Indice dei contenuti

Rev.	Data	Descrizione Modifica	*Verificato* Responsabile del Sistema Integrato	*Approvato* Direzione Generale
1	30/06/2025	Aggiornamento generale dell'intero documento		
0	01/04/2019	Prima versione del documento		

1. SCOPO E CAMPO DI APPLICAZIONE

La presente politica definisce i requisiti di sicurezza da seguire durante il processo di sviluppo software all'interno di NPO Sistemi S.r.l. L'obiettivo principale della Politica di Sviluppo Sicuro è integrare le best practice di sicurezza in ogni fase del ciclo di vita dello sviluppo software, assicurando che i nostri prodotti e servizi siano progettati, sviluppati, testati e mantenuti con un alto livello di sicurezza e resilienza.

Questa politica si applica a:

- Tutte le applicazioni e i sistemi sviluppati da NPO Sistemi S.r.l.
- Tutti i team di sviluppo, inclusi sviluppatori, tester, responsabili di progetto e personale di supporto.
- Fornitori esterni e partner coinvolti nei progetti di sviluppo.

2. RUOLI E RESPONSABILITÀ

- **Team di Sviluppo**: Responsabile dell'adozione delle pratiche di sviluppo sicuro, della gestione del codice e della risoluzione delle vulnerabilità;
- **Responsabile della Sicurezza delle Informazioni**: Responsabile della supervisione dell'implementazione della sicurezza durante il ciclo di sviluppo, della valutazione delle vulnerabilità e dei test di sicurezza;
- **Project Manager**: Responsabile dell'integrazione dei requisiti di sicurezza nei progetti e della verifica della loro implementazione;
- **Fornitori Terzi**: Devono rispettare i requisiti di sicurezza definiti nella presente politica e nelle clausole contrattuali di sicurezza.

3. INTEGRAZIONE DELLA SICUREZZA NEL CICLO DI VITA DELLO SVILUPPO

Al fine di mantenere un alto livello di *sicurezza dei software* prodotti, limitando eventuali disagi ai clienti ed eccessivi costi di manutenzione, ***NPO Sistemi S.r.l.*** si è dotata di procedure di *analisi, sviluppo*, *collaudo* e *rilascio* di tutti i propri applicativi.

In dettaglio, un progetto *software* dovrà prevedere un ***ciclo di vita*** (Figura 1) che includa le seguenti fasi:

- Requisiti
- Progettazione
- Implementazione
- Collaudo
- Rilascio
- Supporto

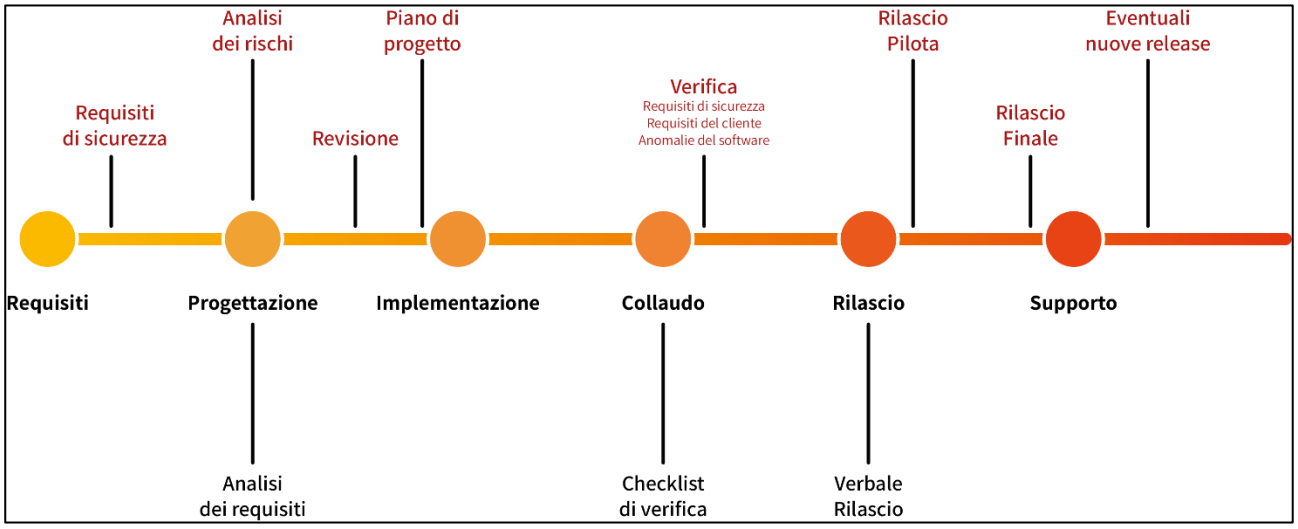

Figura 1 – Ciclo di Vita

3.1. Requisiti

I requisiti di sicurezza sono definiti all'inizio di ogni progetto e integrati nella progettazione al fine di garantire:

- **Riservatezza**: ovvero l'accesso protetto e controllato ai dati
- **Integrità**: ovvero la consistenza e la correttezza dei dati
- **Disponibilità**: ovvero la possibilità di accesso ai dati nelle modalità previste

e di prevenire:

- eventuali intrusioni effettuate da entità esterne al sistema: ovvero attacchi malevoli effettuati tramite rete internet da parte di utenti remoti
- eventuali intrusioni effettuate da entità interne al sistema: ovvero accesso ai sistemi da parte di utenti non autorizzati
- eventi accidentali: ovvero utilizzo inappropriato del software o situazioni impreviste

3.2. Progettazione

Prendere in considerazione la *sicurezza del software*, già dalle prime fasi di un progetto, è un principio indispensabile per lo sviluppo di sistemi sicuri.

La fase di progettazione tiene conto dei *requisiti del cliente* e dei *requisiti di sicurezza aziendali* sopra considerati. Per ogni progetto viene prodotta un'opportuna analisi dei requisiti al cui interno, oltre alle specifiche funzionali, emergono *i potenziali rischi del software e le relative soluzioni*.

Dal punto di vista della protezione, i fattori chiave su cui porre attenzione nella fase di progettazione possono essere ricompresi in:

- i sistemi di login
- la gestione attenta dei profili utente e dei relativi privilegi
- l'utilizzo esclusivo di transazioni criptate tramite protocollo https
- l'utilizzo esclusivo di transazioni gestite da token creati all'accesso e sottoposti a scadenza
- l'utilizzo di sistemi di monitoraggio
- l'utilizzo di sistemi di log
- l'utilizzo di sistemi di backup

NPO Sistemi S.r.l. *raccomanda* l'utilizzo delle *piattaforme hardware e software già disponibili in azienda*, al fine di ereditarne i meccanismi di sicurezza già implementati e collaudati.

3.3. Implementazione

Gli sviluppatori seguono le linee guida per la codifica sicura, come definite da standard internazionali (es. Linee Guida AGID, OWASP) ed effettuano una verifica del codice sviluppato attraverso strumenti automatizzati.

Per ridurre i rischi, gli sviluppatori prestano attenzione alla correttezza del codice e alla verifica della qualità del lavoro svolto, applicando gli standard di codifica e test.

In particolare, è necessario prestare la dovuta attenzione:

1. alla ***normalizzazione*** dei dati, per assicurarne la ***consistenza***
2. agli ***algoritmi di verifica*** dei dati inseriti da interfaccia, per assicurarne la ***correttezza***
3. all'***ottimizzazione del codice***, per assicurare la velocità di esecuzione

Durante la fase di implementazione il team è tenuto a implementare il codice *esclusivamente all'interno degli ambienti di sviluppo forniti dall'azienda* e nel pieno rispetto delle *regole di sicurezza.*

Inoltre:

- gli ambienti di ***Sviluppo***, ***Test*** e ***Produzione*** sono separati logicamente
- eventuali *dati di produzione*, utilizzabili in sede di collaudo per ottenere la maggior verosimiglianza con i *sistemi di produzione*, sono distrutti al termine del collaudo stesso

3.4. Collaudo

Nella fase di collaudo il *software* è completo dal punto di vista funzionale e comincia ad essere sottoposto ai *test beta*. A tale scopo dovrà sempre essere identificato un ***Responsabile di Collaudo***, la cui figura *non dovrà mai coincidere* con quella di ***Responsabile del Progetto***.

Il ***Responsabile di Collaudo*** verificherà:

- la conformità del *software* agli standard di sicurezza di ***NPO Sistemi S.r.l.***
- la conformità del *software* ai requisiti richiesti dal cliente
- l'esistenza di eventuali anomalie del *software*
- il funzionamento delle procedure di rilascio

Al termine delle operazioni verrà redatto uno specifico documento di test che certificherà l'avvenuta esecuzione dello stesso e la cui ***validazione***, da parte del ***Responsabile di Collaudo***, sarà una *condizione necessaria* al rilascio definitivo della procedura.

3.4.1. Test di collaudo

I test possono variare da scenario a scenario, in base allo sviluppo effettuato e in base alle caratteristiche introdotte o modificate e al tipo di applicativo.

3.5. Rilascio

Il rilascio del software avviene solo dopo che i test di sicurezza sono stati completati e tutte le vulnerabilità critiche sono state risolte. Per affrontarlo al meglio è necessario:

1. Sviluppare sempre *software* in grado di effettuare auto-aggiornamenti
2. Eseguire le operazioni di rilascio solo dopo aver concordato con gli organizzatori data, ora e durata prevista dell'intervento
3. Avvisare i clienti del ripristino del servizio
4. Eseguire le operazioni di rilascio, utilizzando esclusivamente procedure verificate in fase di collaudo

5. Prevedere, laddove le condizioni lo consentano, il rilascio di una ***versione pilota del software***, al fine di effettuare una ulteriore messa a punto insieme ai responsabili del progetto presso il cliente, per prevenire eventuali vulnerabilità non emerse in fase di collaudo

3.6. Supporto e manutenzione

Non essendo possibile garantire che la fornitura di un *software* sia del tutto esente da vulnerabilità, nella fase immediatamente successiva al rilascio si dovrà prevedere un intervallo di tempo in cui il *software* sia monitorato con attenzione, al fine di rispondere il più velocemente possibile ad eventuali anomalie.

4. CONTROLLI DI SICUREZZA

- **Gestione delle Identità e degli Accessi**: solo il personale autorizzato può accedere al codice sorgente e agli ambienti di sviluppo. I controlli di accesso sono applicati in base al principio del "minimo privilegio".
- **Protezione dei Dati:** non sono utilizzati dati reali nei test senza adeguate misure di anonimizzazione o crittografia.
- **Ambiente di Sviluppo Isolato**: Gli ambienti di sviluppo, test e produzione sono separati per prevenire il rischio di accesso non autorizzato o di manomissione.

4.1. Tool per la gestione del ciclo di vita del software

Vengono impiegati strumenti avanzati che permettono di implementare un controllo rigoroso degli accessi al codice sorgente basato su profili utente, garantendo che solo il personale autorizzato possa effettuare modifiche. Questi tool offrono inoltre una gestione efficace del versioning, facilitando il tracciamento delle modifiche e la gestione delle diverse versioni del software. Infine, viene assicurata la registrazione dettagliata dei log, monitorando tutte le attività legate allo sviluppo e contribuendo a una maggiore trasparenza e sicurezza operativa.

Il codice sorgente sviluppato viene sottoposto a una revisione approfondita, volta a identificare potenziali problematiche e migliorare la qualità del software. I risultati di questa revisione vengono analizzati e sono fornite raccomandazioni per eventuali miglioramenti.

5. VULNERABILITY ASSESSMENT E PENETRATION TEST

È possibile che l'applicazione sviluppata, esclusivamente per una Major Release o qualora venga concordato con il Cliente, venga sottoposta a un processo di Vulnerability Assessment (VA) e Penetration Testing (PT) per individuare eventuali vulnerabilità del software. Generalmente tutte le applicazioni fruibili da internet vengono sottoposte, in fase di avvio e con periodicità condivisa con il cliente, a penetration test.

In caso di esecuzione di tali test, un ente terzo fornirà i risultati e internamente a NPO Sistemi S.r.l. si procederà alla pianificazione delle attività correttive per risolvere eventuali debolezze emerse. Le tempistiche di intervento varieranno in base al livello di rischio e alla priorità assegnata a ciascuna vulnerabilità.

6. UTILIZZO DI CODICE GENERATO DALL'AI

Attualmente, l'intelligenza artificiale viene impiegata in modo limitato come strumento di supporto per la generazione automatica di codice, assistendo gli sviluppatori nelle fasi più ripetitive o meccaniche del processo di sviluppo. Questo utilizzo permette di velocizzare alcuni aspetti del lavoro, senza tuttavia sostituire l'intervento umano, che resta essenziale per la progettazione e la risoluzione di problemi complessi. L'intelligenza artificiale, quindi, agisce principalmente come un complemento per migliorare l'efficienza e la produttività del team di sviluppo.

7. FORMAZIONE E CONSAPEVOLEZZA

Tutto il personale coinvolto nello sviluppo software partecipa a programmi di formazione regolari riguardanti le migliori pratiche di sviluppo sicuro e la gestione delle vulnerabilità. La formazione viene aggiornata regolarmente per riflettere le nuove minacce e tecnologie.

8. CONFORMITÀ NORMATIVA

NPO Sistemi S.r.l. garantisce che tutte le attività di sviluppo sono conformi alle normative applicabili in materia di sicurezza delle informazioni, tra cui la norma ISO/IEC 27001, il GDPR e altre normative locali o settoriali rilevanti.

9. GESTIONE DEI FORNITORI

I fornitori e partner terzi che eventualmente partecipano allo sviluppo software rispettano questa politica e sono disponibili a sottoporsi a verifiche di sicurezza per garantire che le loro pratiche siano allineate ai nostri standard.