e i flussi di lavoro (Sostituibilità).

Come si dimostra: L'evidenza deriva da funzioni di importazione ed esportazione in formati aperti e da un test di migrazione con dati reali del sistema precedente, verificato in termini di completezza e fedeltà dei dati. Mantenere il modello dei dati e gli script di migrazione sotto controllo di versione.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

9 Sicurezza funzionale**5****9.1 L'intervallo di funzionamento sicuro è definito, e il prodotto rileva e gestisce qualsiasi scostamento da tale intervallo (Vincolo operativo).**

Come si dimostra: L'evidenza deriva da limiti operativi documentati (valori soglia, stati ammissibili) e da test che verificano il comportamento al di fuori di tali limiti. Se il software non presenta alcun potenziale di danno a persone o beni, questo punto tipicamente non si applica; tale classificazione deve essere motivata e documentata, non semplicemente omessa.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

9.2 I pericoli che possono derivare dal prodotto o dal suo utilizzo sono identificati e valutati in modo sistematico (Identificazione dei rischi).[Documento](#)

Come si dimostra: L'evidenza deriva da una valutazione dei pericoli e dei rischi che indichi probabilità, gravità e misura di mitigazione per ciascun pericolo, praticamente in forma di tabella nel fascicolo qualità. Non applicabile solo con una motivazione scritta che attesti l'assenza di potenziale di danno a persone o beni.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

9.3 In caso di malfunzionamento, il prodotto passa a uno stato sicuro (Sicurezza in caso di guasto).

Come si dimostra: L'evidenza deriva da casi di test con un guasto forzato e dalla transizione documentata verso lo stato sicuro, per esempio spegnimento, blocco o ritorno a un valore predefinito verificato. Per software senza potenziale di danno, questo punto tipicamente non si applica, ma la motivazione deve comunque comparire nella valutazione.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

9.4 Il prodotto segnala le condizioni di pericolo imminenti con sufficiente anticipo da rendere possibile una reazione (Avviso di pericolo).

Come si dimostra: L'evidenza deriva da un concetto di allarmi e avvisi con soglie, tempo di anticipo e destinatari, oltre a un registro di test degli avvisi attivati. In assenza di potenziale di danno, questo punto tipicamente non si applica; la motivazione deve essere documentata.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione

9.5 L'interazione con altri sistemi non crea nuovi pericoli (Integrazione sicura).

Come si dimostra: L'evidenza deriva da una revisione dei rischi dell'interfaccia e da test di integrazione con i sistemi collegati, incluso il comportamento quando un sistema partner si guasta o invia dati errati. Per software senza potenziale di danno, questo punto tipicamente non si applica, e ciò deve essere registrato con una motivazione.

Aperto In corso Soddisfatto Non applicabile

Evidenza / motivazione