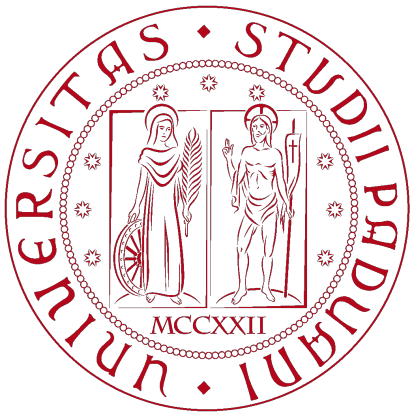


UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Tecnica e Gestione dei Sistemi Industriali

Corso di Laurea Triennale in Ingegneria Meccatronica

Tesi di Laurea Triennale

Validazione del software di sicurezza

Safety software validation

Relatore:
Prof. Diego Dainese

Candidato:
Riccardo Palmarin
2050054

Anno Accademico: 2023 - 2024

Indice

Sommario

La presente tesi ha lo scopo di analizzare e studiare il processo di validazione di un software che svolge funzioni di sicurezza, con l'obbiettivo di sviluppare una linea guida che accompagni i tecnici nel processo di validazione.

L'entrata in vigore del nuovo Regolamento UE 2023/1230 ha introdotto la necessità di certificare i software che svolgono funzioni di sicurezza, rappresentando una delle principali novità rispetto alla Direttiva Macchine 2006/42/CE. Pertanto, le aziende e i professionisti dovranno orientarsi tra diverse normative per validare i propri software di sicurezza.

In questa tesi verrà approfondito il tema dei software legati alla sicurezza e le norme armonizzate da seguire per ottenere la conformità alle Direttive in vigore. Inoltre, si illustrerà la validazione di un software di sicurezza tramite lo strumento SOFTEMA.

La tesi è composta da sette Capitoli:

- Nel capitolo 1 vengono illustrati le principali tipologie di software relativi alla sicurezza, oltre ai concetti chiave per trattare la validazione dei software.
- Nel capitolo 2 viene trattato il contesto normativo a partire dalla Direttiva Macchine 2006/42/CE e dal nuovo Regolamento UE/2023/1230. Si analizzano nello specifico le principali norme armonizzate in modo da formulare i requisiti e gli aspetti principali che un software legato alla sicurezza deve rispettare. Nello specifico si analizzano le norme EN IEC 62061:2022, UNI EN ISO 13849 e EN IEC 61508:2010.
- Nel capitolo 3 viene presentata la metodologia elaborata per la validazione del software, oltre alla creazione di una check-list per accompagnare l'intero processo. Inoltre, viene presentato il software SOFTEMA in tutte le sue funzionalità utili per la validazione dei software legati alla sicurezza.
- Nel capitolo 4 viene presentato un caso studio rilevante per l'applicazione della metodologia elaborata. Si userà lo strumento SOFTEMA per condurre il processo di validazione;
- Nel capitolo 5 vengono discussi i risultati ottenuti dal caso studio, elaborando eventuali accorgimenti o note integrative.
- Nel capitolo 6 vengono sintetizzate le principali scoperte e le principali raccomandazioni pratiche.

Introduzione

Ai fini della trattazione, si applicano i termini e le definizioni seguenti:

- *SRP/CS*: parte di un sistema di comando legata alla sicurezza;
- *Funzione di sicurezza*: funzione di una macchina il cui guasto può determinare un immediato aumento del rischio;
- *SRP*: parte legata alla sicurezza;
- *PL*: livello di prestazione. E' il livello discreto utilizzato per specificare la capacità delle parti dei sistemi di comando legate alla sicurezza di eseguire una funzione di sicurezza in condizioni prevedibili;
- *PLr*: livello di prestazione richiesto. E' il livello di prestazione applicato al fine di conseguire la riduzione del rischio richiesta per ciascuna funzione di sicurezza;
- *Misura di protezione*: misura atta a conseguire una riduzione del rischio;
- *SIL*: livello di integrità della sicurezza. E' il livello discreto per specificare i requisiti di integrità della sicurezza delle funzioni di sicurezza da assegnare ai sistemi Elettrici/Elettronici/Programmabili legati alla sicurezza;
- *LVL*: linguaggio a variabilità limitata. E' il tipo di linguaggio che offre la possibilità di combinare le funzioni di libreria predefinite, specifiche per l'applicazione, per implementare le specifiche dei requisiti di sicurezza;
- *FVL*: linguaggio a variabilità completa. E' il tipo di linguaggio che offre la possibilità di implementare una vasta gamma di funzioni e applicazioni.
- *FB*: blocco di funzioni;
- *FMEA*: analisi delle modalità e degli effetti dei guasti;
- *MMTFd*: tempo medio al guasto pericoloso. E' una previsione del tempo medio al guasto pericoloso;
- *PFHd*: probabilità media di guasto pericoloso per ora;

Capitolo 1

Software che svolgono funzioni di sicurezza

Ogni sistema di controllo di una macchina o di una *quasi-macchina*[1] è dotato di parti relative alla sicurezza che hanno lo scopo di eliminare, o diminuire, un rischio specifico. Le parti legate alla sicurezza dei sistemi di controllo asservono ad una determinata *funzione di sicurezza*.

I sistemi di controllo sono caratterizzati dalla sinergia tra componenti fisici (hardware) e componenti digitali (software). Pertanto, i software legati alla sicurezza costituiscono una parte integrante dei sistemi di controllo che svolgono funzioni di sicurezza. Si evince che i guasti del software che gestisce le funzioni di sicurezza possono provocare un aumento del rischio non accettabile.

1.1 SRASW e SRESW

Le principali norme che trattano gli aspetti relativi a software che svolgono funzioni di sicurezza distinguono due principali categorie di software:

a) SRASW: il *Safety-Related Application Software* è il software specifico dell'applicazione implementato dal costruttore della macchina e generalmente contenente sequenze logiche, limiti ed espressioni che controllano gli appropriati ingressi, uscite, calcoli e decisioni necessari per soddisfare i requisiti della parte del sistema di comando correlata alla sicurezza. Il software è generalmente scritto in un linguaggio a variabilità limitata (LVL).

b) SRESW: il *Safety-Related Embedded Software* è il software facente parte della fornitura del fabbricante del sistema di controllo e che non è accessibile per la modifica al costruttore della macchina. Il software è scritto generalmente in un linguaggio a variabilità completa (FVL).

Gli obiettivi principali dello sviluppo del software correlato alla sicurezza sono 2:

1. evitare i guasti;
2. generare un software leggibile, comprensibile e che possa essere sottoposto a prove e a manutenzione.

[1] In riferimento alle definizioni riportate nella Direttiva Macchine 2006/42/CE e nel Regolamento UE 2023/1230.

1.2 V-Model

Come si approfondirà nei capitoli 2 e 3, il processo di sviluppo più efficace per soddisfare gli obbiettivi principali è basato sul *V-Model*.

Il modello a V distingue tutte le fasi del ciclo di vita del software, incluse le attività di gestione e documentazione, atte a raggiungere la prestazione di sicurezza richiesta.

Lo sviluppo secondo il V-Model include già le attività di validazione e verifica, necessarie per documentare ogni fase del ciclo di vita del software. Lo sviluppo richiede la formulazione della specifica dei requisiti di sicurezza della funzione di sicurezza come dato in ingresso da cui partire. In figura 1.1 è illustrato il V-Model completo.

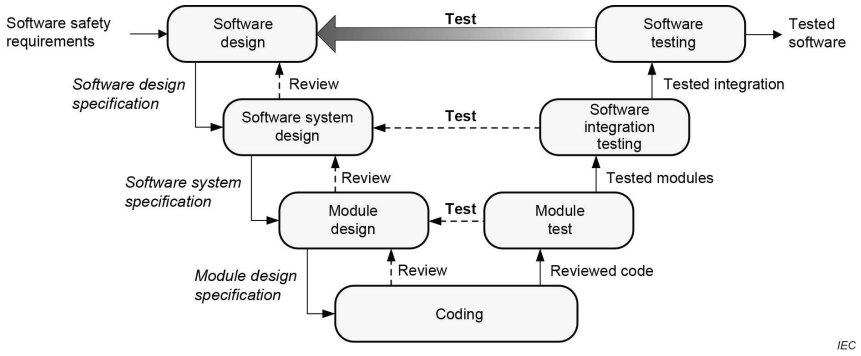


Figura 1.1: V-Model Completo

Si osserva che, al completamento di ogni fase, è prevista un'attività di revisione o di test, prima di spostarsi alla fase successiva. Seguendo le fasi proposte dal V-Model, si sviluppa un software già testato e validato. Inoltre, come illustrato nel capitolo 2, il modello a V è il metodo riconosciuto dalle principali norme armonizzate; oltre ad essere il metodo utilizzato da SOFTEMA.

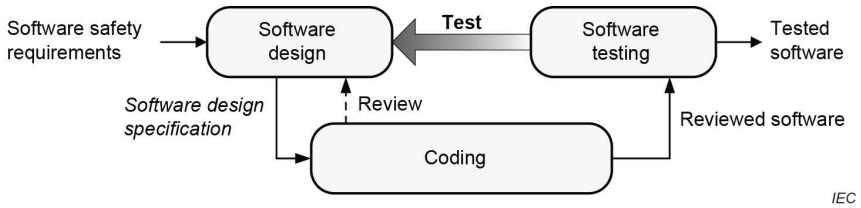


Figura 1.2: V-Model semplificato

Per i software di complessità ridotta, il processo di sviluppo illustrato dal modello a V completo può essere ridotto e semplificato, in modo da ottimizzare l'attività di sviluppo[2]. In figura 1.2 è riportato il modello a V semplificato riconosciuto dalle principali normative di riferimento.

[2]Le ipotesi sulle quali si basa la semplificazione verranno trattate al paragrafo 3.1.

1.2.1 Verifica e Validazione

Uno dei concetti più fraintesi della progettazione di software legati alla sicurezza è la differenza tra l'attività di *verifica* e l'attività di *validazione*. Le normative di riferimento sanciscono una differenza sostanziale tra le due attività.

- La *verifica* ha come obbiettivo il controllo di qualità delle attività svolte durante una fase dello sviluppo. E' un'attività prevalentemente analitica, che può essere accompagnata da test per completarla.
- La *validazione* ha come obbiettivo il controllo di qualità del sistema rispetto alla specifica dei requisiti di sicurezza iniziali. E' un'attività basata principalmente su prove, ma che, all'occorrenza, può essere accompagnata da analisi e simulazioni.

Per facilitare ulteriormente la distinzione tra le due attività, si può far riferimento a due domande:

1. *"Stiamo realizzando correttamente il sistema?"* La risposta è data dalla verifica.
2. *"Stiamo realizzando il sistema corretto?"* La risposta è data dalla validazione.

Capitolo 2

Contesto Normativo

2.1 Direttiva Macchine e Nuovo Regolamento Macchine

Il contesto normativo attuale è disciplinato dalla Direttiva Macchine 2006/42/CE, che verrà sostituita dal Regolamento UE 2023/1230 a partire dal 20 Gennaio 2027.

Il nuovo Regolamento Macchine prevede esplicitamente che i software atti allo svolgimento di funzioni di sicurezza siano marcati CE. Tuttavia, per ottenere la conformità alla Direttiva Macchine, devono essere seguite le stesse norme armonizzate del nuovo Regolamento Macchine. Pertanto, la trattazione sarà basata sul Regolamento UE 2023/1230.

Le principali norme applicabili nell'ambito dei software che svolgono funzioni di sicurezza sono la UNI EN ISO 13849:2023, la EN IEC 61508:2010 e la EN IEC 62061:2022. Il Regolamento [29] sancisce una nuova definizione di *componenti di sicurezza*: [28]:

> [...] Inoltre, la definizione di componenti di sicurezza dovrebbe riguardare non soltanto i dispositivi fisici ma anche quelli *digitali*. [...], il software che svolge una funzione di sicurezza ed è immesso in maniera indipendente sul mercato dovrebbe essere considerato un componente di sicurezza.

Inoltre, sono stati aggiornati gli allegati 1 e 2, includendo i *blocchi logici per assicurare funzioni di sicurezza* [25] nelle categorie di macchine e prodotti per cui deve essere applicata una delle procedure dell'articolo 25[1], e il *software che garantisce funzioni di sicurezza.* nell'elenco indicativo di componenti di sicurezza [26].

Un'ulteriore novità del Regolamento Macchine è riportata nell'allegato 4 [27]. Nella Documentazione Tecnica (ex Fascicolo Tecnico) dovrà essere riportato il codice sorgente o la logica di programmazione del software relativo alla sicurezza al fine di dimostrare la conformità del prodotto.

2.2 UNI EN ISO 13849:2023

Il software SOFTEMA è stato ideato per validare i software relativi alla sicurezza rispetto alla norma armonizza UNI EN ISO 13849. La norma stabilisce le linee guida e i requisiti

[1] L'articolo 25, paragrafo 3, sancisce le modalità di valutazione della conformità per i prodotti correlati alla parte B dell'allegato I

di sicurezza per la progettazione delle parti legate alla sicurezza dei macchinari, focalizzandosi prevalentemente sugli aspetti meccanici o hardware della progettazione, tuttavia fa riferimento anche ad importanti aspetti relativi al software.

Nella parte 1 della norma [9] viene trattata la sola progettazione dei sistemi correlati alla sicurezza, nella parte 2 della norma [3] si tratta nello specifico la validazione dei sistemi che svolgono funzioni di sicurezza.

2.2.1 UNI EN ISO 13849-1

Durante la progettazione di una funzione di sicurezza viene definito il Livello di Performance richiesto o PLr. Il PL è un indice di sicurezza discretizzato su cinque livelli, da a fino ad e per sicurezza crescente. I requisiti di sicurezza da applicare dipendono dal PLr della funzione di sicurezza.

2.2.1.1 Requisiti di sicurezza del software

I requisiti di sicurezza del software dipendono dalla categoria del software, nonché dal PLr.

Nel caso dei *SRESW* per componenti con PLr da a fino a d si deve tracciare il ciclo di vita del software con attività di verifica e validazione, si deve redigere della documentazione di specifica e progettazione, si deve attuare una progettazione modulare e codificata. Si devono prendere delle misure per il controllo dei guasti sistematici e si devono eseguire delle prove funzionali, ad esempio prove a scatola nera.

Il SRASW scritto in FVL deve soddisfare i requisiti del SRESW e può raggiungere un PL da a ad e. Il SRASW scritto in LVL e conforme ai seguenti requisiti può raggiungere un PL da a fino ad e. Per quanto riguarda i *SRASW* per componenti con PLr da a fino a d si deve tracciare il ciclo di vita del software tramite attività di verifica e validazione e occuparsi della redazione di documentazione di specifica e progettazione. Risulta necessario implementare una programmazione modulare e strutturata e si devono eseguire delle prove funzionali più specifiche, ad esempio prove a scatola grigia.

Al software SRESW per componenti con PLr c oppure d devono essere applicate delle misure aggiuntive. Ad esempio, si deve usufruire di un sistema di gestione del progetto conforme alla ISO 9001, si deve redigere la documentazione pertinente a ciascuna attività del ciclo di vita del software, si devono usare linguaggi di programmazione idonei e strumenti computerizzati affidabili. Si deve separare il software non legato alla sicurezza e i moduli devono avere dimensioni limitate e interfacce interamente definite. Devono essere applicate norme di progettazione e di codifica. Si deve verificare la codifica mediante walk-through/revisione e si devono attuare prove funzionali estese.

Il software SRASW per componenti con PLr da c fino ad e deve seguire delle misure aggiuntive con efficienza crescente:

1. La specifica dei requisiti di sicurezza del software deve essere sottoposta a revisione e deve essere a disposizione di ogni persona coinvolta nel ciclo di vita del

software. Deve contenere la descrizione delle funzioni di sicurezza con PLr e modalità di funzionamento, deve riportare i criteri di performance, l'architettura e la struttura dell'hardware e deve rilevare e controllare i guasti esterni.

2. Si devono selezionare strumenti, librerie e linguaggi di programmazione. Gli strumenti devono essere idonei e affidabili ($PL = e$). Si devono usare tecniche in grado di rilevare condizioni che potrebbero causare errori sistematici principalmente durante il tempo di compilazione. Inoltre, quando è ragionevole, si dovrebbero usare delle librerie di FB validate.

3. La progettazione del software deve includere metodi semi-formali per descrivere i dati e il flusso di controllo, deve prevedere una programmazione modulare e strutturata realizzata mediante blocchi funzione da librerie validate legate alla sicurezza e la struttura dovrebbe essere a 3 stadi: Ingressi => Elaborazione => Uscite. Si devono usare tecniche in grado di rilevare guasti esterni e tecniche per una programmazione difensiva che determini uno stato di sicurezza in caso di un guasto o di un errore della logica.

4. Quando un SRASW e un non-SRASW sono combinati in un unico componente non ci deve essere alcuna combinazione logica di dati legati alla sicurezza e dati non legati alla sicurezza.

5. Il codice deve essere leggibile, comprensibile e deve poter essere sottoposto a prove. Si devono usare linee guida di codifica idonee o accettate. Si devono usare controlli di integrità dei dati e il software deve essere sottoposto a simulazione.

6. Il metodo di validazione appropriato prevede prove a scatola nera per testare il comportamento funzionale e il soddisfacimento dei criteri di performance. E' raccomandato pianificare le prove includendo i casi di test e i criteri di superamento/fallimento.

7. Tutte le attività del ciclo di vita devono essere documentate. La documentazione deve essere disponibile, completa, comprensibile e leggibile.

Per SRP/CS con PLr da c fino ad e, si raccomanda il test in scenari di guasto. Dall'analisi del rischio, dagli allegati informativi della norma e dall'esperienza del team di sviluppo si possono elaborare degli scenari di guasto che possono provocare avarie della funzione di sicurezza. I guasti da considerare dipendono dalla tecnologia e dall'applicazione.

2.2.2 UNI EN ISO 13849-2

La seconda parte della norma tratta il *processo di validazione*. Lo scopo è quello di confermare che la progettazione soddisfi complessivamente i requisiti di sicurezza specificati. La validazione deve dimostrare che ogni SRP/CS soddisfi i requisiti della UNI EN ISO 13849-1, con attenzione particolare a:

1. specifica dei requisiti di sicurezza della funzione di sicurezza;
2. PL specificato
3. progettazione ergonomica dell'interfaccia utente.

La norma stabilisce che il processo di validazione deve essere attuato da persone indipendenti dal processo di progettazione della SRP/CS.

La validazione consiste nell'applicazione di analisi e nell'esecuzione di *test funzionali* nelle condizioni previste dal *piano di validazione*. Le attività di analisi devono cominciare il prima possibile, anche parallelamente al processo di progettazione.

2.2.2.1 Piano di validazione e documentazione

Il piano di validazione deve identificare e descrivere i requisiti e le modalità per l'esecuzione del processo di validazione delle funzioni di sicurezza, la loro categoria e il PL. Deve anche riportare gli strumenti e i mezzi per l'esecuzione della validazione. Il piano di validazione deve definire:

- l'identificazione dei documenti di specifica;
- le condizioni operative e ambientali previste durante i test;
- le analisi e i test da attuare;
- lo standard di riferimento da applicare;
- le persone o le parti responsabili per ogni fase del processo di validazione.

E' fondamentale pianificare il test della SRP/CS per i guasti considerati o rilevati durante la fase di specifica dei requisiti di sicurezza.

La validazione, avendo valenza legale, deve generare dei documenti atti a dimostrare il soddisfacimento dei requisiti di progettazione e di sicurezza. I documenti devono contenere informazioni sufficienti, tra cui:

- specifica dei requisiti di sicurezza per ogni funzione di sicurezza, la categoria richiesta e il PL;
- schemi e specifiche delle parti usate;
- diagrammi a blocchi con descrizione funzionale, diagrammi circuitali comprendenti interfacce e connessioni;
- descrizione delle caratteristiche rilevanti per i componenti validati precedentemente;
- informazioni per l'uso.

Di seguito sono riportate alcune tabelle che riassumono la documentazione necessaria in funzione dalla categoria da raggiungere:

Table 2 — Documentation requirements for categories in respect of performance levels

Documentation requirement	Category for which documentation is required				
	B	1	2	3	4
Basic safety principles	X	X	X	X	X
Expected operating stresses	X	X	X	X	X
Influences of processed material	X	X	X	X	X
Performance during other relevant external influences	X	X	X	X	X
Well-tried components	—	X	—	—	—
Well-tried safety principles	—	X	X	X	X

Table 2 *(continued)*

Documentation requirement	Category for which documentation is required				
	B	1	2	3	4
Mean time to dangerous failure ($MTTF_d$) of each channel	X	X	X	X	X
The check procedure of the safety function(s)	—	—	X	—	—
Diagnostic measures performed, including fault reaction	—	—	X	X	X
Checking intervals, when specified	—	—	X	X	X
Diagnostic coverage (DC_{avg})	—	—	X	X	X
Foreseeable single faults considered in the design and the detection method used	—	—	X	X	X
Common-cause failures (CCF) identified and how to prevent them	—	—	X	X	X
Foreseeable single faults excluded	—	—	—	X	X
Faults to be detected	—	—	X	X	X
How the safety function is maintained in the case of each of the faults	—	—	—	X	X
How the safety function is maintained for each of the combinations of faults	—	—	—	—	X
Measures against systematic faults	X	X	X	X	X
Measures against software faults	X	—	X	X	X
X documentation required — documentation not required					
NOTE The categories are those given in ISO 13849-1:2006.					

2.2.2.2 Validazione mediante analisi

La validazione della SRP/CS deve essere eseguita mediante analisi tenendo conto delle caratteristiche, del PL e dei requisiti della funzione di sicurezza; anche la struttura del sistema e gli aspetti quantificabili come $MTTF_d$, DC_{avg} e CCF, oltre agli aspetti non quantificabili che influenzano il comportamento del sistema devono essere considerati. Le tecniche di analisi sono due: *Top-Down* (1) e *Bottom-Up* (2). Tali tecniche saranno analizzate nello specifico al capitolo 3.

2.2.2.3 Validazione mediante test

Quando la validazione mediante analisi non è conclusiva, si devono attuare dei test per completarla. Quindi, il testing è un'attività complementare all'analisi. La validazione mediante test deve essere pianificata e implementata in maniera logica.

Il piano dei test deve essere elaborato prima dell'inizio dei test includendo le specifiche del test, le condizioni di fallimento/superamento e la cronologia dei test.

Deve essere prodotto un registro dei test contenente il nome della persona incaricata, le condizioni ambientali, la procedura e l'attrezzatura necessaria, la data e il risultato.

E' importante che il test venga eseguito il più vicino possibile alla configurazione di funzionamento finale, collegando tutte le interfacce e i dispositivi periferici.

2.2.2.4 Validazione del software relativo alla sicurezza

La validazione del SRASW e del SRESW deve riportare una descrizione del comportamento funzionale specificato e i criteri di performance durante l'esecuzione nell'hardware designato. Deve essere appurato che le misure del software siano sufficienti per raggiungere il PLr specificato della funzione di sicurezza. E' obbligatorio indicare le misure e le attività implementate durante lo sviluppo per evitare i guasti sistematici.

Il primo passo consiste nel controllo della presenza della documentazione di specifiche e progettazione. La documentazione deve essere revisionata per verificarne la completezza, e l'assenza di interpretazioni errate, omissioni e inconsistenze. In generale, si può considerare il software come una scatola nera o una scatola grigia.

In base al PLr, i test devono includere:

1. test del comportamento funzionale e delle prestazioni in modalità black-box;
2. per PLr d oppure e, casi di test al di fuori dei casi limite previsti;
3. casi di test che simulino guasti determinati analiticamente, con risposta attesa in modo da valutare l'adeguatezza delle misure per il controllo dei guasti.

2.3 EN IEC 61508:2010

La norma EN 61508 *"Sicurezza funzionale dei sistemi di sicurezza elettrici/elettronici/elettronici programmabile"* ha per argomento la sicurezza funzionale dei sistemi E/E/PE indipendentemente dall'applicazione. La norma definisce i requisiti dei sistemi di sicurezza, ponendo attenzione particolare alla parte di controllo, in modo più dettagliato rispetto alla UNI EN ISO 13849.

La norma non è armonizzata per la Direttiva Macchine (o il Regolamento Macchine), ma fornisce delle indicazioni più specifiche che completano le richieste della UNI EN ISO 13849. Tuttavia, è correlata alla norma armonizzata EN 62061:2022(2.4) che verrà trattata successivamente. Pertanto, l'indice di sicurezza utilizzato è il *Livello di Integrità della Sicurezza* o *SIL*. Il SIL è legato al PL nel seguente modo:

PL	SIL
a	nessuna corrispondenza
b	1
c	1
d	2
e	3

La norma è sviluppata in diverse parti. Per lo scopo della presente tesi, verranno analizzate le parti prima (2.3.1), seconda (2.3.2) e terza (2.3.3).

2.3.1 EN 61508-1: Requisiti Generali

La prima parte della norma chiarisce alcuni aspetti relativi alla *verifica* del sistema che svolge funzioni di sicurezza o alle sue sotto-parti.

Lo scopo della verifica è quello di dimostrare che il risultato di ogni fase del ciclo di vita del software sia coerente e conforme a tutti gli aspetti degli obbiettivi e dei requisiti specificati. E' richiesta la definizione di un *piano di verifica* che indichi le attività da compiere contemporaneamente allo sviluppo del software oltre ai criteri di superamento/fallimento, le tecniche o le procedure e gli strumenti necessari per le attività di verifica. [10]

Tutte le informazioni riguardanti le attività di verifica devono essere documentate come prova atta a dimostrare l'esito positivo della verifica.

Quest'ultimo aspetto chiarisce ulteriormente la differenza sostanziale tra *validazione* e *verifica*: la validazione è un processo da implementare soprattutto verso la conclusione del processo, la verifica è necessaria per sostenere lo sviluppo del progetto.

2.3.2 EN 61508-2: Requisiti per sistemi E/E/PE

Gli aspetti rilevanti per lo scopo della tesi, descritti nella seconda parte della norma, seguono strettamente la filosofia *Lean*.[2] Viene chiarito come gestire il fallimento delle attività di verifica o di validazione del sistema.

Quando i risultati della validazione e i risultati attesi non corrispondono è necessario documentare le analisi aggiuntive, le motivazioni del fallimento e il processo di modifica del sistema per raggiungere la specifica in esame.

Similmente, nell'attività di verifica è necessario documentare la motivazione del fallimento della verifica oltre al risultato stesso della verifica. Tuttavia, non è necessario documentare le eventuali modifiche correttive del sistema in quanto tale attività ricade nella fase di progettazione.

[2] La procedura descritta permette di sviluppare un metodo che riduce gli sprechi di tempo e che sia standardizzato. La filosofia *lean* mira ad ottimizzare i processi e i risultati attraverso miglioramenti continui.

2.3.3 EN 61508-3: Requisiti del software

La parte terza è sicuramente la più interessante per la validazione del software. Vengono trattati gli aspetti rilevanti per elaborare il piano di validazione (2.3.3.1), per attuare il processo di validazione (2.3.3.2) e per condurre le attività di verifica (2.3.3.3).

2.3.3.1 Piano di validazione degli aspetti software della sicurezza del sistema

Un piano di validazione sviluppato in modo opportuno è essenziale per implementare il processo di validazione in modo efficace, completo ed efficiente. La pianificazione è necessaria per specificare i vari passi da seguire, per definire le tecniche e le procedure di validazione con lo scopo di dimostrare che il software soddisfi i requisiti di sicurezza. Tra le informazioni formali rilevanti da inserire nel piano di validazione si citano:

- l'indicazione di quando avverrà la validazione;
- gli estremi di chi dovrà effettuare la validazione;
- l'identificazione delle modalità rilevanti per l'attività (setup del test, sequenza di operazioni e condizioni al contorno da considerare);
- le misure e le procedure da utilizzare per dimostrare il soddisfacimento dei requisiti specificati per ciascuna funzione di sicurezza;
- i criteri di superamento/fallimento dei test;
- le politiche e le procedure per valutare i risultati della validazione, soprattutto in caso di fallimento.

La validazione può essere condotta tramite tecniche manuali o automatiche, statiche o dinamiche, analitiche o statistiche. La scelta di una strategia o di un'altra o di una loro combinazione, deve essere opportunamente motivata e documentata. La norma stabilisce che i criteri di superamento/fallimento debbano includere:

- segnali di ingresso con le loro sequenze/combinazioni e i loro valori;
- segnali di uscita con le loro sequenze/combinazioni e i loro valori;
- criteri di accettazione basati su fattori oggettivi o su giudizio esperto.

2.3.3.2 Validazione degli aspetti software della sicurezza del sistema

Analogamente alla UNI ISO 13849, viene ribadito che le parti di software già precedentemente validate non devono essere validate nuovamente. E' tuttavia necessario includere nella documentazione informazioni sufficienti riguardo la validazione di tali parti di software.

Inoltre, in base alla natura del software, la responsabilità della valutazione della conformità può essere assegnata a più parti, mediante opportuna documentazione. In generale, per ogni funzione di sicurezza, la validazione del software deve contenere delle informazioni riguardanti:

- la cronologia delle attività di validazione che permetta di tracciare la sequenza delle attività;
- la versione del piano di validazione da utilizzare;
- le funzioni di sicurezza da validare con gli opportuni riferimenti al piano di validazione;
- i dati di calibrazione delle attrezzature usate;
- i risultati delle attività di validazione, con eventuali discrepanze rispetto ai risultati attesi.

Nel caso del software, la modalità di validazione principale deve essere basata sui test. Analisi e modellazione possono essere impiegate come supporto o attività complementari. Il software deve essere *simulato* con i segnali di ingresso presenti durante il normale funzionamento, simulando gli eventi previsti e le condizioni indesiderate che richiedono un'azione diretta del sistema.

I casi di test, i risultati e le condizioni al contorno devono essere documentati per permettere la ripetibilità e l'analisi dei risultati ottenuti. Infine, i risultati documentati devono indicare che il software ha superato la validazione.

2.3.3.3 Verifica del software

La verifica del software ha lo scopo di testare e valutare ciascuna fase del ciclo di vita, permettendo attività di correzione e/o modifica in modo tempestivo. Ogni verifica del software deve essere documentata indicando il componente verificato e le condizioni nelle quali è stata svolta la verifica.

Tra gli aspetti più importanti riguardanti la documentazione delle attività di verifica è doveroso citare:

- le funzionalità richieste;
- leggibilità da parte del team di sviluppo: la documentazione deve essere chiara e disponibile da ciascuna persona coinvolta nel processo di sviluppo;
- livello di integrità della sicurezza, performance e altri requisiti specificati per la fase.

Devono essere svolte le seguenti attività di verifica:

1. verifica dei requisiti di sicurezza del software;
2. verifica dell'architettura del software;
3. verifica della progettazione del software e dei moduli software;
4. verifica del codice, dei dati e delle performance temporali;
5. test dei moduli software, dell'integrazione;

6. validazione degli aspetti legati alla sicurezza del sistema.

Per verificare i requisiti di sicurezza del software (1) si deve considerare se gli aspetti del piano di validazione soddisfano adeguatamente le specifiche dei requisiti di sicurezza del software e si devono controllare eventuali incompatibilità tra:

- le specifiche dei requisiti del software e le specifiche dei requisiti del sistema;
- le specifiche dei requisiti del software e gli aspetti del piano di validazione.

Per verificare l'architettura del software (2) si deve considerare se la progettazione dell'architettura rispetta i requisiti specificati e se i test di integrazione specificati sono adeguati. Si deve considerare se gli attributi di ogni elemento principale sono adeguati in termini di fattibilità delle prestazioni, della provabilità e della leggibilità da parte del team di verifica. E' doveroso controllare le incompatibilità tra:

- la progettazione dell'architettura del software e le specifiche dei requisiti di sicurezza;
- la progettazione dell'architettura del software e i test di integrazione specificati;
- i testi di integrazione e il piano di validazione.

Le stesse considerazioni, opportunamente contestualizzate, devono essere fatte per la verifica della progettazione del software e dei moduli (3).

La verifica del codice e dei dati (4) deve essere basata su metodi statistici e sullo standard di codifica utilizzato. Le strutture di dati devono essere verificate in termini di consistenza, completezza con i requisiti applicativi, compatibilità con il software di sistema e correttezza dei valori. E' richiesta la verifica di tutti i parametri operativi. La verifica delle performance temporali è basata sulla prevedibilità del comportamento del dominio del tempo.

2.4 EN IEC 62061:2021

La norma armonizzata EN IEC 62061 *"Sicurezza del macchinario - Sicurezza funzionale dei sistemi di comando e controllo relativi alla sicurezza"* prescrive i requisiti e fornisce raccomandazioni per la progettazione, integrazione e la convalida dei sistemi di controllo relativi alla sicurezza per le macchine.

Tra gli aspetti rilevanti alla scopo della tesi è necessario trattare la *specifica di una funzione di sicurezza* (2.4.1), la *classificazione dei software* (2.4.2) e la *validazione* (2.4.3).

2.4.1 Specifiche di una funzione di sicurezza

La norma tratta gli aspetti delle SRP/CS in modo più generico rispetto alla EN IEC 61508, ma fornisce delle indicazioni fondamentali per il processo di validazione. In primo luogo, vengono delineate le specifiche dei requisiti di sicurezza delle funzioni di sicurezza. E' necessario disporre delle informazioni relative al risultato dell'analisi del rischio incluse

le funzioni di sicurezza usate per ridurre il rischio e le caratteristiche operative (tempi di ciclo, tempi di risposta e condizioni ambientali).

La specifica dei requisiti funzionali della funzione di sicurezza deve descrivere il dettaglio di ciascuna funzione tra cui la sua descrizione, le condizioni di attivazione, disattivazione, configurazione, parametrizzazione, la sua priorità, il suo ripristino, il tempo di risposta richiesto, i test e le analisi associate e una descrizione delle condizioni ambientali e operative.

2.4.2 Classificazione dei software

La norma descrive tre livelli diversi del software di applicazione in base al programma principale e al sotto-programma principale:

SW	Main Principle	Subprinciple	Esempio
1	Piattaforma pre-progettata secondo IEC 61508, o altri standard di sicurezza funzionale collegati. Il software applicativo usa il linguaggio LVL.	Software applicativo conforme a questo documento.	PLC Safety con LVL o Relè di sicurezza programmabile. (2.4.2.1)
2	Piattaforma pre-progettata secondo IEC 61508, o altri standard di sicurezza funzionale collegati. Il software applicativo usa un linguaggio diverso dall'LVL.	Software applicativo conforme a questo documento.	PLC Safety con FVL. (2.4.2.2)
3	Piattaforma pre-progettata secondo IEC 61508, o altri standard di sicurezza funzionale collegati. Il software applicativo usa un linguaggio diverso dall'LVL.	Software applicativo conforme alla IEC 61508-3,	PLC Safety con LVL o FVL.

Tabella 2.1: Livelli del software

In base al livello software, sono richiesti dei requisiti specifici.

2.4.2.1 Software - Livello 1

Il software di livello 1 può raggiungere un SIL massimo SIL3. Il ciclo di vita deve essere diviso in fasi distinte includendo attività di gestione e documentazione. Il software ha una complessità ridotta dovuta all'utilizzo di hardware di sicurezza pre-progettato e moduli software. Per tale motivo si può applicare il modello a V semplificato della figura 1.2.

I requisiti di sicurezza per supportare l'attività di sviluppo includono la specifica delle funzioni di sicurezza, la configurazione del SCS, i requisiti di risposta temporale, le modalità operative rilevanti della macchina e le linee guida di codifica.

Le specifiche di progettazione del software stabiliscono che il codice sia strutturato, leggibile, comprensibile, testabile, utilizzabile e facilmente manutenibile. Il codice deve

essere sufficientemente dettagliato da consentire verifiche e test. Il codice deve essere riconducibile alla specifica dei requisiti di sicurezza del software. Inoltre, nelle descrizioni e nei commenti non devono essere presenti termini ambigui o irrilevanti nelle descrizioni.

Quando è appropriato si possono usare metodi semi-formali come tabelle cause-effetto, tabelle logiche, blocchi funzione e diagrammi sequenziali.

Nelle specifiche di progettazione devono essere riportate:

- logica delle funzioni di sicurezza;
- casi di test: valori degli ingressi con i quali condurre il test e i risultati attesi;
- tempi di risposte delle funzioni di sicurezza.

E' raccomandato l'uso di moduli pre-progettati per la realizzazione del software quando possibile. I moduli sviluppati e non validati devono essere sottoposti a test a scatola nera, a scatola grigia o a scatola bianca.

2.4.2.2 Software - Livello 2

Il software di livello 2 può raggiungere un SIL massimo pari a SIL2. La complessità del software è maggiore a causa dell'utilizzo di linguaggio FVL. Per tale motivo si deve far riferimento al modello a V completo di figura 1.1.

E' necessario selezionare un insieme adeguato di strumenti per la configurazione, la simulazione e il testing. L'idoneità del software deve essere dimostrata con analisi per identificare possibili effetti di un guasto provocato da tali strumenti e mediante l'applicazione di misure appropriate per evitare i guasti o per controllarli.

La procedura riguardante le specifiche di sicurezza e le specifiche di progettazione è molto simile a quanto esplicato per il software di livello 1 (2.4.2.1). Il test, invece, include due attività:

1. *Analisi Statica*: analisi della documentazione del software tramite revisioni, ispezioni, walk-through, analisi del controllo di flusso e analisi del flusso di dati;
2. *Test Dinamica*: esecuzione del software in modo controllato e sistematico, in modo da dimostrare la presenza del comportamento richiesto e dell'assenza di comportamenti inattesi. Sono inclusi test funzionali, a scatola nera o a scatola grigia.

In base a questa suddivisione, si osserva che nelle prime fasi del ciclo di vita del software la verifica è prevalentemente statica. I test dinamici diventano possibili solo quando il codice è sviluppato. I risultati di entrambe le attività devono essere accuratamente documentati.

2.4.3 Validazione

2.4.3.1 Principi della validazione

La norma stabilisce che la validazione deve dimostrare il rispetto dei requisiti specificati per le funzioni di sicurezza e del SIL indicato. Quando appropriato, si possono ese-

guire validazioni separatamente prima dell'integrazione, includendo simulazioni con input/output appropriati. Successivamente è importante validare gli effetti dell'integrazione delle parti relative alla sicurezza nel resto del sistema di controllo. In figura 2.1 è riportato un flow-chart molto utile per implementare correttamente il processo di validazione.

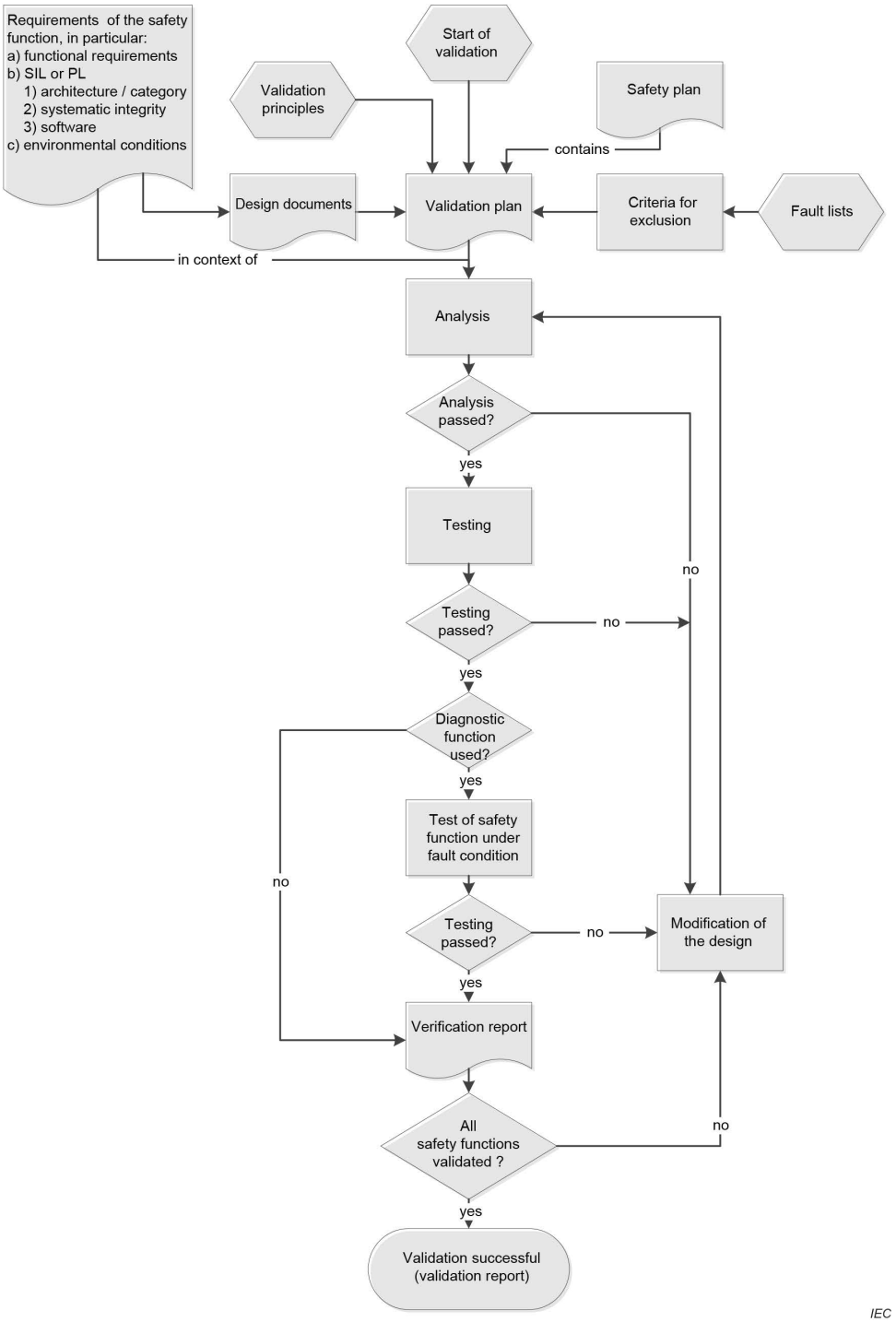


Figura 2.1: Processo di validazione

2.4.3.2 Piano di validazione

Lo scopo del piano di validazione è quello di identificare e descrivere i requisiti per implementare il processo di validazione. Pertanto, deve includere:

- l'identificazione dei documenti di specifica;
- le condizioni operative e ambientali durante i test;
- le analisi e i test da applicare;
- i riferimenti agli standard dei test da applicare;
- l'equipaggiamento necessario.

2.4.3.3 Liste dei guasti

E' importante considerare l'effetto che un eventuale guasto può avere sull'SCS. Si deve stilare una lista di guasti da simulare per verificare che il comportamento del sistema di sicurezza non evolva in uno stato pericoloso. A tal scopo, si può far riferimento alle Appendici dalla A alla D della norma in analisi. Alcuni guasti possono essere esclusi in base all'applicazione se opportunamente giustificato.

2.4.3.4 Analisi come attività di validazione

E' importante selezionare la corretta modalità di analisi da applicare. Le due tecniche principali sono:

1. *Tecnica top-down*: è una tecnica adatta per determinare gli eventi che possono portare agli eventi principali identificati con la loro probabilità. Questa tecnica è usata anche per investigare le conseguenze di guasti multipli;
2. *Tecnica bottom-up*: è una tecnica adatta per investigare le conseguenze dei singoli guasti identificati.

Capitolo 3

Metodo di validazione del software e il software SOFTEMA

In questo capitolo verranno elaborate le richieste delle norme presentate al capitolo 2 per ricavare un metodo di validazione per i SRASW di livello 1 (2.4.2.1). La trattazione sarà completata con l'applicazione del tool SOFTEMA (3.2).

I SRASW di livello 1 devono soddisfare gli stessi requisiti dei SRESW, pertanto le misure da applicare sono:

- Ciclo di vita dello sviluppo con verifica e validazione;
- Documentazione delle specifiche di progettazione;
- Programmazione modulare e strutturata;
- Test funzionali.

E' importante assicurarsi che un errore logico non comporti una situazione pericolosa.

Il metodo di validazione sarà basato sul modello a V semplificato di figura 1.2 e sul processo di validazione rappresentato in figura 2.1. Verrà elaborata una checklist per aiutare il responsabile della validazione a eseguire tutte le fasi in modo logico e ordinato.

3.1 Metodo di validazione

Durante l'analisi del contesto normativo è stato presentato il modello a V completo (figura 1.1). Tramite delle opportune ipotesi, si può semplificare il modello. Infatti, realizzando un SRASW usando combinazioni e interconnessioni di blocchi funzionali già validati e adottando un'architettura a tre stadi, è possibile escludere la fase di progettazione correttiva e la fase di integrazione. Il software e l'hardware in questi casi sono stati testati e certificati direttamente dai produttori. Nella maggior parte dei software relativi alla sicurezza realizzati in ambienti di sviluppo come TIA Portal o PNOZmulti Configurator, queste ipotesi sono rispettate.

3.1.1 Ipotesi e fasi principali del metodo

Il metodo di validazione proposto è basato su alcune ipotesi che permettono di attuare delle semplificazioni, senza omettere alcun aspetto previsto dalle norme. In primo luogo, si suppone che il software applicativo da validare sia già completamente realizzato. Sotto questa ipotesi, si esclude l'attività di progettazione correttiva. Pertanto, l'esito finale della validazione sarà semplicemente positivo o negativo. Questa ipotesi è spesso verificata negli ambiti in cui i costruttori di macchine *custom*[1] sviluppano il proprio codice. La seconda ipotesi presuppone che la documentazione riguardante la specifica dei requisiti di sicurezza del software, la progettazione del software e la specifica delle funzioni di sicurezza sia disponibile. Anche questa ipotesi è spesso verificata nel contesto industriale. Infatti, i team di progettazione e sviluppo compilano la documentazione necessaria a dimostrare la conformità della macchina ai sensi della Direttiva Macchine riportando le informazioni necessarie alla validazione. Sotto queste ipotesi si possono individuare le quattro fasi salienti del metodo di validazione:

A. *Fase preliminare*;

B. *Piano di validazione*;

C. *Analisi e verifiche*;

D. *Test*.

Nella fase A (3.1.2) viene verificata la disponibilità della documentazione necessaria al processo di validazione e si controlla la presenza di incompatibilità tra le specifiche dei requisiti del software e delle funzioni di sicurezza. In questa fase vengono delineati i principi di validazione e viene selezionata la lista guasti da applicare.

Nella fase B (3.1.3) sono processate le informazioni generate dalla fase A per redigere il piano di validazione completo.

Una volta perfezionato il piano di validazione è necessario attuare le attività di analisi e verifica descritte nella fase C (3.1.4).

Infine, nella fase D (3.1.5) il validatore esegue i test previsti dal piano di validazione e compara i risultati con quelli delle attività di analisi descritte nella fase C.

3.1.2 A - Fase preliminare

Il primo passo consiste nel ricavare le informazioni principali delle funzioni di sicurezza tra cui il PLr/SIL richiesto, i tempi di risposta necessari e le descrizioni funzionali. Con queste informazioni si ricava la logica di funzionamento di ciascuna funzione e si definiscono le condizioni di *normale funzionamento*[2]. In questo modo sarà più semplice individuare gli scenari di guasto più probabili o pericolosi.

[1] Si fa riferimento alle macchine o agli impianti sviluppati per un'applicazione molto specifica, che richiedono accorgimenti di progettazione che variano da cliente a cliente

[2] Per normale funzionamento si intende il funzionamento in assenza di guasti secondo le disposizioni su cui si basa la progettazione della funzione di sicurezza. E' escluso l'uso improprio ragionevolmente prevedibile

Successivamente si devono estrapolare le informazioni principali riguardo ai requisiti di sicurezza del software e della sua progettazione tra cui l'architettura, le performance previste, le regole di codifica utilizzate e i moduli integrati. E' importante derivare le informazioni che dimostrano la conformità dei moduli pre-progettatti impiegati. Anche in questo caso, sarà più semplice elaborare degli scenari di guasto e ricavare il comportamento durante il normale funzionamento.

In base alla tecnologia dell'hardware scelto e in base ai componenti con cui la funzione di sicurezza è implementata, è possibile selezionare una lista di guasti adeguata dalle appendici di cui al paragrafo 2.4.3.3. Integrando le informazioni ottenute fino a questo punto con la lista guasti è possibile escludere alcuni scenari di guasto. Le esclusioni possono essere basata sull'architettura del sistema, sul rischio basso che il guasto comporta o sulla ridondanza della funzione di sicurezza. E' obbligatorio motivare ogni esclusione.

La norma UNI EN ISO 13849-1 stabilisce che ogni persona coinvolta nel ciclo di vita di sviluppo del software debba ricevere una descrizione delle funzioni di sicurezza con le modalità operative e il PLr. Lo scopo della documentazione è evitare ogni possibile interpretazione errata. Si deve indicare come attivare e disattivare le funzioni di sicurezza.

3.1.3 B - Piano di validazione

Sfruttando le informazioni acquisite e generate nella *fase preliminare*, si può creare il *piano di validazione*. E' obbligatorio identificare chiaramente i documenti di specifica dei requisiti delle funzioni di sicurezza e del software di sicurezza.

E' necessario formulare una schedulazione delle attività di analisi, verifica e validazione assegnando un riferimento temporale e una persona responsabile. La scelta del team dovrebbe comprendere persone non coinvolte alla progettazione e alla realizzazione del software. Per ogni attività deve essere indicata l'attrezzatura necessaria. E' importante definire le condizioni operative e ambientali nelle quali deve essere eseguita la task.

Nel caso di blocchi software già validati è necessario formulare dei casi di test in cui viene simulato un errore del blocco. Se sono realizzate interconnessioni o combinazioni di blocchi validati, è sufficiente validare la funzione di sicurezza e la parametrizzazione dei moduli. Per ottenere un PL fino a d si devono testare tutte le istruzioni del programma, per ottenere un PL pari a e si devono testare tutti i rami del programma.

Infine, è di fondamentale importanza sviluppare i criteri di superamento/fallimento e le misure e le procedure da applicare. Alla documentazione vanno allegati gli schemi e i diagrammi del sistema con descrizioni funzionali.

Eventuali modifiche del piano di validazione in corso d'opera vanno opportunamente documentate, in quanto il piano è parte integrante della documentazione con valenza legale.

3.1.4 C - Analisi e verifiche

In primo luogo devono essere definite le modalità delle analisi da eseguire documentando le motivazioni. Per strutturare propriamente la verifica è necessario eseguire delle analisi

nelle ipotesi di normale funzionamento e di guasto. Ciò permette di delineare le risposte del sistema realizzato e di confrontarle con i requisiti. In alternativa, l'analisi permette di formulare le performance richieste dal software, affinando i requisiti già esistenti. Durante l'analisi è possibile perfezionare i criteri di superamento/fallimento descritti nel piano di validazione.

Concluse le analisi, si possono cominciare le verifiche. Successivamente, in riferimento alla norma EN IEC 61508:2010, le verifiche da eseguire per la validazione del software sono:

- Verifica dei requisiti di sicurezza del software: è necessario verificare che i requisiti di sicurezza del SRASW siano conformi ai requisiti di sicurezza del sistema e della funzione di sicurezza.
- Verifica dell'architettura del software: la norma UNI EN ISO 13849 stabilisce una architettura a tre stadi. E' importante verificare che il software sia stato implementato rispettando tale requisito.
- Verifica della progettazione del software e dell'integrazione dei moduli: è importante verificare che il software implementato copra tutti i requisiti specificati e che i moduli siano stati integrati in modo corretto[3].
- Verifica del codice e dei dati: occorre esaminare il codice prodotto e determinare se sono state seguire le regole di codifica. La dimensione del codice deve essere la minima possibile e devono essere presenti commenti sufficienti. Inoltre, deve essere leggibile, comprensibile, chiaro e corretto. Per quanto riguarda i dati, si deve appurare che i tipi scelti siano compatibili con l'applicazione e che i range rappresentabili siano adeguati al contesto. Alle variabili globali, agli input e agli output deve essere assegnato un nome univoco che ne indichi la funzionalità. Le uscite devono essere attuate una sola volta per ciclo.
- Verifica delle performance temporali: uno degli aspetti più importanti nell'ambito della sicurezza è il determinismo temporale. Si deve verificare che i tempi di risposta effettivi corrispondano con quelli previsti o specificati.

Concluse le attività di analisi e verifica, possono essere elaborati dei casi di test che simulino guasti aggiuntivi basati sulle criticità individuate.

Ogni task deve essere documentata opportunamente, in modo da tracciare l'intero processo. Prima di cominciare i test è buona pratica revisionare la documentazione prodotta fino a questo punto.

3.1.5 D - Test

L'esecuzione dei test è finalizzata a confermare che il sistema nella sua configurazione *finale* sia conforme alle specifiche dei requisiti di sicurezza e progettazione. Pertanto, è

[3] Diversi ambienti di sviluppo, come ad esempio TIA Portal, forniscono la documentazione necessaria per l'uso e l'integrazione dei moduli pre-progettati.

di fondamentale importanza attuare dei casi di test che trattino tutti gli aspetti rilevati con le analisi precedenti.

L'impostazione di una cronologia dei test prima di avviare l'attività aiuta a visualizzare tutti gli aspetti considerati dai test. La cronologia, oltre al riferimento temporale dell'esecuzione del test, deve riportare l'attrezzatura necessaria. Per ogni attrezzatura devono essere documentati i dati di calibrazione e di configurazione.

La norma UNI EN ISO 13849-1 fa riferimento ai test funzionali, mentre la norma UNI EN ISO 13849-2 fa riferimento ai test funzionali estesi. E' necessario testare i software secondo entrambe le modalità. Prima dell'esecuzione dei test è necessario visionare la documentazione contenente le specifiche dei requisiti di progettazione e sicurezza del software.

Il metodo per condurre i *test funzionali* è il black-box. E' richiesto di verificare i requisiti di prestazione del software. E' opportuno individuare dei casi di test che permettano il completamento in una seconda fase. E' obbligatorio il test I/O per garantire che tutti i segnali correlati alla sicurezza siano usati correttamente.

Per condurre i *test funzionali estesi* è possibile ricorrere ai metodi black-box, gry-box o white-box. Lo scopo di queste prove è scoprire gli errori di sviluppo. Devono essere applicati vettori di input poco probabili o non specificati nella documentazione e verificare che il sistema evolva in uno stato indefinito o pericoloso. Nel caso in cui siano presenti segnali analogici, è auspicabile l'esecuzione di test in casi limite come divisioni per zero, array pieni, elementi vuoti o overflow. E' consigliato prevedere dei casi di test in cui l'output è forzato oltre i limiti previsti.

Si possono quindi eseguire le prove nei casi di test previsti e nelle condizioni di guasto riportate nella lista guasti. Il risultato di ogni prova deve essere documentato e, in caso di esito negativo, va motivato il fallimento. Gli scenari di test che possono danneggiare il sistema possono essere simulati opportunamente, documentando il risultato.

Infine, è importante revisionare la documentazione prodotta.

3.1.5.1 Risultato della validazione

Se tutte le verifiche e i test hanno avuto esito positivo, la validazione è conclusa positivamente e si può raggruppare la documentazione prodotta. La documentazione del software dovrebbe essere divisa in sezioni dedicate ai moduli o alle funzioni di sicurezza. Si deve indicare la funzione svolta, la descrizione degli ingressi e delle uscite, il programmatore, la versione della libreria e i commenti alle istruzioni. Può essere resa disponibile in formato cartaceo o digitale. E' obbligatorio inserire le informazioni riguardanti l'esecuzione della validazione.

La documentazione deve essere completa, comprensibile, leggibile e disponibile. Tutte le persone coinvolte nel ciclo di vita devono poter accedere alla documentazione.

3.2 SOFTEMA

Nonostante il processo di validazione sia stato semplificato da alcune ipotesi, è comunque molto articolato e complesso. Per migliorare l'efficienza e l'efficacia del processo è utile usufruire di uno strumento in grado di guidare il responsabile della validazione. Per questa finalità è stato sviluppato SOFTEMA.

3.2.1 Il metodo a matrice IFA

L'*Istituto per la Sicurezza sul Lavoro (IFA)* tedesco e l'*Assicurazione obbligatoria contro gli infortuni (DGUV)* tedesca hanno collaborato nel progetto *FF-FP0319* per dimostrare lo sviluppo e la documentazione del SRASW secondo gli standard richiesti. Il risultato della ricerca è il *metodo a matrice IFA*.

Per poter applicare questo metodo è necessario che il SRASW sia sviluppato secondo l'architettura a tre stadi[4]. E' basato sul modello a V semplificato della norma UNI EN ISO 13849-1.

Il metodo a matrice è applicato per specificare e documentare il software in modo strutturato attraverso l'uso di tabelle, generalmente implementate su Excel. L'utente può specificare la relazione ingresso-uscita compilando la matrice C+E (Causa-Effetto) per facilitare la verifica e la validazione dei requisiti. I passi principali del metodo sono i seguenti:

1. Definizione delle funzioni di sicurezza.
2. Documentazione dei nomi e degli indirizzi delle variabili.
3. Selezione delle misure di prevenzione degli errori.
4. Determinazione dei requisiti normativi.
5. Documentazione dell'architettura delle funzioni di sicurezza.
6. Documentazione dell'architettura dei moduli utilizzati.
7. Creazione matrice C+E.
8. Revisione del codice.
9. Validazione del software.

3.2.2 Introduzione a SOFTEMA

I costruttori di macchine utilizzano sempre più spesso i SRASW per implementare funzioni di sicurezza. Come visto al capitolo 2, le due norme armonizzate principali sono la UNI EN ISO 13849 e la EN IEC 62061. Il *metodo a matrice IFA* può essere utilizzato per specificare, validare e documentare il software applicativo conformemente alle normative

[4] Ingresso => Elaborazione => Uscita.

di riferimento. Per implementare questo metodo in modo semplice e garantendo la qualità è stato sviluppato il software gratuito SOFTEMA. Il software si basa sui fogli di lavoro Excel.

SOFTEMA è uno strumento molto avanzato e molto articolato. Inoltre, al momento della scrittura della presente tesi, è disponibile esclusivamente la versione in tedesco. Per acquisire familiarità con le interfacce è fondamentale conoscere e comprendere le norme di riferimento. L'ordine delle tabelle utilizzate nel software segue l'iter specificato dal modello a V semplificato. Inoltre, esiste un concetto di gestione degli utenti che permette a ciascun membro del team di accedere solo alle funzione necessarie per il proprio ruolo. La struttura delle schede segue strettamente il modello a V semplificato.

Nei paragrafi successivi verranno illustrati gli aspetti pratici per creare un progetto relativo ad un SRASW.

3.2.2.1 Impostazione di un progetto

Dopo aver aperto SOFTEMA è necessario impostare un nuovo progetto. Dal menù "Datei" si deve selezionare `__SOFTEMA_Template__.xlsx` dalla scheda "Wiederherstellen". Al momento del salvataggio nella directory designata è importante aggiungere `.xlsx` al nome del file, altrimenti SOFTEMA non sarà in grado di aprire il file creato.

Una volta creato il file è necessario aprirlo con il comando "Offnen" nel menù "Datei". Verranno visualizzate una serie di tabelle non modificabili. Il primo passo, molto importante, è la definizione del team di sviluppo e validazione. Si deve selezionare la funzione "Benutzerverwaltung" nel menù "Extras". La password predefinita per l'amministratore è "admin". Inserendo la password, verrà richiesto di impostarne una nuova e di scegliere una domanda di sicurezza. Conclusa questa operazione, verrà chiesto di inserire le nuove credenziali di amministratore.

Inserendo le credenziali si sbloccano le funzionalità di gestione del team di lavoro. Alla scheda "Neue Benutzer" si possono definire le generalità dei membri del team indicando nome e azienda.

E' importante definire il ruolo di ciascun membro. Alla scheda "Benutzer verwalten" è possibile selezionare ogni persona e assegnarle uno dei seguenti ruoli:

- *Projektleiten*: responsabile del progetto.
- *Projektieren*: progettista.
- *Inbetriebnehmen*: responsabile della messa in servizio.
- *Validieren*: responsabile della validazione.
- *Prüfen 1*: controllo 1.
- *Prüfen 2*: controllo 2.
- *Superuser*: è un membro che può svolgere tutte i ruoli precedenti.

Ora ogni membro può accedere al progetto tramite il menù "Login/Logout".

3.2.2.2 Tabelle principali

Il processo di validazione basato su SOFTEMA prevede la compilazione di alcune tabelle che seguono strettamente l'iter descritto dalla norma UNI EN ISO 13849. Alcuni aspetti trattati dal metodo presentato al paragrafo 3.1 non sono presenti in queste tabelle. E' auspicabile che tali aspetti vengano comunque considerati in modo da completare l'attività di validazione.

Come illustrato in figura 3.1, le tabelle di SOFTEMA sono dodici:

- *Projekt*: sono riassunte le informazioni principali del progetto, incluse le assegnazioni dei ruoli ai vari membri del team. E' possibile dichiarare il collegamento alla documentazione preliminare su cui si basa il progetto.
- *A1 - Sicherheitsfunktionen*: nella tabella vengono dichiarate le funzioni di sicurezza implementate dal software, indicando il nome identificativo, la priorità, il PLr e gli I/O interessati.
- *A2.4 - IO-Liste*: rappresenta la lista degli ingressi e delle uscite dell'unità logica utilizzata. E' importante definire un simbolo univoco e assegnare l'indirizzo fisico della variabile.
- *A3 - Maßnahmen*: in questa tabella è riportato un elenco delle misure richieste dalla norma UNI EN ISO 13849 applicabili nel caso del SRASW. E' importante escludere le misure non applicate a causa del PLr o di altre norme seguite.
- *A4 - Anforderungen*: viene riportato un elenco delle misure applicabili nello sviluppo del software, suddivisi in base alla fase del ciclo di vita e al PL applicabile. Corrispondono ai requisiti trattati dalle norme presentate al capitolo 2.
- *B3 - Modularchitektur*: si riportano le architetture dei moduli utilizzati, indicando ingressi, uscite e parametri. E' importante assegnare un nome all'istanza del blocco utilizzato. E' possibile utilizzare il "Modul-Manager" per inserire in modo comodo i moduli sviluppati su misura.
- *B4 - Matrix C+E*: è la tabella più importante del software. Viene rappresentata la relazione logica tra ingressi e uscite di sicurezza. Si possono definire i vettori di ingressi per eseguire i test specificati nel piano di validazione, riportando i valori attesi delle uscite.
- *B4 - Matrix kompakt*: riassume le informazioni presenti nella scheda B4. E' utile per visualizzare velocemente il comportamento delle uscite e lo stato della verifica e della validazione.
- *C1 - Codereview*: sono riassunti gli stati delle verifiche da eseguire sul codice, suddivise in base alla scheda di interesse.
- *D1 - Validierung*: vengono riportati gli stati delle validazioni in base alla tabella di riferimento. E' necessario compilare le ultime righe per ultimare la validazione.

- *Änderungen*: in questa tabella si può creare un registro delle modifiche del software o del progetto.
- *Protokoll*: costituisce un registro dei login al progetto, in modo da tracciare le attività svolte.

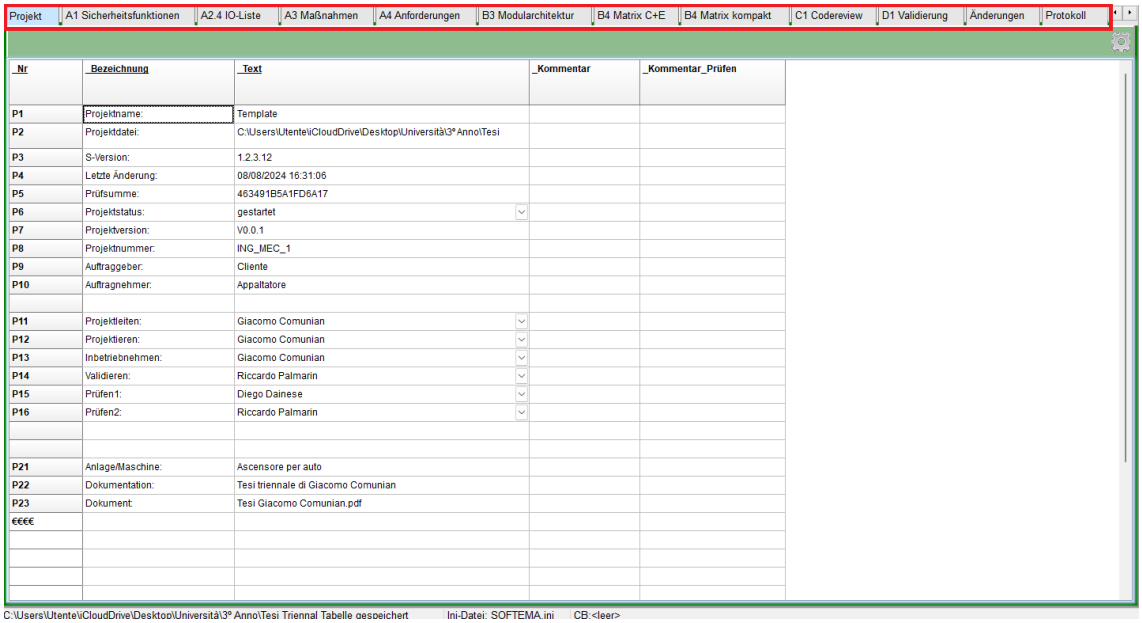


Figura 3.1: Tabelle principali nel tool SOFTEMA

Utilizzando opportunamente le tabelle si possono automatizzare le varie fasi dei processi di progettazione e validazione. L'utilizzo e la compilazione delle varie voci saranno ampiamente trattate nel capitolo 4 tramite un esempio applicativo della validazione.

Capitolo 4

Validazione di un software legato alla sicurezza con SOFTEMA

Nel corso del capitolo verrà presentato il processo di validazione del software di sicurezza sviluppato nella tesi di laurea triennale di Giacomo Comunian. Il software gestisce le funzioni di sicurezza di un prototipo di ascensore per automobili. Il sistema è stato progettato e realizzato, e la validazione è stata eseguita utilizzando il tool SISTEMA,con particolare attenzione agli aspetti dell'hardware del sistema. Tuttavia, il software non è stato validato secondo i metodi indicati dalle norme di riferimento presentate al capitolo 2.

Per condurre la validazione, si farà riferimento al metodo sviluppato al capitolo 3, applicando le funzionalità offerte da SOFTEMA. La tesi di Giacomo Comunian verrà utilizzata come la documentazione principale per effettuare la validazione.

4.1 Impostazione del progetto in SOFTEMA

Innanzitutto è necessario definire i membri del team del progetto e assegnare i rispettivi ruoli. Nel caso in specifico, il redattore della presente tesi riveste i ruoli di *responsabile del progetto*, di *validatore* e si occupa del primo controllo. Giacomo Comunian è designato come *progettista* e *responsabile della messa in servizio*. L'ultimo controllo, se necessario, è responsabilità del relatore.

Con questa suddivisione dei ruoli viene rispettato il principio di separazione dell'attività di validazione con l'attività di sviluppo del software[1] sancito dalla norma UNI EN ISO 13849.

In figura 4.1 sono riportati i dati sommari del progetto, oltre alla suddivisione dei ruoli. Nelle righe `P22` e `P23`, il PDF della tesi triennale di Giacomo Comunian viene ufficialmente riconosciuto come documentazione del progetto per le specifiche di sicurezza e progettazione, soddisfacendo così il requisito della UNI EN ISO 13849-2 che prevede di identificare i documenti di specifica.

[1] La norma UNI EN ISO 13849-1 raccomanda fortemente che il team addetto alla validazione non sia coinvolto fasi del ciclo di vita del software.

Nr	Bezeichnung	Text	Kommentar
P1	Projektname:	Validazione del software di sicurezza	
P2	Projektdatei:	C:\Users\Utente\iCloudDrive\Desktop\Univ	
P3	S-Version:	1.2.3.12	
P4	Letzte Änderung:	11/08/2024 12:21:27	
P5	Prüfsumme:	5666FC09A1FD6A17	
P6	Projektstatus:	gestartet	
P7	Projektversion:	V1.0.1	Versione
P8	Projektnummer:	Tesi_Triennale	Numero
P9	Auftraggeber:	UniPD	Cliente
P10	Auftragnehmer:	UniPD	Appaltatore
P11	Projektleiten:	Riccardo Palmarin	Resp. progetto
P12	Projektieren:	Giacomo Comunian	Progettista
P13	Inbetriebnehmen:	Giacomo Comunian	Resp.
P14	Validieren:	Riccardo Palmarin	Validatore
P15	Prüfen1:	Riccardo Palmarin	Controllo 1
P16	Prüfen2:	Diego Dainese	Controllo 2
P21	Anlage/Maschine:	Ascensore per auto	Macchina/impia
P22	Dokumentation:	Tesi triennale di Giacomo Comunian	Rif.
P23	Dokument:	Tesi Giacomo Comunian.pdf	Link

Figura 4.1: Impostazione del progetto in SOFTEMA.

4.2 A - Fase preliminare

Nel corso della fase preliminare si devono ricavare tutte le informazioni preliminari necessarie ad impostare la validazione in modo corretto. La prima parte di questo processo consiste nell'analizzare la documentazione del sistema per ricavare le funzioni di sicurezza implementate e i loro requisiti. La seconda parte prevede di compilare le tabelle designate su SOFTEMA.

4.2.1 Funzioni di sicurezza

Dalla documentazione[2] si possono ricavare le funzioni di sicurezza implementate:

- Arresto di emergenza.
- Arresto per intervento dei finecorsa.
- Arresto per intervento delle barriere fotoelettriche di sicurezza.

Secondo l'analisi del rischio, il PLr richiesto per tutte le funzioni di sicurezza è *d*. A livello hardware, è stato dimostrato il raggiungimento del PLr utilizzando il tool SISTEMA.

L'attivazione di ciascuna funzione di sicurezza può essere riconosciuta tramite l'azione sul pulsante `Sb3`. Di seguito si analizzano nello specifico le singole funzioni di sicurezza.

Nomenclatura funzioni di sicurezza: la documentazione del tool SOFTEMA consiglia un sistema di nomenclatura delle funzioni di sicurezza che verrà utilizzato nella trattazione. Le funzioni di sicurezza saranno individuate da un nome univoco con la seguente struttura: `-SFXX.Y.Z`, dove `SF` rappresenta una *funzione di sicurezza*, `XX` identifica la funzione di sicurezza, `Y` indica il dispositivo di sicurezza e `Z` descrive l'effetto sulle uscite

[2] Pagina 56 della tesi di Giacomo Comunian.

dell'attivazione della funzione. La nomenclatura presentata in questa fase sarà utilizzata nell'interfaccia di SOFTEMA.

4.2.1.1 -SF10.1.1: Arresto di emergenza

Il pulsante di emergenza `Sb0` è dotato di 2 contatti in parallelo che agiscono su due ingressi distinti del PLC. In figura 4.2 è riportata l'associazione tra la nomenclatura dei contatti e l'indirizzo hardware del PLC.

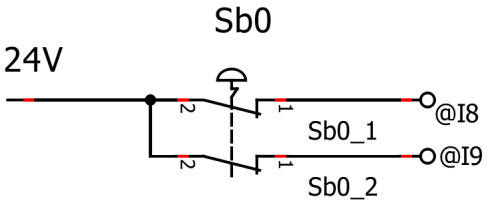


Figura 4.2: Collegamento del pulsante di emergenza al PLC

Quando uno dei due contatti `Sb0_1` o `Sb0_2` assume uno stato logico basso, il motore della pompa deve essere fermato. E' possibile che uno dei conduttori delle linee di ingresso subisca un guasto a causa della disconnessione fisica o di una dispersione verso terra di uno dei conduttori. In tal caso, sui due ingressi `I8` e `I9` sono presenti 2 stati diversi. Questa situazione deve essere segnalata la condizione di guasto e il motore deve essere fermato.

4.2.1.2 -SF20.1.1: Arresto per intervento della barriera di sicurezza al piano terra

La barriera fotoelettrica al piano terra è identificata con la nomenclatura `BFT1`. E' dotata di due uscite, `BFT1_1` e `BFT1_2`, per segnalare l'assenza di oggetti nel fascio infrarosso, associate a due ingressi del PLC secondo lo schema riportato in figura 4.3.

Anche in questo caso il PLC deve ricevere lo stesso stato logico su entrambi gli ingressi; in caso contrario, si verifica una condizione di guasto. La barriera indica lo stato sicuro mediante uno stato logico alto su entrambe le uscite. Se tale condizione non è verifica, il motore deve essere fermato.

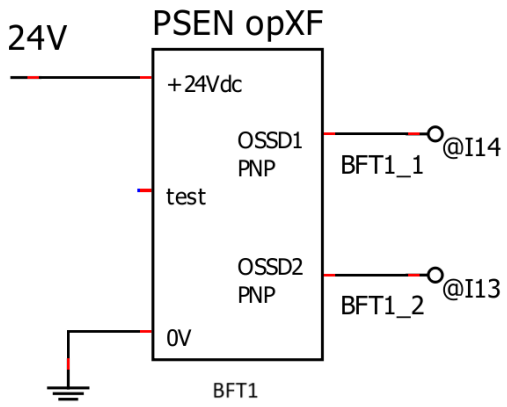


Figura 4.3: Collegamento della barriera di sicurezza piano terra al PLC

4.2.1.3 -SF20.2.1: Arresto per intervento della barriera di sicurezza al primo piano

Per la barriera di sicurezza al primo piano, indicata con `BFT2`, valgono le stesse considerazioni fatte al paragrafo precedente. Lo schema riportato in figura 4.4 illustra l'associazione tra le uscite della barriera ricevente, `BFT2_1` e `BFT2_2`, e gli ingressi, `I15` e `IM16`, del PLC.

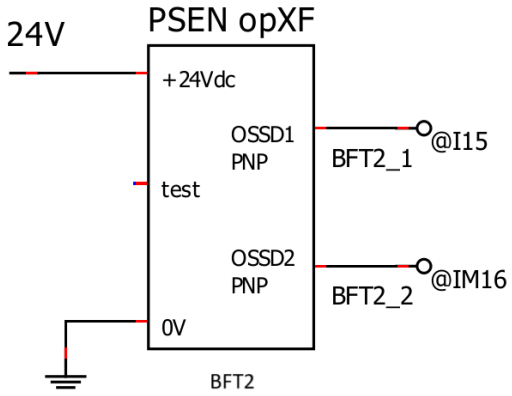


Figura 4.4: Collegamento della barriera di sicurezza primo piano al PLC

4.2.1.4 -SF30.1.1: Arresto per intervento dei finecorsa al piano terra

Al piano terra sono disposti due finecorsa, `FC3` e `FC4`, collegati a due differenti ingressi, `I6` e `I7`, del PLC secondo lo schema riportato in figura 4.5.

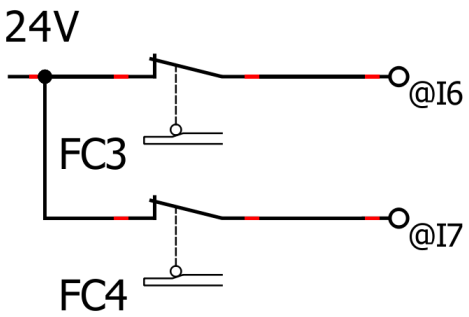


Figura 4.5: Collegamento dei finecorsa piano terra al PLC

I finecorsa devono trasmettere lo stesso stato logico agli ingressi del PLC salvo condizione di guasto. Quando sono attivati, impediscono al motore della pompa di far proseguire la discesa dell'ascensore.

4.2.1.5 -SF30.2.1: Arresto per intervento dei finecorsa al primo piano

Al primo piano sono disposti altri due finecorsa, `FC1` e `FC2`, collegati a due ingressi del PLC, `I4` e `I5`, del PLC secondo lo schema riportato nella figura 4.6.

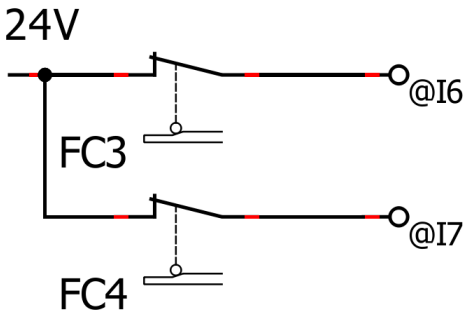


Figura 4.6: Collegamento dei finecorsa primo piano al PLC

I finecorsa nel normale funzionamento devono inviare lo stesso stato logico agli ingressi del PLC. Quando sono azionati, il motore della pompa non può far continuare la salita dell'ascensore.

4.2.2 Informazioni aggiuntive

4.2.2.1 Lista dei guasti

Si possono individurare i primi scenari di guasto basati sull'hardware scelto. La norma UNI EN ISO 13849-2, all'appendice D [1], consiglia di considerare i seguenti guasti di natura elettrica ed elettromeccanica:

1. Cortocircuito tra contatti adiacenti isolati;

2. Contatto impedito all'apertura/chiusura;

3. Il teleruttore non si attiva/disattiva;

4. Apertura di ciascuna connessione;

5. Guasti in ciascuna parte della funzione, inclusi guasti software;

La norma prevede di escludere il guasto (1). L'unico dispositivo dotato di contatti adiacenti isolati è il fungo di emergenza. In caso di cortocircuito, il componente mantiene la sua funzionalità, poiché è in grado di mantenere e modificare il suo stato logico. La ridondanza integrata della funzione di sicurezza consente di escludere tale scenario di guasto.

4.2.2.2 Regole di codifica

Dalla tabella F.2 dell'allegato F [22], norma EN IEC 62061:2022, si ricavano alcune indicazioni riguardanti la codifica. Il nome delle variabili deve permette una distinzione immediata tra quelle relative alla sicurezza e quelle non correlate alla sicurezza. Inoltre dovrebbero essere autoesplicativi, ad esempio contenendo il nome o l'identificativo del dispositivo in considerazione. Ogni dichiarazione deve essere corredata da un commento.

E' molto importante che le uscite siano assegnate una sola volta nel programma. Inoltre, ogni riga di codice o ogni rete deve essere accompagnata da un commento specifico.

4.2.3 Aggiornamento tabelle SOFTEMA

4.2.3.1 A1 - Specifica delle funzioni di sicurezza

Con le informazioni ottenute al paragrafo 4.2.1 si può compilare la tabella *A1 - Specifica delle funzioni di sicurezza*. La tabella è dotata di dodici colonne da compilare. Ogni riga corrisponde ad una specifica funzione di sicurezza.

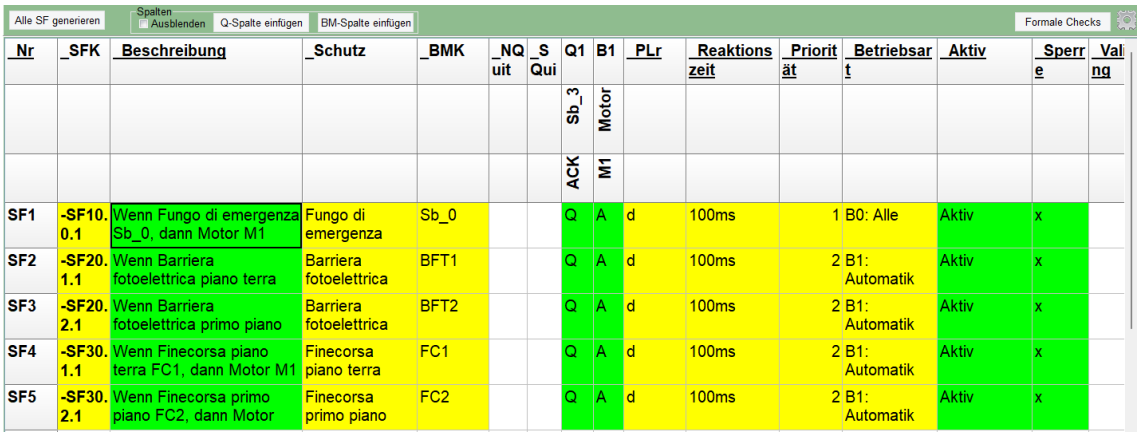
Nella colonna `SFK` si deve indicare il nome univoco della funzione di sicurezza. La colonna `Beschreibung` verrà compilata da SOFTEMA stesso. Le colonne `Schutz` e `BMK` indicano rispettivamente la descrizione del componente di sicurezza e il suo identificativo. Nella colonna `Q1` si deve indicare quale componente è utilizzato per riconoscere l'attivazione della funzione di sicurezza; nella colonna `B1` si deve selezionare l'effetto che la funzione di sicurezza ha sull'attuatore. Successivamente si deve indicare il `PLr` della funzione e il tempo di reazione. Nelle colonne `Prioritat` e `Betriebsart` di deve indicare rispettivamente la priorità della funzione e la modalità di funzionamento.

Si consiglia di assegnare la priorità 1 e la modalità di funzionamento *B0* esclusivamente alle funzioni di arresto di emergenza. Per le altre funzioni di sicurezza si consiglia

la modalità di funzionamento *B1*. Il tempo di reazione scelto per l'applicazione è pari a 100ms[3].

Dopo aver compilato le colonne si consiglia di eseguire un controllo formale per rilevare eventuali errori di compilazione o informazioni mancanti agendo sul controllo `Formale Checks` in alto a destra . Se il controllo va a buon fine, è necessario cliccare sul pulsante `Alle SF generieren` per compilare la colonna `Beschreibung` e confermare le funzioni di sicurezza selezionate.

In figura 4.7 si può visionare la compilazione specifica per il caso studio.



Nr	_SFK	Beschreibung	_Schutz	_BMK	_NQuit	_SQui	Q1	B1	PLr	Reaktionszeit	Priorität	Betriebsart	Aktiv	Sperre	Validierung
							Sb_3	Motor							
							ACK	M1							
SF1	-SF10.0.1	Wenn Fungo di emergenza Sb_0, dann Motor M1	Fungo di emergenza	Sb_0			Q	A	d	100ms	1	B0: Alle	Aktiv	x	
SF2	-SF20.1.1	Wenn Barriera fotoelettrica piano terra	Barriera fotoelettrica	BFT1			Q	A	d	100ms	2	B1: Automatik	Aktiv	x	
SF3	-SF20.2.1	Wenn Barriera fotoelettrica primo piano	Barriera fotoelettrica	BFT2			Q	A	d	100ms	2	B1: Automatik	Aktiv	x	
SF4	-SF30.1.1	Wenn Finecorsa piano terra FC1, dann Motor M1	Finecorsa piano terra	FC1			Q	A	d	100ms	2	B1: Automatik	Aktiv	x	
SF5	-SF30.2.1	Wenn Finecorsa primo piano FC2, dann Motor	Finecorsa primo piano	FC2			Q	A	d	100ms	2	B1: Automatik	Aktiv	x	

Figura 4.7: Compilazione della tabella A1 in SOFTEMA

La colonna `Aktiv` consente di attivare o disattivare la funzione all'interno del progetto, mentre la colonna `Sperre` permette di bloccare le modifiche delle righe. Infine, la colonna `Validerung` è dedicata alla fase di validazione e verifica.

4.2.3.2 A2.4 - Lista ingressi e uscite

Per completare la compilazione delle informazioni fin qui ricavate, si deve compilare la tabella relativa alla lista I/O. Ad ogni ingresso/uscita si deve associare un simbolo univoco e si deve indicare l'indirizzo hardware. E' importante definire il tipo della variabile e inserire una descrizione. Si compilano quattro colonne. Nella colonna `Beschreibung` viene riportata la descrizione della variabile, nelle colonne `Symbol` e `Adresse` devono essere specificati i simboli e gli indirizzi hardware. Nella colonna `Datentyp` si deve selezionare il tipo di dati utilizzato nella programmazione. In figura 4.8 è riportata la compilazione della tabella.

[3] Il PLC esegue un controllo della simultaneità dei canali di ciascuna funzione di sicurezza entro un tempo di 50ms. Gli ulteriori 50ms considerano eventuali tempi di assestamento degli attuatori.

_Nr	_Beschreibung	_Symbo	_Adresse	_Datentyp	_Modul	_Aktiv_in...	_Aktiv	_Spe...	_SW-Verif.	_IO-Test	_DIAG-Test
	Eingänge										
I1	Finecorsa salita n1 (NC)	FC1	Mb0.I4	SF_BOOL		Aktiv	Aktiv	o			
I2	Finecorsa salita n2 (NC)	FC2	Mb0.I5	SF_BOOL		Aktiv	Aktiv	o			
I3	Finecorsa discesa n1 (NC)	FC3	Mb0.I6	SF_BOOL		Aktiv	Aktiv	o			
I4	Finecorsa discesa n2 (NC)	FC4	Mb0.I7	SF_BOOL		Aktiv	Aktiv	o			
I5	Pulsante di emergenza contatto n1 (NC)	Sb0_1	Mb0.I8	SF_BOOL		Aktiv	Aktiv	o			
I6	Pulsante di emergenza contatto n2 (NC)	Sb0_2	Mb0.I9	SF_BOOL		Aktiv	Aktiv	o			
I7	Pulsante salita (NO)	Sb_1	Mb0.I10	SF_BOOL		Aktiv	Aktiv	o			
I8	Pulsante discesa (NO)	Sb_2	Mb0.I11	SF_BOOL		Aktiv	Aktiv	o			
I9	Pulsante riarmo (NO)	Sb_3	Mb0.I12	SF_BOOL		Aktiv	Aktiv	o			
I10	Barriera di sicurezza PT contatto n2 (NC)	BFT_1_2	Mb0.I13	SF_BOOL		Aktiv	Aktiv	o			
I11	Barriera di sicurezza PT contatto n1 (NC)	BFT_1_1	Mb0.I14	SF_BOOL		Aktiv	Aktiv	o			
I12	Barriera di sicurezza P1 contatto n1 (NC)	BFT_2_1	Mb0.I15	SF_BOOL		Aktiv	Aktiv	o			
I13	Barriera di sicurezza P1 contatto n2 (NC)	BFT_2_2	Mb0.IM16	SF_BOOL		Aktiv	Aktiv	o			
	Ausgänge										
O1	Teleruttore salita n1	KM1	Mb0.O0	BOOL		Aktiv	Aktiv	o			
O2	Teleruttore salita n2	KM2	Mb0.O1	BOOL		Aktiv	Aktiv	o			
O3	Teleruttore discesa n1	KM3	Mb0.O2	BOOL		Aktiv	Aktiv	o			
O4	Teleruttore discesa n2	KM4	Mb0.O3	BOOL		Aktiv	Aktiv	o			

Figura 4.8: Compilazione parziale della tabella A2 in SOFTEMA

La colonna `Modul` può essere aggiornata solo successivamente alla compilazione della tabella B3. Mediante i controlli `Aktiv in C+E` e `Aktiv` si possono escludere gli ingressi/uscite dalla tabella Causa-Effetto o dall'intero progetto. Si consiglia di disattivare gli ingressi che non legati alla sicurezza ma che sono cablati con il PLC safety.

4.2.3.3 A3 - Misure applicabili al software

Nella tabella A3 sono riportate 44 misure applicabili al SRASW, estratte dalla norma UNI EN ISO 13849-1. E' importante disattivare le misure non pertinenti all'applicazione, sia per incompatibilità con il sistema che per l'adozione di ulteriori misure seguite. E' possibile aggiungere ulteriori misure relative ad altre norme di riferimento. Nel caso dei SRASW programmati in LVL, le misure indicate sono generalmente sufficienti. Tuttavia, sarà necessario adattare le informazioni specifiche del produttore del sistema di sicurezza. Per impostazione predefinita, SOFTEMA riporta le informazioni relative alle CPU-Safety di Siemens.

La tabella raggruppa le varie misure secondo le seguenti categorie:

- Misure generali:
  - Nomenclatura delle variabili: ingressi, uscite e variabili interne;
  - Commenti;
  - Elaborazione del segnale;
  - Sviluppo dei propri blocchi funzione;
  - Attività dopo le modifiche;
  - Documentazione;
  - Varie.
- Misure specifiche:

- Editor del programma;
- Elaborazione del segnale.

Secondo la norma EN IEC 62061, per i SRASW a complessità limitata è ammessa l'omissione della regola di programmazione che richiede una nomenclatura differenziata per variabili correlate e non-correlate alla sicurezza. Per tale motivo si possono disattivare le righe da `R1` a `R6`.

Le richieste inerenti ai commenti non possono essere disattivate, poiché sono richieste indipendentemente dall'applicazione.

I teleruttori scelti per l'applicazione non permettono l'implementazione di un ramo di feedback, pertanto si possono escludere le misure `R15` e `R16`.

Poiché nel programma non sono stati sviluppati dei blocchi funzione, si possono disattivare le righe da `R19` a `R22`. Inoltre, come riportato nelle ipotesi iniziali, non sono previste attività di modifica. Per tale motivo si escludono le misure `R23` e `R24`.

Alla sezione dell'editor del programma si deve indicare l'editor utilizzato (`R32`), nel caso in istanza PNOZmulti Configurator 11.0.4, e si deve selezionare il linguaggio di programmazione impiegato (`R33`), in questo caso linguaggio a blocchi funzionali o FBD.

4.2.3.4 A4 - Requisiti normativi

Nella tabella A4 sono riportati 39 requisiti per il SRASW indicando per quale PLr devono essere applicati. Nel caso in esame, il PLr per tutte le funzioni di sicurezza è `d`. Si possono escludere i requisiti che si applicano esclusivamente PLr e. Nella fattispecie si può disattivare il requisito in riga `A11`.

Si possono escludere anche i requisiti relativi alle modifiche del software, in quanto non previste. Pertanto si disattivano i requisiti alle righe `A5` e `A39`. La colonna `Validierung` sarà usata durante la fase di validazione per riconoscere quali di questi requisiti sono stati rispettati.

In questo caso, non è necessario aggiungere ulteriori requisiti, poiché la scheda è sufficientemente dettagliata.

4.3 B - Piano di validazione

Il piano di validazione, oltre a essere parte integrante della documentazione, assume un'importante connotazione organizzativa.

La norma EN IEC 62061 sancisce che per il raggiungimento del SIL 3, equivalente al PLr d, è necessario che a condurre le attività di revisione, verifica, testing e validazione sia una persona indipendente dalle fasi di progettazione e realizzazione. In base all'assegnazione dei ruoli del paragrafo 4.1 si può affermare che il requisito è rispettato.

Ciascuna funzione di sicurezza è stata programmando interconnettendo blocchi funzione sviluppati e validati da Pilz. L'attività di analisi, verifica e validazione dovrà dimostrare la consistenza delle funzioni di sicurezza. Si può strutturare il piano di valida-

zione individuando due principali modalità di funzionamento: *normale funzionamento* e *funzionamento in caso di guasto*.

4.3.1 Normale funzionamento

Nell'ipotesi di normale funzionamento si deve analizzare il comportamento atteso del sistema dopo l'attivazione di ciascuna funzione di sicurezza. Si possono predisporre dei vettori ingresso da sottoporre al sistema. Quando il numero degli ingressi è limitato, come nel caso in esame, si può utilizzare una tabella di verità per descrivere il comportamento. Successivamente si devono pianificare le prove da eseguire nella configurazione finale.

4.3.1.1 Pianificazione analisi

Si può costruire una tabella in cui si identificano le analisi da condurre, corredate da una descrizione, dalla data di esecuzione, dagli estremi del responsabile e dall'attrezzatura necessaria.

Durante le attività pianificate si deve definire il comportamento del sistema nelle ipotesi di funzionamento indicato. Tale evoluzione andrà confrontata con il reale comportamento del sistema finale in fase di verifica e validazione. Si devono formulare anche i criteri di superamento/fallimento e le modalità di esecuzione del test. Sarà necessario esplicitare le attività di verifica da eseguire.

I vettori di ingresso saranno definiti sulla base delle analisi da `ANF01` a `ANF05`. Inoltre, a partire da queste analisi, sarà possibile sviluppare scenari di guasto da aggiungere a quelli già individuati.

In tabella 4.1 è riportata la pianificazione delle attività di analisi specifica per il caso in esame.

ID	Descrizione	Strumentazione	Data	Responsabile
ANF01	Analisi comportamento con attivazione della -SF10.1.1 in salita e in discesa	Workstation o Laptop	13/08/2024	Riccardo Palmarin
ANF02	Analisi comportamento con attivazione della -SF20.1.1 in salita e in discesa	Workstation o Laptop	13/08/2024	Riccardo Palmarin
ANF03	Analisi comportamento con attivazione della -SF20.2.1 in salita e in discesa	Workstation o Laptop	13/08/2024	Riccardo Palmarin
ANF04	Analisi comportamento con attivazione della -SF30.1.1	Workstation o Laptop	13/08/2024	Riccardo Palmarin
ANF05	Analisi comportamento con attivazione della -SF30.2.1	Workstation o Laptop	13/08/2024	Riccardo Palmarin
ANF06	Analisi comportamento con vettori di ingressi non previsti	Workstation o Laptop	13/08/2024	Riccardo Palmarin

Tabella 4.1: Pianificazione delle analisi in ipotesi di normale funzionamento

4.3.1.2 Pianificazione verifiche

Mediante simulazioni e analisi del codice realizzato, si verifica se il comportamento teorico del sistema corrisponde a quanto identificato in fase di analisi. Sulla base dei requisiti di superamento/fallimento formulati, verrà riportato l'esito delle verifiche. In tabella 4.2 è riportata la pianificazione delle verifiche nell'ipotesi di normale funzionamento.

Dopo aver completato le attività di verifica, è possibile compilare la tabella B3 in SOFTEMA, relativa all'architettura dei moduli utilizzati.

Le verifiche obbligatorie sancite dalla norma EN IEC 61508 verranno pianificate nella sezione dedicata (4.3.3).

4.3.1.3 Pianificazione test

Secondo le modalità definite in fase di analisi si deve convalidare il comportamento del sistema nella configurazione finale verificando che rispetti le condizioni analizzate e verificate. Per convalidare il sistema si devono eseguire i test pianificati nella tabella 4.3.

ID	Descrizione	Strumentazione	Data	Responsabile
VNF01	Verifica dell'analisi ANF01	Workstation e codice	14/08/2024	Riccardo Palmarin
VNF02	Verifica dell'analisi ANF02	Workstation e codice	14/08/2024	Riccardo Palmarin
VNF03	Verifica dell'analisi ANF03	Workstation e codice	14/08/2024	Riccardo Palmarin
VNF04	Verifica dell'analisi ANF04	Workstation e codice	14/08/2024	Riccardo Palmarin
VNF05	Verifica dell'analisi ANF05	Workstation e codice	14/08/2024	Riccardo Palmarin
VNF06	Verifica dell'analisi ANF06	Workstation e codice	14/08/2024	Riccardo Palmarin

Tabella 4.2: Pianificazione delle verifiche in ipotesi di normale funzionamento

ID	Descrizione	Strumentazione	Data	Responsabile
TNF01	Convalida dell'analisi ANF01	Workstation e PNOZmulti configurator	03/09/2024	Riccardo Palmarin
TNF02	Convalida dell'analisi ANF02	Workstation e PNOZmulti configurator	03/09/2024	Riccardo Palmarin
TNF03	Convalida dell'analisi ANF03	Workstation e PNOZmulti configurator	03/09/2024	Riccardo Palmarin
TNF04	Convalida dell'analisi ANF04	Workstation e PNOZmulti configurator	03/09/2024	Riccardo Palmarin
TNF05	Convalida dell'analisi ANF05	Workstation e PNOZmulti configurator	03/09/2024	Riccardo Palmarin
TNF06	Convalida dell'analisi ANF06	Workstation e PNOZmulti configurator	03/09/2024	Riccardo Palmarin

Tabella 4.3: Pianificazione dei test in ipotesi di normale funzionamento

ID	Descrizione	Strumentazione	Data	Responsabile
AGF01	Dispersione verso terra di una delle linee di collegamento al fungo di emergenza	Workstation o Laptop	16/08/2024	Riccardo Palmarin
AGF02	Scollegamento dei conduttori che collegano CPU e fungo di emergenza	Workstation o Laptop	16/08/2024	Riccardo Palmarin
AGF03	Impedimento disattivazione/attivazione teleruttori	Workstation o Laptop	16/08/2024	Riccardo Palmarin
AGF04	Impedimento apertura/chiusura contatti dei finecorsa	Workstation o Laptop	16/08/2024	Riccardo Palmarin
AGF05	Scollegamento dei conduttori che collegano CPU e finecorsa	Workstation o Laptop	16/08/2024	Riccardo Palmarin
AGF06	Scollegamento dei conduttori che collegano CPU e barriere di sicurezza	Workstation o Laptop	16/08/2024	Riccardo Palmarin
AGF07	Guasto delle barriere di sicurezza	Workstation o Laptop	16/08/2024	Riccardo Palmarin
AGF08	Possibili guasti individuati nelle analisi precedenti	Workstation o Laptop	16/08/2024	Riccardo Palmarin

Tabella 4.4: Pianificazione delle analisi in ipotesi di guasto

4.3.2 Funzionamento in caso di guasto

Nelle ipotesi di guasto formulate al paragrafo 2.4.3.3 si deve analizzare il comportamento atteso del sistema dopo l'attivazione di ciascuna funzione di sicurezza. I vettori di ingresso da utilizzare sono gli stessi utilizzati nell'ipotesi di normale funzionamento.

Successivamente, si possono integrare gli scenari di guasto formulati durante le analisi nell'ipotesi di normale funzionamento. Infine, si devono pianificare le prove della configurazione finale.

4.3.2.1 Pianificazione analisi

Come per l'analisi nell'ipotesi di normale funzionamento, è necessario pianificare le analisi considerando i guasti individuati nella lista guasti e nel processo dedicato al normale funzionamento. La pianificazione dettagliata è riportata nella tabella 4.4.

ID	Descrizione	Strumentazione	Data	Responsabile
VGF01	Verifica dell'analisi AGF01	Workstation o Laptop	17/08/2024	Riccardo Palmarin
VGF02	Verifica dell'analisi AGF02	Workstation o Laptop	17/08/2024	Riccardo Palmarin
VGF03	Verifica dell'analisi AGF03	Workstation o Laptop	17/08/2024	Riccardo Palmarin
VGF04	Verifica dell'analisi AGF04	Workstation o Laptop	17/08/2024	Riccardo Palmarin
VGF05	Verifica dell'analisi AGF05	Workstation o Laptop	17/08/2024	Riccardo Palmarin
VGF06	Verifica dell'analisi AGF06	Workstation o Laptop	17/08/2024	Riccardo Palmarin
VGF07	Verifica dell'analisi AGF07	Workstation o Laptop	17/08/2024	Riccardo Palmarin
VGF08	Verifica dell'analisi AGF08	Workstation o Laptop	17/08/2024	Riccardo Palmarin

Tabella 4.5: Pianificazione delle verifiche in ipotesi di guasto

4.3.2.2 Pianificazione verifiche

Mediante le simulazioni e le analisi condotte si deve verificare che il comportamento del sistema corrisponda a quello individuato in fase di analisi. In tabella 4.5 è riportata la pianificazione delle attività di verifica.

4.3.2.3 Pianificazione test

Secondo le modalità definite durante le analisi, si devono condurre le prove per convalidare il comportamento del sistema in configurazione finale. La pianificazione dei test è quella riportata nella tabella 4.6.

4.3.3 Verifiche normate

Infine è necessario pianificare le verifiche sancite dalla norma EN IEC 61508 relative al codice, di cui al paragrafo 2.3.3.3. Di seguito, nella tabella 4.7 è riportata la pianificazione di tali attività di verifica.

4.4 C - Analisi e Verifiche

Nella presente sezione vengono presentate le modalità di analisi e di verifica nei casi specificati al paragrafo 4.3. Le modalità di analisi saranno descritte in dettaglio nella sezione all'analisi specifica.

ID	Descrizione	Strumentazione	Data	Responsabile
TGF01	Convalida dell'analisi AGF01	Workstation e PNOZmulti configurator	03/09/2024	Riccardo Palmarin
TGF02	Convalida dell'analisi AGF02	Workstation e PNOZmulti configurator	03/09/2024	Riccardo Palmarin
TGF03	Convalida dell'analisi AGF03	Workstation e PNOZmulti configurator	03/09/2024	Riccardo Palmarin
TGF04	Convalida dell'analisi AGF04	Workstation e PNOZmulti configurator	03/09/2024	Riccardo Palmarin
TGF05	Convalida dell'analisi AGF05	Workstation e PNOZmulti configurator	03/09/2024	Riccardo Palmarin
TGF06	Convalida dell'analisi AGF06	Workstation e PNOZmulti configurator	03/09/2024	Riccardo Palmarin
TGF07	Convalida dell'analisi AGF07	Workstation e PNOZmulti configurator	03/09/2024	Riccardo Palmarin
TGF08	Convalida dell'analisi AGF08	Workstation e PNOZmulti configurator	03/09/2024	Riccardo Palmarin

Tabella 4.6: Pianificazione dei test in ipotesi di guasto

ID	Descrizione	Strumentazione	Data	Responsabile
VN01	Verifica dei requisti di sicurezza del software	Workstation o Laptop	17/08/2024	Riccardo Palmarin
VN02	Verifica dell'architettura del software	Workstation o Laptop	17/08/2024	Riccardo Palmarin
VN03	Verifica della progettazione del software	Workstation o Laptop	17/08/2024	Riccardo Palmarin
VN04	Verifica del codice, dei dati e delle performance	Workstation o Laptop	17/08/2024	Riccardo Palmarin

Tabella 4.7: Pianificazione delle verifiche normate dalla EN IEC 61508

4.4.1 Normale funzionamento

4.4.1.1 Analisi e verifica -SF10.1.1

Il comportamento statico in risposta all'attivazione del fungo di emergenza `Sb0` può essere descritto mediante una tabella di verità semplificata. Successivamente, a partire da tale risultato è possibile determinare il comportamento dinamico atteso dal sistema. Il sistema deve reagire all'azione sul fungo di emergenza disattivando i segnali di controllo sugli attuatori.

FC1	FC2	FC3	FC4	Sb0_1	Sb0_2	Sb_1	Sb_2	Sb_3	BFT_1_1	BFT_1_2	BFT_2_1	BFT_2_2	K1	K2	K3	K4
Mb0.I4	*Mb0.I5*	*Mb0.I6*	*Mb0.I7*	*Mb0.I8*	*Mb0.I9*	*Mb0.I10*	*Mb0.I11*	*Mb0.I12*	*Mb0.I14*	*Mb0.I13*	*Mb0.I15*	*Mb0.IM16*	*Mb0.O0*	*Mb0.O1*	*Mb0.O2*	*Mb0.O3*
X	X	X	X	0	0	X	X	X	X	X	X	X	0	0	0	0
X	X	X	X	1	1	X	X	X	X	X	X	X	X	X	X	X

Figura 4.9: Analisi ANF01: Tabella di verità relativa alla -SF10.1.1.

Indifferentemente dallo stato degli altri ingressi,l'attivazione della funzione di sicurezza *SF10.1.1* deve inibire ogni movimento. Dal punto di vista dinamico, affinché non venga rilevato un guasto, i due contatti devono modificare il loro stato entro 50ms l'uno dall'altro.

Si può simulare il codice nell'ambiente di sviluppo PNOZmulti Configurator, utilizzando i vettori di ingressi caratteristici durante il normale funzionamento. Per testare esclusivamente la funzione di sicurezza relativa allo stop di emergenza, si assume che le altre funzioni di sicurezza non siano attivate. Si considera superata l'analisi se il comportamento simulato del sistema corrisponde al comportamento rilevato all'analisi.

Il vettore di ingresso V1 riassume i casi di richiesta di salita e discesa con la funzione di sicurezza attiva. I vettori V4 e V5 individuano il comportamento del sistema con funzione di sicurezza disattivata in discesa e in salita. Di seguito si riporta la tabella con il risultato della verifica.

N°	Sb0_1	Sb0_2	Sb1	Sb2	Sb3	K1	K2	K3	K4	Esito
V1	0	0	X	X	X	**0**	**0**	**0**	**0**	OK
V2	1	1	0	0	0	**0**	**0**	**0**	**0**	OK
V3	1	1	0	0	1	**0**	**0**	**0**	**0**	OK
V4	1	1	0	1	0	**0**	**0**	**1**	**1**	OK
V5	1	1	1	0	0	**1**	**1**	**0**	**0**	OK

Tabella 4.8: Risultato della verifica VNF01 della funzione di sicurezza SF10.1.1. In grassetto è riportato lo stato dell'uscita quando è applicato il vettore degli ingressi `Vi`.

La verifica nell'ipotesi di normale funzionamento della funzione di sicurezza ha avuto esito *positivo*.

4.4.1.2 Analisi e verifica -SF20.1.1

Quando la barriera fotoelettrica di sicurezza al piano terra rileva un oggetto, devono essere inibiti tutti i movimenti. Il segnale di entrambi i canali deve commutare entro 50ms

l'uno dall'altro, altrimenti viene rilevato un guasto della funzione di sicurezza. Il comportamento statico del sistema all'attivazione della funzione di sicurezza *-SF20.1.1* è descritto nella tabella seguente tabella di verità.

FC1	FC2	FC3	FC4	Sb0_1	Sb0_2	Sb_1	Sb_2	Sb_3	BFT_1_1	BFT_1_2	BFT_2_1	BFT_2_2	K1	K2	K3	K4
Mb0.I4	*Mb0.I5*	*Mb0.I6*	*Mb0.I7*	*Mb0.I8*	*Mb0.I9*	*Mb0.I10*	*Mb0.I11*	*Mb0.I12*	*Mb0.I14*	*Mb0.I13*	*Mb0.I15*	*Mb0.IM16*	*Mb0.O0*	*Mb0.O1*	*Mb0.O2*	*Mb0.O3*
X	X	X	X	X	X	X	X	X	0	0	X	X	0	0	0	0
X	X	X	X	X	X	X	X	X	1	1	X	X	X	X	X	X

Figura 4.10: Analisi ANF02: Tabella di verità relativa alla -SF20.1.1.

Come nel caso precedente, si può simulare il comportamento del sistema sottoponendo dei vettori in ingresso. Si assume che la *-SF20.1.1* sia l'unica funzione di sicurezza che può attivarsi. I vettori caratteristici del normale funzionamento sono quelli usati nella tabella 4.8.

N°	BFT_1_1	BFT_1_2	Sb1	Sb2	Sb3	K1	K2	K3	K4	Esito
V1	0	0	X	X	X	**0**	**0**	**0**	**0**	OK
V2	1	1	0	0	0	**0**	**0**	**0**	**0**	OK
V3	1	1	0	0	1	**0**	**0**	**0**	**0**	OK
V4	1	1	0	1	0	**0**	**0**	**1**	**1**	OK
V5	1	1	1	0	0	**1**	**1**	**0**	**0**	OK

Tabella 4.9: Risultato della verifica VNF02 della funzione di sicurezza SF20.1.1. In grassetto è riportato lo stato dell'uscita quando è applicato il vettore degli ingressi `Vi`.

La verifica nell'ipotesi di normale funzionamento della funzione di sicurezza ha avuto esito *positivo*.

4.4.1.3 Analisi e verifica -SF20.2.1

Analogamente a quanto analizzato al paragrafo 4.4.1.2, quando la barriera fotoelettrica di sicurezza al primo piano rileva un oggetto, devono essere inibiti tutti i movimenti. In questo caso la tabella di verità che descrive il comportamento del sistema è la seguente.

FC1	FC2	FC3	FC4	Sb0_1	Sb0_2	Sb_1	Sb_2	Sb_3	BFT_1_1	BFT_1_2	BFT_2_1	BFT_2_2	K1	K2	K3	K4
Mb0.I4	*Mb0.I5*	*Mb0.I6*	*Mb0.I7*	*Mb0.I8*	*Mb0.I9*	*Mb0.I10*	*Mb0.I11*	*Mb0.I12*	*Mb0.I14*	*Mb0.I13*	*Mb0.I15*	*Mb0.IM16*	*Mb0.O0*	*Mb0.O1*	*Mb0.O2*	*Mb0.O3*
X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0
X	X	X	X	X	X	X	X	X	X	X	1	1	X	X	X	X

Figura 4.11: Analisi ANF03: Tabella di verità relativa alla -SF20.2.1.

La verifica dell'analisi è eseguita come specificato nel paragrafo dedicato alla *-SF20.1.1*. In tabella 4.10 è riportato il risultato della verifica.

N°	BFT_2_1	BFT_2_2	Sb1	Sb2	Sb3	K1	K2	K3	K4	Esito
V1	0	0	X	X	X	**0**	**0**	**0**	**0**	OK
V2	1	1	0	0	0	**0**	**0**	**0**	**0**	OK
V3	1	1	0	0	1	**0**	**0**	**0**	**0**	OK
V4	1	1	0	1	0	**0**	**0**	**1**	**1**	OK
V5	1	1	1	0	0	**1**	**1**	**0**	**0**	OK

Tabella 4.10: Risultato della verifica VNF03 della funzione di sicurezza SF20.2.1. In grassetto è riportato lo stato dell'uscita quando è applicato il vettore degli ingressi `Vi`.

La verifica nell'ipotesi di normale funzionamento della funzione di sicurezza ha avuto esito *positivo*.

4.4.1.4 Analisi e verifica -SF30.1.1

Quando i finecorsa al piano terra vengono attivati, la discesa dell'ascensore deve essere inibita . Pertanto, i teleruttori K3 e K4 devono essere necessariamente disattivati. La tabella che descrive il comportamento statico quando viene attivata al funzione di sicurezza è la seguente.

FC1	FC2	FC3	FC4	Sb0_1	Sb0_2	Sb_1	Sb_2	Sb_3	BFT_1_1	BFT_1_2	BFT_2_1	BFT_2_2	K1	K2	K3	K4
Mb0.I4	*Mb0.I5*	*Mb0.I6*	*Mb0.I7*	*Mb0.I8*	*Mb0.I9*	*Mb0.I10*	*Mb0.I11*	*Mb0.I12*	*Mb0.I14*	*Mb0.I13*	*Mb0.I15*	*Mb0.IM16*	*Mb0.O0*	*Mb0.O1*	*Mb0.O2*	*Mb0.O3*
X	X	0	0	X	X	X	X	X	X	X	X	X	0	0	X	X
X	X	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X

Figura 4.12: Analisi ANF04: Tabella di verità relativa alla -SF30.1.1.

Anche in questo caso, la commutazione dei due finecorsa deve avvenire entro 50ms. I vettori con cui sollecitare il sistema, in questo caso, sono diversi. E' essenziale riportare chiaramente il comportamento del sistema quando è attivata la funzione di sicurezza.

N°	FC3	FC4	Sb1	Sb2	Sb3	K1	K2	K3	K4	Esito
V1	0	0	0	0	X	**0**	**0**	**0**	**0**	OK
V2	0	0	0	1	0	**0**	**0**	**0**	**0**	OK
V3	0	0	1	0	0	**1**	**1**	**0**	**0**	OK
V4	1	1	0	0	0	**0**	**0**	**0**	**0**	OK
V5	1	1	0	0	1	**0**	**0**	**0**	**0**	OK
V6	1	1	0	1	0	**0**	**0**	**1**	**1**	OK
V7	1	1	1	0	0	**1**	**1**	**0**	**0**	OK

Tabella 4.11: Risultato della verifica VNF04 della funzione di sicurezza SF30.1.1. In grassetto è riportato lo stato dell'uscita quando è applicato il vettore degli ingressi `Vi`.

La verifica nell'ipotesi di normale funzionamento della funzione di sicurezza ha avuto esito *positivo*.

4.4.1.5 *Analisi e verifica -SF30.2.1*

Analogamente a quanto visto al paragrafo 4.4.1.4, quando i finecorsa al primo piano vengono attivati la salita deve essere inibita. Tale comportamento è descritto dalla seguente tabella di verità.

FC1	FC2	FC3	FC4	Sb0_1	Sb0_2	Sb_1	Sb_2	Sb_3	BFT_1_1	BFT_1_2	BFT_2_1	BFT_2_2	K1	K2	K3	K4
Mb0.I4	*Mb0.I5*	*Mb0.I6*	*Mb0.I7*	*Mb0.I8*	*Mb0.I9*	*Mb0.I10*	*Mb0.I11*	*Mb0.I12*	*Mb0.I14*	*Mb0.I13*	*Mb0.I15*	*Mb0.IM16*	*Mb0.O0*	*Mb0.O1*	*Mb0.O2*	*Mb0.O3*
0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0
1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Figura 4.13: Analisi ANF05: Tabella di verità relativa alla -SF30.2.1.

La verifica è condotta in modo analogo a quanto fatto per la *-SF30.2.1*. Nella seguente tabella è riportato il risultato.

N°	FC3	FC4	Sb1	Sb2	Sb3	K1	K2	K3	K4	Esito
V1	0	0	0	0	X	**0**	**0**	**0**	**0**	OK
V2	0	0	0	1	0	**0**	**0**	**1**	**1**	OK
V3	0	0	1	0	0	**0**	**0**	**0**	**0**	OK
V4	1	1	0	0	0	**0**	**0**	**0**	**0**	OK
V5	1	1	0	0	1	**0**	**0**	**0**	**0**	OK
V6	1	1	0	1	0	**0**	**0**	**1**	**1**	OK
V7	1	1	1	0	0	**1**	**1**	**0**	**0**	OK

Tabella 4.12: Risultato della verifica VNF05 della funzione di sicurezza SF30.2.1. In grassetto è riportato lo stato dell'uscita quando è applicato il vettore degli ingressi `Vi`.

La verifica nell'ipotesi di normale funzionamento della funzione di sicurezza ha avuto esito *positivo*.

4.4.1.6 *Analisi e verifica con vettori di ingresso non previsti*

I vettori non previsti nel normale funzionamento sono quelle combinazioni degli ingressi che non hanno un significato logico, ma che potrebbero essere sottoposti al sistema a causa di malfunzionamenti o di un uso improprio ragionevolmente prevedibile.

Ad esempio, potrebbero essere richieste contemporaneamente le operazioni di salita e di discesa. In tal caso, il programma prevede un interblocco tra le uscite con la seguente logica di attivazione:

- Salita: $KM1 = KM2 = Sb_1 \wedge \overline{Sb_2}$
- Discesa: $KM3 = KM4 = \overline{Sb_1} \wedge Sb_2$

In questo scenario, la tabella di verità è la seguente:

FC1	FC2	FC3	FC4	Sb0_1	Sb0_2	Sb_1	Sb_2	Sb_3	BFT_1_1	BFT_1_2	BFT_2_1	BFT_2_2	K1	K2	K3	K4
Mb0.I4	*Mb0.I5*	*Mb0.I6*	*Mb0.I7*	*Mb0.I8*	*Mb0.I9*	*Mb0.I10*	*Mb0.I11*	*Mb0.I12*	*Mb0.I14*	*Mb0.I13*	*Mb0.I15*	*Mb0.IM16*	*Mb0.O0*	*Mb0.O1*	*Mb0.O2*	*Mb0.O3*
X	X	X	X	X	X	1	1	X	X	X	X	X	0	0	0	0

Figura 4.14: Analisi ANF06: Tabella di verità con richiesta simultanea di salita e discesa.

Pertanto, quando entrambi gli ingressi sono attivati, i teleruttori sono tutti disalimentati. Mediante una simulazione, si verifica che il sistema si comporti come previsto. In tabella sono riportati gli esiti

SF Attiva	Sb1	Sb2	KM1	KM2	KM3	KM4	Esito
Nessuna	1	1	**0**	**0**	**0**	**0**	OK
-SF10.1.1	1	1	**0**	**0**	**0**	**0**	OK
-SF20.1.1	1	1	**0**	**0**	**0**	**0**	OK
-SF20.2.1	1	1	**0**	**0**	**0**	**0**	OK
-SF30.1.1	1	1	**0**	**0**	**0**	**0**	OK
-SF30.2.1	1	1	**0**	**0**	**0**	**0**	OK

Tabella 4.13: Risultato della verifica VNF06 quando è richiesta la salita contemporaneamente alla discesa. In grassetto è riportato lo stato dell'uscita.

Più rilevante è il caso in cui l'ingresso di riconoscimento delle funzioni di sicurezza Sb_3 venga mantenuto allo stato logico 1 mentre la funzione di sicurezza è attiva. In questo scenario, il rischio è che la funzione di sicurezza venga inibita durante la pressione del pulsante. Il sistema può disattivare una funzione di sicurezza solo quando gli ingressi specifici associati a tale funzione si trovano nello stato di sicurezza. Il comportamento atteso delle funzioni di sicurezza non varia rispetto a quanto analizzato nei paragrafi precedenti. Nella seguente tabella si riporta il comportamento atteso:

SF Attiva	Sb3	KM1	KM2	KM3	KM4
-SF10.1.1	1	0	0	0	0
-SF20.1.1	1	0	0	0	0
-SF20.2.1	1	0	0	0	0
-SF30.1.1	1	0	0	X	X
-SF30.2.1	1	X	X	0	0

Tabella 4.14: Analisi ANF06: Comportamento atteso in caso di tentativo di bypass delle funzioni di sicurezza.

La verifica mediante simulazione fornisce i risultati riportati nella tabella 4.15.

SF Attiva	Sb3	KM1	KM2	KM3	KM4	Esito
-SF10.1.1	1	**0**	**0**	**0**	**0**	OK
-SF20.1.1	1	**0**	**0**	**0**	**0**	OK
-SF20.2.1	1	**0**	**0**	**0**	**0**	OK
-SF30.1.1	1	**0**	**0**	**0**	**0**	OK
-SF30.2.1	1	**0**	**0**	**0**	**0**	OK

Tabella 4.15: Risultato della verifica VNF06 quando si prova ad inibire l'azione delle funzioni di sicurezza. In grassetto è riportato lo stato dell'uscita.

Altre combinazioni di ingressi rappresentano già le combinazioni logiche delle due casistiche precedentemente analizzate. Pertanto, non è necessario esaminarle ulteriormente o verificarle.

4.4.2 Funzionamento in caso di guasto

4.4.2.1 Analisi e verifica guasto fungo di emergenza

In riferimento all'analisi AGF01, la funzione di sicurezza deve mantenere la sua operatività anche nel caso in cui una delle linee di collegamento al fungo di emergenza provochi una dispersione verso terra. In tale scenario, il comportamento atteso è quello descritto dalla tabella seguente.

FC1	FC2	FC3	FC4	Sb0_1	Sb0_2	Sb_1	Sb_2	Sb_3	BFT_1_1	BFT_1_2	BFT_2_1	BFT_2_2	K1	K2	K3	K4
Mb0.I4	*Mb0.I5*	*Mb0.I6*	*Mb0.I7*	*Mb0.I8*	*Mb0.I9*	*Mb0.I10*	*Mb0.I11*	*Mb0.I12*	*Mb0.I14*	*Mb0.I13*	*Mb0.I15*	*Mb0.IM16*	*Mb0.O0*	*Mb0.O1*	*Mb0.O2*	*Mb0.O3*
X	X	X	X	1	0	X	X	X	X	X	X	X	0	0	0	0
X	X	X	X	0	1	X	X	X	X	X	X	X	0	0	0	0

Figura 4.15: Tabella di verità in caso di guasto dei contatti del fungo di emergenza. Analisi AGF01

I moduli software prodotti da Pilz per la gestione dell'arresto di emergenza prevedono un controllo su due canali. Se vengono rilevati due stati differenti, il modulo attiva la funzione di sicurezza segnalando il guasto del componente. Mediante una simulazione si verifica che il software si comporti come previsto:

Sb_0_1	Sb_0_2	KM1	KM2	KM3	KM4	Esito
0	1	**0**	**0**	**0**	**0**	OK
1	0	**0**	**0**	**0**	**0**	OK

Tabella 4.16: Risultato della verifica VGF01. In grassetto è riportato lo stato dell'uscita rilevato.

Per quanto riguarda l'analisi AGF02, se i conduttori che connettono la CPU al fungo di emergenza vengono interrotti, viene attivata la funzione di sicurezza grazie all'utilizzo della logica negata. Se vengono disconnessi entrambi i cavi, si rientra nello scenario

analizzato nella ANF01. Per tali motivi non è necessario studiare ulteriormente queste casistiche.

4.4.2.2 Analisi e verifica guasto teleruttori

Lo studio del guasto dei teleruttori richiede considerazioni prevalentemente a livello hardware.Per un approfondimento su questo tema, si rimanda alla tesi di Giacomo Comunian.

4.4.2.3 Analisi e verifica guasto finecorsa

In riferimento alle analisi AGF04 e AGF05, le funzioni di sicurezza `-SF30.1.1` e `-SF30.2.1` richiedono che le coppie di finecorsa FC3, FC4 e FC1, FC2 trasmettano lo stesso stato alla CPU. Nel caso in cui uno dei finecorsa non commutasse il proprio stato, il comportamento del sistema sarà descritto dalle seguenti tabelle di verità:

FC1	FC2	FC3	FC4	Sb0_1	Sb0_2	Sb_1	Sb_2	Sb_3	BFT_1_1	BFT_1_2	BFT_2_1	BFT_2_2	K1	K2	K3	K4
Mb0.I4	*Mb0.I5*	*Mb0.I6*	*Mb0.I7*	*Mb0.I8*	*Mb0.I9*	*Mb0.I10*	*Mb0.I11*	*Mb0.I12*	*Mb0.I14*	*Mb0.I13*	*Mb0.I15*	*Mb0.IM16*	*Mb0.O0*	*Mb0.O1*	*Mb0.O2*	*Mb0.O3*
X	X	1	0	X	X	X	X	X	X	X	X	X	0	0	X	X
X	X	0	1	X	X	X	X	X	X	X	X	X	0	0	X	X

Figura 4.16: Analisi AGF04: Tabella di verità in caso di guasto dei finecorsa FC3 e FC4.

FC1	FC2	FC3	FC4	Sb0_1	Sb0_2	Sb_1	Sb_2	Sb_3	BFT_1_1	BFT_1_2	BFT_2_1	BFT_2_2	K1	K2	K3	K4
Mb0.I4	*Mb0.I5*	*Mb0.I6*	*Mb0.I7*	*Mb0.I8*	*Mb0.I9*	*Mb0.I10*	*Mb0.I11*	*Mb0.I12*	*Mb0.I14*	*Mb0.I13*	*Mb0.I15*	*Mb0.IM16*	*Mb0.O0*	*Mb0.O1*	*Mb0.O2*	*Mb0.O3*
1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0
0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0

Figura 4.17: Analisi AGF05: Tabella di verità in caso di guasto dei finecorsa FC1 e FC1.

Il programma utilizza un controllo a doppio canale sullo stato dei finecorsa. E' sufficiente che uno dei due commuti per attivare la funzione di sicurezza. Nella seguente tabella è riportato il risultato della verifica tramite simulazione:

FC3	FC4	KM3	KM4	Esito
0	1	**0**	**0**	OK
1	0	**0**	**0**	OK

Tabella 4.17: Risultato della verifica AGF04 con guasto dei finecorsa FC3 e FC4. In grassetto è riportato lo stato dell'uscita rilevato.

FC1	FC2	KM1	KM2	Esito
0	1	**0**	**0**	OK
1	0	**0**	**0**	OK

Tabella 4.18: Risultato della verifica AGF05 con guasto dei finecorsa FC1 e FC2. In grassetto è riportato lo stato dell'uscita rilevato.

Il guasto dei conduttori che collegano i finecorsa al PLC non provocano la perdita della funzione di sicurezza grazie all'utilizzo della logica negata e del controllo su due canali. La situazione a seguito del guasto richiede lo stesso comportamento analizzato per l'analisi AGF04. Pertanto, non è necessario esaminare ulteriormente questa casistica.

4.4.2.4 Analisi e verifica guasto barriere di sicurezza

In riferimento alle analisi AGF06 e AGF07, è necessario analizzare il comportamento del sistema in caso di guasto delle barriere fotoelettriche di sicurezza. Le barriere fotoelettriche sono dotate di una doppia uscita con logica NC. Qualora una delle uscite dovesse essere allo stato logico basso, la funzione di sicurezza deve intervenire. Nelle seguenti tabelle di verità si riporta il comportamento atteso del sistema.

FC1	FC2	FC3	FC4	Sb0_1	Sb0_2	Sb_1	Sb_2	Sb_3	BFT_1_1	BFT_1_2	BFT_2_1	BFT_2_2	K1	K2	K3	K4
Mb0.I4	*Mb0.I5*	*Mb0.I6*	*Mb0.I7*	*Mb0.I8*	*Mb0.I9*	*Mb0.I10*	*Mb0.I11*	*Mb0.I12*	*Mb0.I14*	*Mb0.I13*	*Mb0.I15*	*Mb0.IM16*	*Mb0.O0*	*Mb0.O1*	*Mb0.O2*	*Mb0.O3*
X	X	X	X	X	X	X	X	X	1	0	X	X	0	0	0	0
X	X	X	X	X	X	X	X	X	0	1	X	X	0	0	0	0

Figura 4.18: Analisi AGF06: Tabella di verità in caso di guasto della barriera di sicurezza BFT1.

FC1	FC2	FC3	FC4	Sb0_1	Sb0_2	Sb_1	Sb_2	Sb_3	BFT_1_1	BFT_1_2	BFT_2_1	BFT_2_2	K1	K2	K3	K4
Mb0.I4	*Mb0.I5*	*Mb0.I6*	*Mb0.I7*	*Mb0.I8*	*Mb0.I9*	*Mb0.I10*	*Mb0.I11*	*Mb0.I12*	*Mb0.I14*	*Mb0.I13*	*Mb0.I15*	*Mb0.IM16*	*Mb0.O0*	*Mb0.O1*	*Mb0.O2*	*Mb0.O3*
X	X	X	X	X	X	X	X	X	X	X	1	0	0	0	0	0
X	X	X	X	X	X	X	X	X	X	X	0	1	0	0	0	0

Figura 4.19: Analisi AGF06: Tabella di verità in caso di guasto della barriera di sicurezza BFT2.

Mediante simulazione, la verifica restituisce il seguente risultato:

BFT_1_1	BFT_1_2	KM1	KM2	KM3	KM4	Esito
0	1	**0**	**0**	**0**	**0**	OK
1	0	**0**	**0**	**0**	**0**	OK

Tabella 4.19: Risultato della verifica AGF06 con guasto della barriera BFT1. In grassetto è riportato lo stato dell'uscita rilevato.

BFT_2_1	BFT_2_2	KM1	KM2	KM3	KM4	Esito
0	1	**0**	**0**	**0**	**0**	OK
1	0	**0**	**0**	**0**	**0**	OK

Tabella 4.20: Risultato della verifica AGF06 con guasto della barriera BFT2. In grassetto è riportato lo stato dell'uscita rilevato.

Come nelle analisi precedenti, la rottura dei conduttori di collegamento, non comporta la perdita della funzione di sicurezza. Pertanto, non è necessario studiare tale scenario.

4.4.2.5 Analisi e verifica guasti software

Nel corso delle analisi precedenti non sono stati individuati ulteriori scenari di guasto da analizzare.

4.4.3 Verifiche normate

4.4.3.1 Verifica dell'architettura software

L'architettura del software rispetta il paradigma a tre stadi: ogni ingresso interagisce su un blocco funzionale dedicato alla gestione della funzione di sicurezza. Il blocco elabora l'ingresso e aggiorna una variabile che viene usata alla fine del codice per controllare l'uscita. Pertanto, tale verifica è superata.

4.4.3.2 Verifica della progettazione del software

Il software implementa correttamente i requisiti di sicurezza specificati, garantendo il livello di integrità della sicurezza specificato. Le verifiche e le analisi precedenti hanno confermato la solidità della programmazione, senza rilevare situazioni pericolose e garantendo il rispetto del determinismo. La verifica è quindi superata.

4.4.3.3 Verifica del codice e dei dati

Le variabili sono state dichiarate con un nome autoesplicativo, con riferimento allo schema elettrico e con un commento. Ciascuna rete è dotata di un commento nella parte dichiarativa.

Le uscite vengono controllate una sola volta, alla fine del codice. I pulsanti di emergenza, le barriere fotoelettriche di sicurezza e i finecorsa agiscono sui blocchi funzionali dedicati forniti e validati da Pilz. Il controllo a due canali rileva eventuali discrepanze superiori a 50ms.

Il codice è nel complesso leggibile, comprensibile, chiaro e corretto. Non è necessario eseguire particolari verifiche sui dati poiché sono utilizzati solo tipi booleani.

La verifica è superata con successo.

4.4.3.4 Verifica delle performance temporali

La verifica del determinismo temporale del controllo a due canali non è necessaria, poiché è già garantita dalla certificazione del PLC. Durante le attività di simulazione, non sono emersi ritardi o stalli nella commutazione delle uscite all'attivazione delle funzioni di sicurezza. La programmazione garantisce che non si verifichino loop infiniti o cicli di stallo. Il codice ha la complessità e lunghezza minima, permettendo di massimizzare la velocità di esecuzione.

4.4.3.5 Esito delle verifiche normate

Nella seguente tabella sono riportati gli esiti delle verifiche normate effettuate.

ID	Descrizione	Data	Responsabile	Esito
VN01	Verifica dei requisiti di sicurezza del software	17/08/2024	Riccardo Palmarin	OK
VN02	Verifica dell'architettura del software	17/08/2024	Riccardo Palmarin	OK
VN03	Verifica della progettazione del software	17/08/2024	Riccardo Palmarin	OK
VN04	Verifica del codice, dei dati e delle performance	17/08/2024	Riccardo Palmarin	OK

Tabella 4.21: Esito delle verifiche normate

4.4.4 Aggiornamento tabelle SOFTEMA

A seguito delle analisi e delle verifiche, si possono aggiornare le tabelle in SOFTEMA in modo da tracciare il processo di verifica.

4.4.4.1 Aggiornamento tabella A3

In riferimento alle verifiche normate, si può aggiornare la colonna `Verifikation` segnando l'esito delle verifiche condotte precedentemente. I requisiti da `R7` a `R18` sono rispettati.

La tabella costituirà una parte integrante della documentazione e, quando completa, è adatta per l'archiviazione. Pertanto, si possono verificare i requisiti `R25` e `R26`.

L'utilizzo di un PLC safety con programmazione in LVL permette un controllo attivo sul tempo di ciclo di esecuzione. Superata la soglia impostata, il PLC entra in uno stato di allarme sicuro. Le funzioni di sicurezza operano con logica NC e il programma non può essere modificato durante il funzionamento. La modifica del programma può essere implementata solo offline dal team di progettazione. Il documento riportato in SOFTEMA è a disposizione di tutto il team che ha implementato il ciclo di vita del software. Per tali motivi, si possono verificare i requisiti da `R27` a `R31`.

Si possono verificare anche i requisiti inerenti all'ambiente di programmazione e all'utilizzo dei blocchi funzione sulla base delle analisi e delle verifiche precedenti.

Alla fine della tabella si può vedere l'esito della verifica alla voce `Summe`. E' importante impostare la data e il nome di chi ha eseguito la verifica. Al termine di questa fase l'esito della verifica riportato deve essere *OK*, come riportato in figura 4.20.

Summe	OK
Datum	19/08/2024
Name	Riccardo Palmarin
Signatur	Riccardo Palmarin

Figura 4.20: Tabella A3: Esito della verifica delle misure adottate.

4.4.4.2 Aggiornamento tabella B3

E' necessario documentare i blocchi funzionali utilizzati nella programmazione. Nel caso in esame sono stati usati 3 blocchi funzionali: uno per la gestione dell'arresto di emergenza, uno per la gestione delle barriere di sicurezza e uno per la gestione dei finecorsa.

Mediante il `Modul-Manager` è possibile creare e gestire i blocchi funzionali utilizzati nel programma indicando gli ingressi, le uscite e i parametri. E' importante indicare anche le informazioni relative alla versione e al produttore del blocco. Conclusa la definizione dei blocchi funzionali si può aggiornare la tabella. Si devono inserire cinque righe nella sezione ingressi.

Nella sezione ingressi si devono inserire un SF_ESTOP, due SF_BFT e due SF_FC necessari per l'implementazione delle cinque funzioni di sicurezza. Si consiglia di inserire una descrizione del blocco funzionale. E' obbligatorio definire un nome di istanza. Nella colonna `Eingang` si devono specificare gli ingressi della tabella *A2.4* che agiscono sul blocco funzionale. Nella colonna `Ausgang` si deve inserire il nome dell'uscita del blocco. Conclusa la compilazione, è importante bloccare la modifica agendo sulla colonna `Sperre`. In figura 4.21 è riportata la corretta compilazione della tabella.

_Nr	_FB-Name	_Instanzname	_Eingäng	_Ausgang	_Beschreibung	_Parameter	_Hersteller	_Version/Signatur	_Aktiv	_Sperre
	Eingangsmodule									
IM1	SF_ESTOP	ESTOP_1	Sb0_1 Sb0_2	EMST_OK	Arresto di emergenza		Pilz	1.0.1/10.8.2024	Aktiv	x
IM2	SF_BFT	SF_BFT_1	BFT_1_1 BFT_1_2	BFT_OK_	Barriera fotoelettrica		Pilz	1.0.1/10.08.2024	Aktiv	x
IM3	SF_BFT	SF_BFT_2	BFT_2_1 BFT_2_2	BFT_OK_	Barriera fotoelettrica		Pilz	1.0.1/10.08.2024	Aktiv	x
IM4	SF_FC	SF_FC_1	FC3 FC4	FC_OK_	Finecorsa		Pilz	1.0.1/10.08.2024	Aktiv	x
IM5	SF_FC	SF_FC_2	FC1 FC2	FC_OK_	Finecorsa		Pilz	1.0.1/10.08.2024	Aktiv	x

Figura 4.21: Compilazione della tabella B3 per documentare i moduli utilizzati.

I moduli utilizzati sono stati sviluppati, verificati e certificati da Pilz. Si può agire sulla colonna `Verifikation` per riportare l'esito positivo della verifica. E' importante riportare la data di quando è stata aggiornata la tabella e gli estremi di chi ha eseguito la modifica.

L'esito globale della verifica conclusa la compilazione deve essere `OK`.

4.4.4.3 *Aggiornamento tabella A2.4*

Dopo la compilazione della tabella *B3* si può aggiornare la colonna `Modul`. Si devono associare gli ingressi ai blocchi funzionali su cui agiscono. Ad esempio, gli ingressi `Sb0_1` e `Sb0_2` agiscono sul modulo `ESTOP_1`. Dopo aver ultimato la compilazione si deve agire sulla colonna `Sperre` per impedire la modifica. In figura 4.22 è riportata la corretta compilazione della tabella.

Nr	Beschreibung	Symbol	Adresse	Datentyp	Modul	Aktiv in C+E	Aktiv	Sperre
	Eingänge							
I1	Finecorsa salita n1 (NC)	FC1	Mb0.I4	SF_BOOL	SF_FC_2	Aktiv	Aktiv	x
I2	Finecorsa salita n2 (NC)	FC2	Mb0.I5	SF_BOOL	SF_FC_2	Aktiv	Aktiv	x
I3	Finecorsa discesa n1 (NC)	FC3	Mb0.I6	SF_BOOL	SF_FC_1	Aktiv	Aktiv	x
I4	Finecorsa discesa n2 (NC)	FC4	Mb0.I7	SF_BOOL	SF_FC_1	Aktiv	Aktiv	x
I5	Pulsante di emergenza contatto n1 (NC)	Sb0_1	Mb0.I8	SF_BOOL	ESTOP_1	Aktiv	Aktiv	x
I6	Pulsante di emergenza contatto n2 (NC)	Sb0_2	Mb0.I9	SF_BOOL	ESTOP_1	Aktiv	Aktiv	x
I7	Pulsante salita (NO)	Sb_1	Mb0.I10	SF_BOOL		Aktiv	Aktiv	x
I8	Pulsante discesa (NO)	Sb_2	Mb0.I11	SF_BOOL		Aktiv	Aktiv	x
I9	Pulsante riarmo (NO)	Sb_3	Mb0.I12	SF_BOOL		Aktiv	Aktiv	x
I10	Barriera di sicurezza PT contatto n2 (NC)	BFT_1_2	Mb0.I13	SF_BOOL	SF_BFT_1	Aktiv	Aktiv	x
I11	Barriera di sicurezza PT contatto n1 (NC)	BFT_1_1	Mb0.I14	SF_BOOL	SF_BFT_1	Aktiv	Aktiv	x
I12	Barriera di sicurezza P1 contatto n1 (NC)	BFT_2_1	Mb0.I15	SF_BOOL	SF_BFT_2	Aktiv	Aktiv	x
I13	Barriera di sicurezza P1 contatto n2 (NC)	BFT_2_2	Mb0.IM16	SF_BOOL	SF_BFT_2	Aktiv	Aktiv	x
	Ausgänge							
O1	Teleruttore salita n1	KM1	Mb0.O0	BOOL		Aktiv	Aktiv	x
O2	Teleruttore salita n2	KM2	Mb0.O1	BOOL		Aktiv	Aktiv	x
O3	Teleruttore discesa n1	KM3	Mb0.O2	BOOL		Aktiv	Aktiv	x
O4	Teleruttore discesa n2	KM4	Mb0.O3	BOOL		Aktiv	Aktiv	x

Figura 4.22: Compilazione della tabella A2.4.

Nel corso delle verifiche sono state verificate le funzionalità degli ingressi e delle uscite. Ogni I/O è conforme alla funzionalità prevista, pertanto si può aggiornare la colonna `SW-Verif`. Si deve dichiarare chi ha eseguito la verifica e la data dell'esecuzione. Conclusa questa compilazione, l'esito della verifica dovrebbe essere *OK*.

4.4.4.4 *Aggiornamento tabella B4 Matrix C+E*

La compilazione della tabella *B4 Matrix C+E* è di fondamentale importanza ed è particolarmente complicata. Come prima cosa, si deve agire sul controllo `Tabelle aktualisieren` per generare la tabella di partenza. Ogni riga rappresenta una funzione di sicurezza. Si può agire sullo stato degli ingressi per indicare la condizione di attivazione della funzione di sicurezza nell'ipotesi di normale funzionamento. In figura 4.23 è riportata la compilazione corretta sulla base delle analisi eseguite alla sezione 4.4.1.

Nr	Betriebsart	Test	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12	I13	SF-Nr	SFK	Prio	SF-Name
			FC1 [SF_FC_2	FC2 [SF_FC_2	FC3 [SF_FC_1	FC4 [SF_FC_1	Sb0_1 [ESTOP_1]	Sb0_2 [ESTOP_1]	Sb_1 [Mb0.I10]	Sb_2 [Mb0.I11]	Sb_3 [Mb0.I12]	BFT_1_2 [SF_BFT_1	BFT_1_1 [SF_BFT_1	BFT_2_1 [SF_BFT_2	BFT_2_2 [SF_BFT_2				
C0			1	1	1	1	1	1	0	0	0	1	1	1	1				ALLOK
C1	B0: Alle	C0	1	1	1	1	0	0	1	1	1	1	1	1	1	SF1	-SF10.0.1	1	Wenn Fungo di emergenza Sb_0, dann Motor M1 abschalten, mit Sb_3 ACK quittieren.
C2	B1: Automatik	C0	1	1	1	1	1	1	1	1	1	0	0	1	1	SF2	-SF20.1.1	2	Wenn Barriera fotoelettrica piano terra BFT1, dann Motor M1
C3	B1: Automatik	C0	1	1	1	1	1	1	1	1	1	1	1	0	0	SF3	-SF20.2.1	2	Wenn Barriera fotoelettrica primo piano BFT2, dann Motor M1
C4	B1: Automatik	C0	1	1	0	0	1	1	1	0	0	1	1	1	1	SF4	-SF30.1.1	2	Wenn Finecorsa piano terra FC1, dann Motor M1 abschalten, mit Sb_3
C5	B1: Automatik	C0	0	0	1	1	1	1	0	1	0	1	1	1	1	SF5	-SF30.2.1	2	Wenn Finecorsa primo piano FC2, dann Motor M1 abschalten, mit Sb_3

Figura 4.23: Compilazione ingressi della tabella B4 C+E.

Si devono inserire cinque righe che rappresentano le condizioni di guasto analizzate precedentemente. Poiché non è possibile inserire più di un vettore per riga, è necessario aggiungere un commento che spieghi la causa del guasto. Nella figura 4.24 è riportata la compilazione delle righe che rappresentano le condizioni di guasto.

Successivamente si deve modificare un'impostazione della tabella tramite il controllo in alto a destra. Si devono attivare tutte e cinque le spunte in modo da poter compilare le celle logiche delle uscite.

C6	B0: Alle	C0	1	1	1	1	1	0	0	0	0	1	1	1	1	TF_ESTOP		1	Guasto contatto fungo di emergenza
C7	B0: Alle	C0	1	1	1	1	1	1	0	0	0	0	1	1	1	TF_BFT_1		2	Guasto Barriera di sicurezza piano terra
C8	B0: Alle	C0	1	1	1	1	1	1	0	0	1	1	1	0	1	TF_BFT_2		2	Guasto Barriera di sicurezza primo piano
C9	B1: Automatik	C0	1	1	0	1	1	1	1	0	0	1	1	1	1	TF_FC_1		2	Guasto finecorsa piano terra
C10	B1: Automatik	C0	1	0	1	1	1	1	0	1	0	1	1	1	1	TF_FC_2		2	Guasto finecorsa primo piano

Figura 4.24: Compilazione casi di guasto della tabella B4 C+E.

SF-Name	O1	O2	O3	O4	Sperre
	KM1 [Mb0.O0]	KM2 [Mb0.O1]	KM3 [Mb0.O2]	KM4 [Mb0.O3]	
ALLOK	ON (*I7*) Sb_1 AND (*IM1*)	ON (*I7*) Sb_1 AND (*IM1*)	ON (*I8*) Sb_2 AND (*IM1*)	ON (*I7*) Sb_1 AND (*IM1*)	o
Wenn Fungo di emergenza Sb_0, dann Motor M1 abschalten, mit Sb_3 ACK quittieren.	OFF not (*IM1*) ESTOP_1.EMST_OK	OFF not (*IM1*) ESTOP_1.EMST_OK	OFF not (*IM1*) ESTOP_1.EMST_OK	OFF not (*IM1*) ESTOP_1.EMST_OK	o
Wenn Barriera fotoelettrica piano terra BFT1, dann Motor M1	OFF not (*IM2*) SF_BFT_1.BFT_OK_	OFF not (*IM2*) SF_BFT_1.BFT_OK_	OFF not (*IM3*) SF_BFT_2.BFT_OK_	OFF not (*IM3*) SF_BFT_2.BFT_OK_	o
Wenn Barriera fotoelettrica primo piano BFT2, dann Motor M1	OFF not (*IM3*) SF_BFT_2.BFT_OK_	OFF not (*IM3*) SF_BFT_2.BFT_OK_	OFF not (*IM3*) SF_BFT_2.BFT_OK_	OFF not (*IM3*) SF_BFT_2.BFT_OK_	o
Wenn Finecorsa piano terra FC1, dann Motor M1 abschalten, mit Sb_3	NOP	NOP	OFF not (*IM4*) SF_FC_1.FC_OK_	OFF not (*IM4*) SF_FC_1.FC_OK_	o
Wenn Finecorsa primo piano FC2, dann Motor M1 abschalten, mit Sb_3	OFF not (*IM5*) SF_FC_2.FC_OK_	OFF not (*IM5*) SF_FC_2.FC_OK_	NOP	NOP	o

Figura 4.25: Compilazione uscite della tabella B4 C+E.

Con un doppio click su ciascuna cella si può selezione lo stato previsto per la condizione descritta dalla riga. Il Logikeditor consente di inserire le condizioni logiche per raggiungere lo stato specificato. In figura 4.25 sono illustrati gli stati previsti all'attivazione di ciascuna funzione di sicurezza. Questi sono stati ottenuti direttamente dalle tabelle di verità presentate durante le analisi.

In figura 4.26 sono riportate le uscite previste in caso di guasto, sulla base della analisi precedentemente condotte.

SF-Name	O1	O2	O3	O4	Sperre
	KM1 [Mb0.O0]	KM2 [Mb0.O1]	KM3 [Mb0.O2]	KM4 [Mb0.O3]	
Guasto contatto fungo di emergenza	OFF ((*I5*) Sb0_1 AND	OFF ((*I5*) Sb0_1 AND	OFF ((*I5*) Sb0_1 AND	OFF ((*I5*) Sb0_1 AND	o
Guasto Barriera di sicurezza piano terra	OFF ((*I10*) BFT_1_2 AND	OFF ((*I10*) BFT_1_2 AND	OFF ((*I10*) BFT_1_2 AND	OFF ((*I10*) BFT_1_2 AND	o
Guasto Barriera di sicurezza primo piano	OFF ((*I12*) BFT_2_1 AND	OFF ((*I12*) BFT_2_1 AND	OFF ((*I12*) BFT_2_1 AND	OFF ((*I12*) BFT_2_1 AND	o
Guasto finecorsa piano terra	NOP	NOP	OFF	OFF	o
Guasto finecorsa primo piano	OFF	OFF	NOP	NOP	o

Figura 4.26: Compilazione uscite in caso di guasto della tabella B4 C+E.

Conclusa la compilazione, si deve agire sulla colonna `Sperre` per impedire ulteriori modifiche. Infine, si riportano gli esiti delle verifiche condotte, che in questo caso sono stati positivi. E' obbligatorio riportare la data della modifica e gli estremi di chi ha condotto la verifica.

4.4.4.5 Aggiornamento tabella B4 Matrix Kompakt

La matrice compatta riassume tutte le informazioni salienti della tabella *B4 C+E*. Sono riportate le condizioni logiche di attivazione di ciascuna uscita in base alla funzione di sicurezza o al caso di test.

Per creare la tabella in modo automatico è sufficiente cliccare sul pulsante `Tabelle aktualisieren`. SOFTEMA, sulla base delle informazioni inserite nella tabella *B4 C+E*, gestisce in autonomia la creazione delle celle. La modifica manuale è sconsigliata. In figura 4.27 è riportata la tabella compilata. Se durante l'attività di verifica non sono state rilevate criticità di funzionamento, è possibile procedere alla verifica delle righe della tabella.

Le condizioni logiche riportate nella tabella, sono state ricavate dalle tabelle di verità presentate durante le analisi.

Nr	Beschreibung	B0: Alle	B1: Automatik	SF (Prio)
E1	Teleruttore salita n1	OFF: not (*IM1*) ESTOP_1.EMST_OK OFF: ((*I5*) Sb0_1 AND not ((*I6*) Sb0_2)) OR ((*I6*) Sb0_2 AND not ((*I5*) Sb0_1)) OFF: ((*I10*) BFT_1_2 AND not ((*I11*) BFT_1_1)) OR ((*I11*) BFT_1_1 AND not ((*I10*) BFT_1_2)) OFF: ((*I12*) BFT_2_1 AND not ((*I13*) BFT_2_2)) OR ((*I13*) BFT_2_2 AND not ((*I12*) BFT_2_1))	OFF: not (*IM2*) SF_BFT_1.BFT_OK_ OFF: not (*IM3*) SF_BFT_2.BFT_OK_ OFF: not (*IM5*) SF_FC_2.FC_OK_	B0:SF1 (1),TF_ESTOP (1),TF_BFT_1 (2),TF_BFT_2 (2), B1:SF2 (2),SF3 (2),SF5 (2),
E2	Teleruttore salita n2	OFF: not (*IM1*) ESTOP_1.EMST_OK OFF: ((*I5*) Sb0_1 AND not ((*I6*) Sb0_2)) OR ((*I6*) Sb0_2 AND not ((*I5*) Sb0_1)) OFF: ((*I10*) BFT_1_2 AND not ((*I11*) BFT_1_1)) OR ((*I11*) BFT_1_1 AND not ((*I10*) BFT_1_2)) OFF: ((*I12*) BFT_2_1 AND not ((*I13*) BFT_2_2)) OR ((*I13*) BFT_2_2 AND not ((*I12*) BFT_2_1))	OFF: not (*IM2*) SF_BFT_1.BFT_OK_ OFF: not (*IM3*) SF_BFT_2.BFT_OK_ OFF: not (*IM5*) SF_FC_2.FC_OK_	B0:SF1 (1),TF_ESTOP (1),TF_BFT_1 (2),TF_BFT_2 (2), B1:SF2 (2),SF3 (2),SF5 (2),
E3	Teleruttore discesa n1	OFF: not (*IM1*) ESTOP_1.EMST_OK OFF: ((*I5*) Sb0_1 AND not ((*I6*) Sb0_2)) OR ((*I6*) Sb0_2 AND not ((*I5*) Sb0_1)) OFF: ((*I10*) BFT_1_2 AND not ((*I11*) BFT_1_1)) OR ((*I11*) BFT_1_1 AND not ((*I10*) BFT_1_2)) OFF: ((*I12*) BFT_2_1 AND not ((*I13*) BFT_2_2)) OR ((*I13*) BFT_2_2 AND not ((*I12*) BFT_2_1))	OFF: not (*IM3*) SF_BFT_2.BFT_OK_ OFF: not (*IM3*) SF_BFT_2.BFT_OK_ OFF: not (*IM4*) SF_FC_1.FC_OK_	B0:SF1 (1),TF_ESTOP (1),TF_BFT_1 (2),TF_BFT_2 (2), B1:SF2 (2),SF3 (2),SF4 (2),
E4	Teleruttore discesa n2	OFF: not (*IM1*) ESTOP_1.EMST_OK OFF: ((*I5*) Sb0_1 AND not ((*I6*) Sb0_2)) OR ((*I6*) Sb0_2 AND not ((*I5*) Sb0_1)) OFF: ((*I10*) BFT_1_2 AND not ((*I11*) BFT_1_1)) OR ((*I11*) BFT_1_1 AND not ((*I10*) BFT_1_2)) OFF: ((*I12*) BFT_2_1 AND not ((*I13*) BFT_2_2)) OR ((*I13*) BFT_2_2 AND not ((*I12*) BFT_2_1))	OFF: not (*IM3*) SF_BFT_2.BFT_OK_ OFF: not (*IM3*) SF_BFT_2.BFT_OK_ OFF: not (*IM4*) SF_FC_1.FC_OK_	B0:SF1 (1),TF_ESTOP (1),TF_BFT_1 (2),TF_BFT_2 (2), B1:SF2 (2),SF3 (2),SF4 (2),

Figura 4.27: Auto-compilazione tabella B4 compatta.

4.4.4.6 Aggiornamento tabella C1

In questa tabella è riportato il resoconto della revisione del codice. I requisiti in riferimento alle tabelle `A3`, `A2.4`, `B3` e `B4 C+E` sono aggiornati automaticamente da SOFTEMA.

Considerando che la struttura del sistema è stata implementata in modo appropriato sia a livello hardware che software, è possibile registrare l'esito della verifica dei requisiti `R2` e `R4`. E' importante inserire la data e il responsabile della verifica.

In figura 4.28 è riportata la compilazione adeguata della tabella.

Nr	Beschreibung	Referenzblatt	Verifikation	Kommentar
R1	Sind die vereinbarten fehlervermeidenden Maßnahmen	A3 Maßnahmen	OK	Sono state rispettate le misure concordate contro gli errori, gli strumenti e le regole di programmazione?
R2	Ist der Systemaufbau der Hardware umgesetzt worden?	A2.3 Systemaufbau	OK	È stata implementata la struttura di sistema dell'hardware?
R3	Ist die Verschaltung der I/O-Signale korrekt umgesetzt?	A2.4 IO-Liste	OK	Il collegamento dei segnali I/O è implementato correttamente?
R4	Ist die Architektur des Sicherheitsprogramms eingehalten	B1 Architektur Sicherheitspr.	OK	L'architettura del programma di sicurezza è stata rispettata?
R5	Ist die Modularchitektur eingehalten worden?	B3 Modularchitektur	OK	L'architettura del modulo è stata rispettata?
R6	Ist die Spezifikation der Software aus der Matrix	B4 Matrix C+E	OK	Le specifiche software della matrice sono state implementate?
€€€				
		Summe	OK	

Figura 4.28: Aggiornamento della tabella C1 per la revisione del codice e risultato della verifica.

4.5 D - Test

Conclusa la fase di analisi e verifica, se gli esiti sono stati positivi, è possibile procedere all'esecuzione dei test programmati nel piano di validazione. L'obbiettivo dei test è dimostrare che il comportamento del sistema corrisponde a quanto stabilito nelle analisi. La metodologia adottata sarà il test a scatola nera. Conclusa l'attività di testing, si possono aggiornare le tabelle in SOFTEMA con i risultati delle validazioni.

4.5.1 Normale funzionamento

4.5.1.1 Test e validazione I/O

E' necessario verificare che ogni I/O sia collegato correttamente al PLC. Per eseguire questo controllo si può agire sugli ingressi e verificare che il PLC rilevi le commutazioni in modo accurato. Il test può essere condotto monitorando online le unità I/O del PLC tramite il monitor integrato in PNOZmulti Configurator.

Se il PLC rileva le commutazioni degli ingressi, il test è da considerarsi superato. Per testare le uscite, si possono forzare gli stati con il monitor integrato e verificare che le commutazioni avvengano correttamente a livello hardware. Nel caso specifico, è obbligatorio testare una sola uscita alla volta per evitare cortocircuiti nell'alimentazione.

L'esito del test è riportato nella colonna `I/O Test` della tabella `A2.4`.

4.5.1.2 Test e validazione -SF10.1.1

Per testare la funzione di sicurezza *-SF10.1.1* si studiano tre scenari: salita, discesa e attesa. Durante ciascuna prova, si deve azionare il fungo di emergenza e verificare che i movimenti siano bloccati fino alla disattivazione della funzione di sicurezza. Il test si considera fallito se, dopo la pressione del fungo di emergenza, i movimenti risultano ancora possibili.

Per l'esecuzione del test in salita, si deve mantenere la pressione sul pulsante `Sb1`, quindi, dopo 2 secondi, è necessario premere il pulsante `Sb0`. Verificare che il motore si arresti e che i teleruttori siano disalimentati.

Per lo scenario in discesa, mantenere la pressione sul pulsante `Sb2`, quindi, dopo circa 2 secondi, premere il pulsante `Sb0`. Verificare che il motore si arresti e che i teleruttori siano disalimentati.

Infine, senza premere alcun pulsante, azionare il fungo di emergenza `Sb0`. Successivamente si devono premere in sequenza i pulsanti `Sb1` e `Sb2` verificando che il motore non si muova.

Nella tabella seguente sono riportati gli esiti dei test, la data di esecuzione e il responsabile.

ID	Descrizione	Data	Responsabile	Esito
TNF01.1	Convalida -SF10.1.1 in salita	03/09/2024	Riccardo Palmarin	OK
TNF01.2	Convalida -SF10.1.1 in discesa	03/09/2024	Riccardo Palmarin	OK
TNF01.3	Convalida -SF10.1.1 in attesa	03/09/2024	Riccardo Palmarin	OK
TNF01	Convalida dell'analisi ANF01	03/09/2024	Riccardo Palmarin	**OK**

Tabella 4.22: Test TNF01: Risultato del test della funzione di sicurezza -SF10.1.1

4.5.1.3 Test e validazione -SF20.1.1

Per testare la funzione di sicurezza *-SF20.1.1* è sufficiente considerare due scenari: rilevazione oggetto durante la salita e la discesa, richiesta di salita e discesa con un oggetto

presente nella zona pericolosa. Per simulare la presenza di un oggetto, si porrà una mano nel fascio infrarosso della barriera `BFT1` e verrà mantenuta per 2 secondi.

In riferimento al primo scenario, si mantiene la pressione sul pulsante `Sb1` e si simula l'ingresso di un oggetto nella zona pericolosa. Si deve verificare che il motore si arresti e che i teleruttori siano disalimentati. E' necessario ripetere la prova premendo il pulsante `Sb2`.

In riferimento al secondo scenario, si simula la presenza di un oggetto nella zona pericolosa e poi si preme il pulsante `Sb1`. Si deve verificare che il motore non possa muoversi. Successivamente, si ripete la prova azionando il pulsante `Sb2`.

Nella seguente tabella sono riportati gli esiti dei test, la data di esecuzione e gli estremi del responsabile.

ID	Descrizione	Data	Responsabile	Esito
TNF02.1.1	Rilevazione oggetto durante salita	03/09/2024	Riccardo Palmarin	OK
TNF02.1.2	Rilevazione oggetto durante discesa	03/09/2024	Riccardo Palmarin	OK
TNF02.2.1	Richiesta salita con oggetto	03/09/2024	Riccardo Palmarin	OK
TNF02.2.2	Richiesta discesa con oggetto	03/09/2024	Riccardo Palmarin	OK
TNF02	Convalida dell'analisi ANF02	03/09/2024	Riccardo Palmarin	**OK**

Tabella 4.23: Test TNF02: Risultati del test della funzione di sicurezza -SF20.1.1

4.5.1.4 Test e validazione -SF20.2.1

Il test della funzione di sicurezza *-SF20.2.1* è analogo a quello della *-SF20.1.1*. Per simulare un oggetto nella zona pericolosa, si pone una mano per 2 secondi nel fascio infrarosso della barriera `BFT2`. Nella tabella di seguito sono riportati gli esiti delle prove.

ID	Descrizione	Data	Responsabile	Esito
TNF03.1.1	Rilevazione oggetto durante salita	03/09/2024	Riccardo Palmarin	OK
TNF03.1.2	Rilevazione oggetto durante discesa	03/09/2024	Riccardo Palmarin	OK
TNF03.2.1	Richiesta salita con oggetto	03/09/2024	Riccardo Palmarin	OK
TNF03.2.2	Richiesta discesa con oggetto	03/09/2024	Riccardo Palmarin	OK
TNF03	Convalida dell'analisi ANF03	03/09/2024	Riccardo Palmarin	**OK**

Tabella 4.24: Test TNF03: Risultati del test della funzione di sicurezza -SF20.2.1

4.5.1.5 Test e validazione -SF30.1.1

Per testare la funzione di sicurezza *-SF30.1.1*, è necessario considerare sue scenari: attivazione dei finecorsa durante la discesa e richiesta di discesa con i finecorsa già attivi. Utilizzando un'asse di legno di possono attivare i finecorsa `FC3` e `FC4` quando necessario.

In primo luogo, si deve premere il pulsante `Sb2` per simulare la discesa. Dopo 2 secondi, si attivano i finecorsa e si deve verificare che il motore si arresti.

Successivamente, con i finecorsa attivi, si deve premere il pulsante `Sb2`. Il motore non deve muoversi.

Nella tabella seguente sono riportati gli esiti dei test, la data di esecuzione e il responsabile.

ID	Descrizione	Data	Responsabile	Esito
TNF04.1	Raggiungimento del piano terra	03/09/2024	Riccardo Palmarin	OK
TNF04.2	Richiesta discesa al piano terra	03/09/2024	Riccardo Palmarin	OK
TNF04	Convalida dell'analisi ANF04	03/09/2024	Riccardo Palmarin	**OK**

Tabella 4.25: Test TNF04: Risultati del test della funzione di sicurezza -SF30.1.1

4.5.1.6 Test e validazione -SF30.2.1

Il test della funzione di sicurezza *-SF30.2.1* è analogo a quello della *-SF30.1.1*. In questo caso si devono attivare i finecorsa `FC1` e `FC2` e si deve premere il pulsante `Sb1` per richiedere la salita. Nella seguente tabella sono riportati i risultati del test.

ID	Descrizione	Data	Responsabile	Esito
TNF05.1	Raggiungimento del primo piano	03/09/2024	Riccardo Palmarin	OK
TNF05.2	Richiesta discesa al primo piano	03/09/2024	Riccardo Palmarin	OK
TNF05	Convalida dell'analisi ANF05	03/09/2024	Riccardo Palmarin	**OK**

Tabella 4.26: Test TNF05: Risultati del test della funzione di sicurezza -SF30.2.1

4.5.1.7 Test e validazione con ingressi non previsti

Come rilevato nell'analisi condotta al paragrafo 4.4.1.6, è possibile che le funzione di sicurezza vengano bypassate mantenendo la pressione sul pulsante di riconoscimento `Sb3`. Per convalidare il comportamento del sistema è necessario ripetere tutti i test condotti fino a questo punto mantenendo la pressione sul pulsante `Sb3`.

Il comportamento atteso rimane invariato rispetto a quanto riportato per le prove precedenti. Il test è da considerarsi superato se e solo se tutte le prove hanno esito positivo. Nella tabella di seguito sono riportati gli esiti delle prove.

ID	Descrizione	Data	Responsabile	Esito
TNF01.1.b	Convalida -SF10.1.1 in salita con bypass	03/09/2024	Riccardo Palmarin	OK
TNF01.2.b	Convalida -SF10.1.1 in discesa con bypass	03/09/2024	Riccardo Palmarin	OK
TNF01.3.b	Convalida -SF10.1.1 in attesa con bypass	03/09/2024	Riccardo Palmarin	OK
TNF02.1.1.b	Rilevazione oggetto durante salita con bypass	03/09/2024	Riccardo Palmarin	OK
TNF02.1.2.b	Rilevazione oggetto durante discesa con bypass	03/09/2024	Riccardo Palmarin	OK
TNF02.2.1.b	Richiesta salita con oggetto e bypass	03/09/2024	Riccardo Palmarin	OK
TNF02.2.2.b	Richiesta discesa con oggetto e bypass	03/09/2024	Riccardo Palmarin	OK
TNF03.1.1.b	Richiesta discesa con oggetto e bypass	03/09/2024	Riccardo Palmarin	OK
TNF03.1.2.b	Rilevazione oggetto durante discesa con bypass	03/09/20244	Riccardo Palmarin	OK
TNF03.2.1.b	Richiesta salita con oggetto e bypass	03/09/2024	Riccardo Palmarin	OK
TNF03.2.2.b	Richiesta discesa con oggetto e bypass	03/09/2024	Riccardo Palmarin	OK
TNF04.1.b	Raggiungimento del piano terra con bypass	03/09/2024	Riccardo Palmarin	OK
TNF04.2.b	Richiesta discesa al piano terra con bypass	03/09/2024	Riccardo Palmarin	OK
TNF05.1.b	Raggiungimento del primo piano con bypass	03/09/2024	Riccardo Palmarin	OK
TNF05.2.b	Richiesta discesa al primo piano con bypass	03/09/2024	Riccardo Palmarin	OK
TNF06	Convalida dell'analisi ANF06	03/09/2024	Riccardo Palmarin	**OK**

Tabella 4.27: Test TNF06: Risultati del test delle funzioni di sicurezza con combinazioni di ingressi non previste

4.5.2 Funzionamento in caso di guasto

4.5.2.1 Test guasto fungo di emergenza

Per simulare il guasto delle linee di ingresso è sufficiente scollegare un cavo per volta a valle del fungo di emergenza. Le configurazioni circuitali da realizzare sono quelle riportare in figura 4.29.

In caso di guasto, il sistema deve attivare la funzione di sicurezza e inibire ogni movimento. La procedura di test da seguire dopo il guasto di ciascun contatto è descritta nella sezione 4.5.1.2.

Il test è da considerarsi superato se non viene persa la funzionalità della funzione di sicurezza.

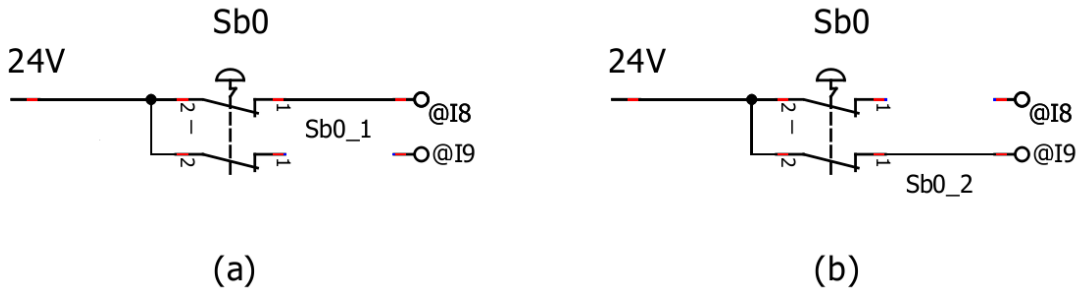


Figura 4.29: Simulazione del guasto del fungo di emergenza `Sb0`

Nella seguente tabella sono riportati gli esiti delle prova, la data di esecuzione e il responsabile.

ID	Descrizione	Data	Responsabile	Esito
TGF01.1	Convalida -SF10.1.1 in salita	03/09/2024	Riccardo Palmarin	OK
TGF01.2	Convalida -SF10.1.1 in discesa	03/09/2024	Riccardo Palmarin	OK
TGF01.3	Convalida -SF10.1.1 in attesa	03/09/2024	Riccardo Palmarin	OK
TGF01, TGF02	Convalida dell'analisi AGF01 e AGF02	03/09/2024	Riccardo Palmarin	**OK**

Tabella 4.28: Test TGF01 e TGF02: Risultati del test in caso di guasto della funzione di sicurezza -SF10.1.1

4.5.2.2 Test guasto finecorsa

Per simulare il guasto dei finecorsa è necessario scollegare un conduttore di collegamento alla volta realizzando le configurazioni circuitali illustrate in figura 4.30.

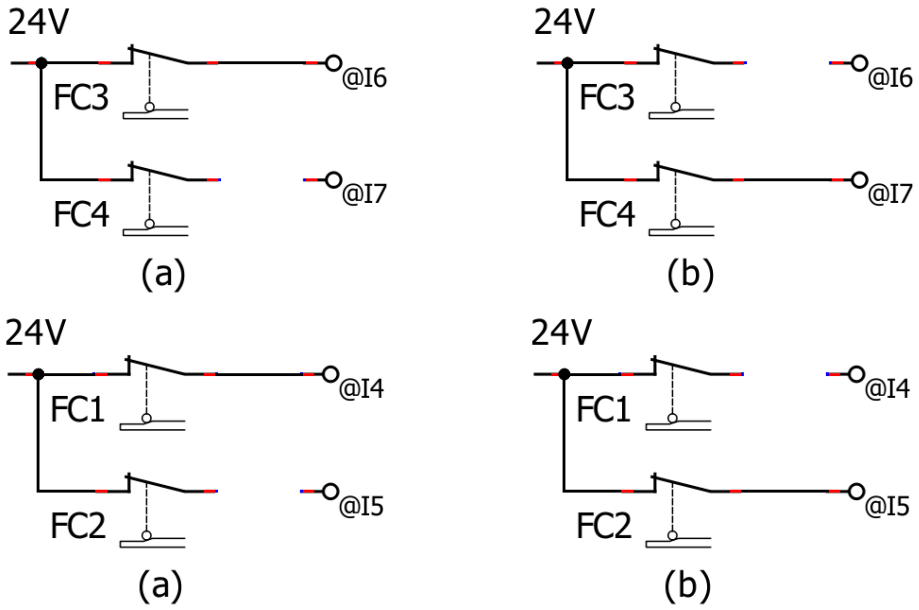


Figura 4.30: Simulazione del guasto dei finecorsa

La procedura da seguire durante i test è quella riportata alla sezione 4.5.1.5. Nella tabella seguente sono riportati i risultati delle prove.

ID	Descrizione	Data	Responsabile	Esito
TGF04.1	Raggiungimento del piano terra	03/09/2024	Riccardo Palmarin	OK
TGF04.2	Richiesta discesa al piano terra	03/09/2024	Riccardo Palmarin	OK
TGF04.3	Raggiungimento del primo piano	03/09/2024	Riccardo Palmarin	OK
TGF04.4	Richiesta discesa al primo piano	03/09/2024	Riccardo Palmarin	OK
TGF04, TGF05	Convalida dell'analisi AGF04 e AGF05	03/09/2024	Riccardo Palmarin	**OK**

Tabella 4.29: Test TGF04 e TGF05: Risultati del test in caso di guasto delle funzioni di sicurezza -SF30.X.1

4.5.2.3 Test guasto barriere di sicurezza

Analogamente alle prove precedenti, per simulare i guasti delle barriere di sicurezza è necessario scollegare un conduttore di collegamento alla volta ed eseguire i test secondo le procedure descritte alla sezione 4.5.1.3. In figura 4.31 sono riportate le configurazioni circuitali da realizzare durante le prove.

Il test è da considerarsi superato se la funzionalità della funzione di sicurezza non viene persa. Nella tabella seguente sono riportati i risultati dei test.

ID	Descrizione	Data	Responsabile	Esito
TGF06.1.1	Rilevazione oggetto durante salita	03/09/2024	Riccardo Palmarin	OK
TGF06.1.2	Rilevazione oggetto durante discesa	03/09/2024	Riccardo Palmarin	OK
TGF06.2.1	Richiesta salita con oggetto	03/09/2024	Riccardo Palmarin	OK
TGF06.2.2	Richiesta discesa con oggetto	03/09/2024	Riccardo Palmarin	OK
TGF06.3.1	Rilevazione oggetto durante salita	03/09/2024	Riccardo Palmarin	OK
TGF06.3.2	Rilevazione oggetto durante discesa	03/09/2024	Riccardo Palmarin	OK
TGF06.4.1	Richiesta salita con oggetto	03/09/2024	Riccardo Palmarin	OK
TGF06.4.2	Richiesta discesa con oggetto	03/09/2024	Riccardo Palmarin	OK
TGF06, TGF07	Convalida dell'analisi AGF03 e TGF07	03/09/2024	Riccardo Palmarin	**OK**

Tabella 4.30: Test TGF06 e TGF07: Risultati del test in caso di guasto delle funzioni di sicurezza -SF20.X.1

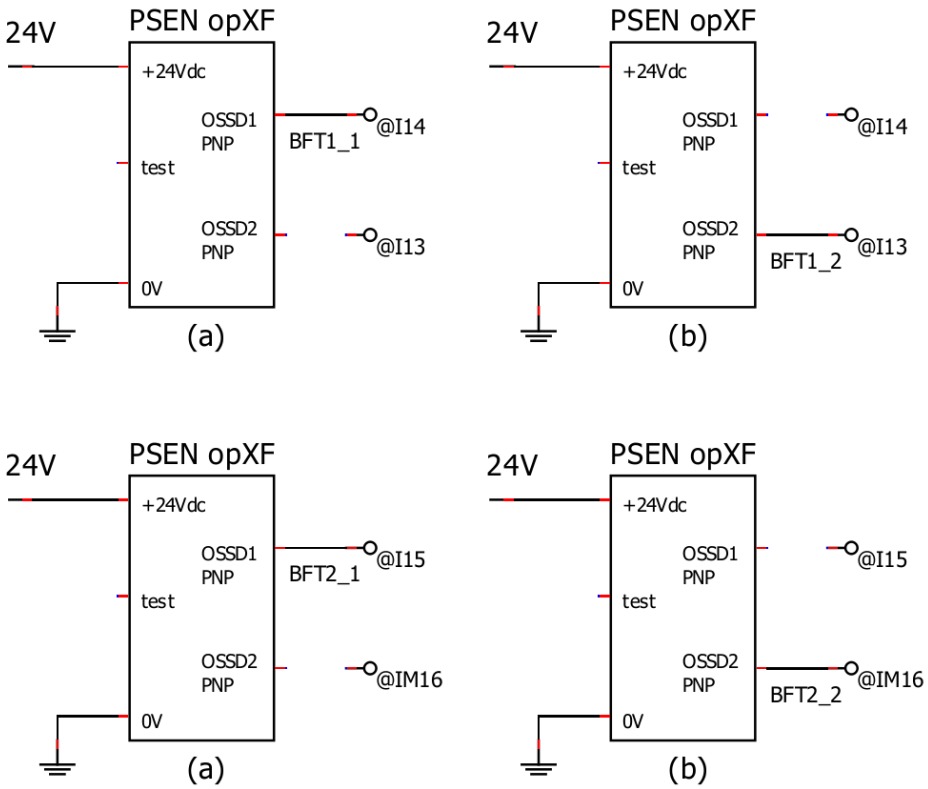


Figura 4.31: Simulazione del guasto delle barriere

4.5.3 Aggiornamento tabelle SOFTEMA

Una volta concluse le prove per validare le funzioni di sicurezza, è possibile finalizzare la validazione compilando accuratamente le tabelle in SOFTEMA. Questa operazione è molto importante, poiché costituisce la base del verbale di validazione, che ha valenza legale.

4.5.3.1 Aggiornamento tabella A2.4

In riferimento al test e alla validazione I/O di cui alla sezione 4.5.1.1, si deve compilare la colonna `IO-Test` con l'esito della prova. Se il PLC è in grado di rilevare le commutazioni degli ingressi e di attuare le commutazioni delle uscite, si deve selezionare la voce `OK`. Nel caso in esame tutti gli ingressi e tutte le uscite hanno superato la prova. E' importante riportare la data e gli estremi di chi ha condotto il test.

Durante i test del comportamento in caso di guasto, è stata valutata la capacità del PLC di individuare avarie nei collegamenti degli ingressi tramite un controllo a due canali. Se il PLC non è stato in grado di rilevare un guasto in un canale, selezione la voce `not OK` nella colonna `DIAG-Test`; altrimenti selezionare `OK`. Nell'applicazione in esame, le uscite non sono dotate di feedback, pertanto non è necessario selezionare alcuna voce. E' utile inserire un commento per specificare questo aspetto. Anche in questo caso si devono indicare i dati dell'esecuzione del test.

4.5.3.2 Aggiornamento tabella A4

Si possono validare i requisiti normativi adottati durante lo sviluppo del progetto. Il ciclo di vita del progetto ha compreso attività di validazione e verifica, integrate da una documentazione dettagliata delle specifiche e e delle fasi del ciclo stesso. Inoltre, è stata implementata una programmazione modulare e strutturata e sono stati eseguiti dei test funzionali completi. Per tali motivi si possono validare i requisiti da `A1` a `A5`.

Le funzioni di sicurezza sono state specificate correttamente e sono stati definiti i criteri di prestazione. L'architettura hardware è dotata di interfacce di segnale esterne ed è in grado di rilevare e controllare i guasti esterni. Si possono convalidare i requisiti da `A6` a `A9`.

Gli strumenti e l'hardware impiegati sono più che adeguati all'applicazione, pertanto si può validare il requisito `A10`. La semplicità del programma e dell'applicazione consente di escludere il requisito `A12` poiché non sono implementate allocazioni di memoria e non sono utilizzati diversi tipi di dati. L'ambiente di sviluppo e la CPU attuano dei controlli sul programma durante la compilazione durante il runtime. Il codice è stato implementato con blocchi funzionali provenienti dalla libreria validata sviluppata da Pilz e il linguaggio LVL utilizzato è conforme ai requisiti per l'applicazione. Per questi motivi si possono validare i requisiti da `A13` a `A15`.

Per descrivere i dati e il flusso di controllo sono state impiegate le tabelle in SOFTEMA. I blocchi funzionali validati utilizzati per la programmazione modulare e strutturata hanno la lunghezza minima. Le assegnazioni di ingressi e uscite sono realizzate una

sola volta nel programma seguendo l'architettura a tre stadi. Il SRASW e il software non correlato alla sicurezza sono codificati in blocchi funzionali diversi e non sono presenti collegamenti logici tra dati non rilevanti per la sicurezza e quelli relativi alla sicurezza. E' possibile convalidare i requisiti da `A16` a `A24`.

Il codice è leggibile, comprensibile e testabile. Sono state seguite le linee guida di programmazione riportate nella norma *EN IEC 61508*. L'ambiente di sviluppo implementa dei controlli di integrità. Il codice è stato testato mediante simulazione e tramite controllo e analisi del flusso di dati. Il metodo di validazione impiegato è il test a scatola nera. Si possono, quindi, validare i requisiti da `A25` a `A30`.

E' possibile escludere il requisito `A31` poiché sono stati utilizzati esclusivamente dati di tipo booleano. I test sono stati pianificati opportunamente e i test I/O sono stati superati. E', quindi, possibile validare i requisiti `A32` e `A33`.

Infine, ogni fase del ciclo di vita è stata documentata. La documentazione è completa, disponibile, leggibile e comprensibile. La documentazione del codice all'interno del codice sorgente è adeguata ai requisiti normativi. Sono state implementate attività di revisione, ispezione e sono state introdotte procedure di backup dei dati per identificare e archiviare tutti i documenti relativi all'SRASW. Pertanto, si possono validare i requisiti da `A34` a `A38`.

Si devono riportare gli estremi dell'addetto alla validazione e la data di modifica della tabella. In figura 4.32 è riportato l'esito della validazione dei requisiti normativi.

Summe	OK
Datum	27/08/2024
Name	Riccardo Palmarin
Signatur	Riccardo Palmarin

Figura 4.32: Risultato della validazione dei requisiti normativi. Aggiornamento della tabella A4.

4.5.3.3 Aggiornamento tabella B4 C+E

La colonna `Validierung` della tabella *B4 Matrix C+E* va compilata sulla base degli esiti delle prove condotte alle sezioni 4.5.1 e 4.5.2. Poiché ogni caso di test ha avuto esito positivo, si possono validare tutte le righe della tabella. In figura 4.33 è riportato il risultato della validazione.

x	OK	OK
Datum	17/08/2024	27/08/2024
Name	Riccardo Palmarin	Riccardo Palmarin
Signatur	Riccardo Palmarin	Riccardo Palmarin

Figura 4.33: Risultato della validazione delle funzioni di sicurezza. Aggiornamento della tabella B4 C+E.

4.5.3.4 *Aggiornamento tabella B4 Kompakt*

Nella colonna `Validierung` della tabella *B4 Matrix kompakt* sono validati i comportamenti delle uscite. Se nel corso delle prove non sono state rilevate anomalie nel funzionamento delle uscite, si può selezionare la voce `OK`, come nel caso in esame. Nella figura 4.34 è riportata la tabella compilata con l'esito della validazione.

Nr	Ausgang	Beschreibung	B0: Alle	B1: Automatik	SF (Prio)	Verifikation	Validierung
E1	O1: KM1 [Mb0.O0]	Teleruttore salita n1	OFF: not (*IM1*) ESTOP_1.EMST_OK OFF: ((*I5*) Sb0_1 AND not ((*I6*) Sb0_2)) OR ((*I6*) Sb0_2 AND not ((*I5*) Sb0_1)) OFF: ((*I10*) BFT_1_2 AND not ((*I11*) BFT_1_1)) OR ((*I11*) BFT_1_1 AND not ((*I10*) BFT_1_2)) OFF: ((*I12*) BFT_2_1 AND not ((*I13*) BFT_2_2)) OR ((*I13*) BFT_2_2 AND not ((*I12*) BFT_2_1))	OFF: not (*IM2*) SF_BFT_1.BFT_OK_ OFF: not (*IM3*) SF_BFT_2.BFT_OK_ OFF: not (*IM5*) SF_FC_2.FC_OK_ OFF:	B0:SF1 (1),TF_ESTOP (1),TF_BFT_1 (2),TF_BFT_2 (2), B1:SF2 (2),SF3 (2),SF5 (2),TF_FC_2 (2),	OK	OK
E2	O2: KM2 [Mb0.O1]	Teleruttore salita n2	OFF: not (*IM1*) ESTOP_1.EMST_OK OFF: ((*I5*) Sb0_1 AND not ((*I6*) Sb0_2)) OR ((*I6*) Sb0_2 AND not ((*I5*) Sb0_1)) OFF: ((*I10*) BFT_1_2 AND not ((*I11*) BFT_1_1)) OR ((*I11*) BFT_1_1 AND not ((*I10*) BFT_1_2)) OFF: ((*I12*) BFT_2_1 AND not ((*I13*) BFT_2_2)) OR ((*I13*) BFT_2_2 AND not ((*I12*) BFT_2_1))	OFF: not (*IM2*) SF_BFT_1.BFT_OK_ OFF: not (*IM3*) SF_BFT_2.BFT_OK_ OFF: not (*IM5*) SF_FC_2.FC_OK_ OFF:	B0:SF1 (1),TF_ESTOP (1),TF_BFT_1 (2),TF_BFT_2 (2), B1:SF2 (2),SF3 (2),SF5 (2),TF_FC_2 (2),	OK	OK
E3	O3: KM3 [Mb0.O2]	Teleruttore discesa n1	OFF: not (*IM1*) ESTOP_1.EMST_OK OFF: ((*I5*) Sb0_1 AND not ((*I6*) Sb0_2)) OR ((*I6*) Sb0_2 AND not ((*I5*) Sb0_1)) OFF: ((*I10*) BFT_1_2 AND not ((*I11*) BFT_1_1)) OR ((*I11*) BFT_1_1 AND not ((*I10*) BFT_1_2)) OFF: ((*I12*) BFT_2_1 AND not ((*I13*) BFT_2_2)) OR ((*I13*) BFT_2_2 AND not ((*I12*) BFT_2_1))	OFF: not (*IM3*) SF_BFT_2.BFT_OK_ OFF: not (*IM3*) SF_BFT_2.BFT_OK_ OFF: not (*IM4*) SF_FC_1.FC_OK_ OFF:	B0:SF1 (1),TF_ESTOP (1),TF_BFT_1 (2),TF_BFT_2 (2), B1:SF2 (2),SF3 (2),SF4 (2),TF_FC_1 (2),	OK	OK
E4	O4: KM4 [Mb0.O3]	Teleruttore discesa n2	OFF: not (*IM1*) ESTOP_1.EMST_OK OFF: ((*I5*) Sb0_1 AND not ((*I6*) Sb0_2)) OR ((*I6*) Sb0_2 AND not ((*I5*) Sb0_1)) OFF: ((*I10*) BFT_1_2 AND not ((*I11*) BFT_1_1)) OR ((*I11*) BFT_1_1 AND not ((*I10*) BFT_1_2)) OFF: ((*I12*) BFT_2_1 AND not ((*I13*) BFT_2_2)) OR ((*I13*) BFT_2_2 AND not ((*I12*) BFT_2_1))	OFF: not (*IM3*) SF_BFT_2.BFT_OK_ OFF: not (*IM3*) SF_BFT_2.BFT_OK_ OFF: not (*IM4*) SF_FC_1.FC_OK_ OFF:	B0:SF1 (1),TF_ESTOP (1),TF_BFT_1 (2),TF_BFT_2 (2), B1:SF2 (2),SF3 (2),SF4 (2),TF_FC_1 (2),	OK	OK
€€€€							
					Summe	OK	OK
					Datum	20/08/2024	27/08/2024
					Name	Riccardo Palmarin	Riccardo Palmarin
					Signatur	Riccardo Palmarin	Riccardo Palmarin

Figura 4.34: Risultato della validazione del comportamento delle uscite. Aggiornamento della tabella B4 compatta.

4.5.3.5 *Aggiornamento tabella A1*

Le validazioni precedenti hanno avuto esito positivo, quindi sulla base delle prove condotte si possono riportare gli esiti delle validazioni delle funzioni di sicurezza.

In figura 4.35 è riportato il risulto della validazione.

x	OK
Datum	27/08/2024
Name	Riccardo Palmarin
Signatur	Riccardo Palmarin

Figura 4.35: Risultato della validazione delle funzioni di sicurezza. Aggiornamento della tabella A1.

4.5.3.6 *Aggiornamento tabella D1*

La compilazione della tabella *D1 Validierung* segna la conclusione dell'attività di validazione. I test dei dispositivi periferici hanno avuto esito positivo e i sensori sono stati controllati opportunamente. Si possono validare le righe `V8` e `V9`.

Si possono validare anche i requisiti da `D1` a `D8` in quanto sono disponibili i PDF di tutti i software di sicurezza, delle configurazioni hardware, sono stati archiviati i manuali di tutti i componenti del sistemi e sono stati rispettati le normative di riferimento.

In figura 4.36 è riportato il risultato della validazione del software di sicurezza.

Nr	Beschreibung	Referenzblatt	Validierung
	Wurden die Aktivitäten durchgeführt?		
V1	Validierung Sicherheitsfunktionen (D1)	A1 Sicherheitsfunktionen	OK
V2	Validierung I/O-Check (D1)	A2.4 IO-Liste	OK
V3	Validierung normativer Anforderungen (D1)	A4 Anforderungen	OK
V4	Verifikation der Modularchitektur (V1)	B3 Modularchitektur	OK
V5	Verifikation der Matrix (V1)	B4 Matrix C+E	OK
V6	Validierung Matrix (D1)	B4 Matrix C+E	OK
V7	Verifikation Codereview	C1 Codereview	OK
V8	Prüfung der Peripheriegeräte		OK
V9	Prüfung der Sensoren		OK
	Ist die Dokumentation komplett?		
D1	Dokumente des V-Modells aus diesem		OK
D2	PDF-Ausdruck aller sicherheitsrelevanten Software		OK
D3	PDF-Ausdruck der Hardwarekonfiguration (mit allen		OK
D4	Archivierung der Handbücher aller		OK
D5	PDF-Ausdruck der Konfiguration von Peripheriegeräten		OK
D6	Abnahmevorschriften der Hersteller (z.B.		OK
D7	Einzuhaltende Vorgaben aus C-Normen		OK
D8	Einzuhaltende Vorgaben aus B-Normen		OK
€€€			
		Summe	OK
		Datum	27/08/2024
		Name	Riccardo Palmarin
		Signatur	Riccardo Palmarin

Figura 4.36: Risultato della validazione del software. Aggiornamento della tabella D1.

Capitolo 5

Risultati

Al termine delle attività di validazione e dopo aver completato la compilazione delle tabelle in SOFTEMA, è importante raggruppare, organizzare e archiviare tutta la documentazione prodotta. Durante le attività di verifica e convalida condotte non sono state rilevate criticità nella programmazione e nell'implementazione del software, pertanto la validazione del SRASW in esame ha avuto esito positivo.

Il software soddisfa tutti i requisiti normativi richiesti per implementare delle funzioni di sicurezza con PLr *d* o SIL3. Nella tabella seguente riporta un riepilogo delle attività svolte, da inserire nel verbale di validazione finale.

Descrizione	Verifica	Validazione	Data	Responsabile
-SF10.1.1	OK	OK	03/09/2024	Riccardo Palmarin
-SF20.1.1	OK	OK	03/09/2024	Riccardo Palmarin
-SF20.2.1	OK	OK	03/09/2024	Riccardo Palmarin
-SF30.1.1	OK	OK	03/09/2024	Riccardo Palmarin
-SF30.2.1	OK	OK	03/09/2024	Riccardo Palmarin
Requisiti di sicurezza del software	OK	OK	03/09/2024	Riccardo Palmarin
Architettura del software	OK	OK	03/09/2024	Riccardo Palmarin
Progettazione del software	OK	OK	03/09/2024	Riccardo Palmarin
Codice, dati e performance	OK	OK	03/09/2024	Riccardo Palmarin
Software SRASW	**OK**	**OK**	03/09/2024	Riccardo Palmarin

Poiché la validazione è stata completata con esito positivo, non è richiesto di documentare o specificare gli interventi di modifica del software.

Si sottolinea l'importanza di riportare in dettaglio le procedure con le quali vengono condotti i test e di documentare accuratamente eventuali esiti negativi.

Nell'appendice 6 è inclusa una checklist che risulta particolarmente utile per guidare il processo di validazione di SRASW con complessità limitata. Si può constatare la corrispondenza tra la checklist sviluppata e il processo di validazione implementato nel contesto della presente tesi.

Capitolo 6

Conclusioni

Nel contesto dei software applicativi relativi alla sicurezza programmati con un linguaggio a variabilità limitata, il processo di validazione si è rivelato particolarmente complesso e articolato. Una solida conoscenza delle normative di riferimento e un'opportuna programmazione delle attività da svolgere sono essenziali per garantire che ogni aspetto venga adeguatamente considerato e gestito.

L'adozione di un metodo di lavoro preciso e strutturato consente di condurre l'intero processo in modo efficiente e ripetibile. Alla conclusione di ciascuna fase, è fondamentale sviluppare la documentazione necessaria e revisionare quella precedentemente prodotta. L'applicazione di strumenti software come SOFTEMA facilita notevolmente il tracciamento della validazione e la produzione della documentazione richiesta dalle principali normative.

L'implementazione del processo di verifica e validazione del software durante la fase di progettazione e sviluppo risulta più efficace per ottimizzare la codifica e le prestazioni del software. In tal caso, è importante documentare le attività correttive in modo opportuno e ripetere la validazione successivamente alle modifiche.

Nel caso specifico, si potrebbe sviluppare un blocco funzionale relativo alla sicurezza parametrizzabile, in modo da facilitarne l'applicazione a sistemi simili con maggiore agilità. Il processo per validare e certificare il blocco funzionale è quello descritto nel contesto della presente tesi. Durante lo sviluppo di altri software, il blocco funzionale validato potrebbe essere impiegato, limitando l'attività di validazione alla sola esecuzione dei test funzionali in scenari di guasto.

Il processo di validazione implementato è conforme al *Regolamento UE 2023/1230*, pertanto la presente tesi può essere utilizzata come linea guida applicativa per la validazione di SRASW programmati in linguaggi LVL.

La tesi potrebbe essere ampliata attraverso lo studio della progettazione e della validazione di un blocco funzionale appositamente sviluppato per l'applicazione in un linguaggio a variabilità completa.

Bibliografia

[1] CEN. Appendice D: Lista dei guasti. In *UNI EN ISO 13849-2:2013 - Sicurezza del macchinario - Parti dei sistemi di comando legate alla sicurezza - Parte 2: Validazione*. UNI, 2013.

[2] CEN. Processo di validazione. In *UNI EN ISO 13849-2:2013 - Sicurezza del macchinario - Parti dei sistemi di comando legate alla sicurezza - Parte 2: Validazione*, pages 1–6. UNI, 2013.

[3] CEN. *UNI EN ISO 13849-2:2013 - Sicurezza del macchinario - Parti dei sistemi di comando legate alla sicurezza - Parte 2: Validazione*. UNI, 2013.

[4] CEN. Validazione del software relativo alla sicurezza. In *UNI EN ISO 13849-2:2013 - Sicurezza del macchinario - Parti dei sistemi di comando legate alla sicurezza - Parte 2: Validazione*, page 13. UNI, 2013.

[5] CEN. Validazione mediante analisi. In *UNI EN ISO 13849-2:2013 - Sicurezza del macchinario - Parti dei sistemi di comando legate alla sicurezza - Parte 2: Validazione*, pages 6–7. UNI, 2013.

[6] CEN. Validazione mediante test. In *UNI EN ISO 13849-2:2013 - Sicurezza del macchinario - Parti dei sistemi di comando legate alla sicurezza - Parte 2: Validazione*, pages 7–8. UNI, 2013.

[7] CEN. Considerazione delle avarie, esclusione delle avarie. In *UNI EN ISO 13849-1:2023 - Sicurezza del macchinario - Parti dei sistemi di comando legate alla sicurezza - Parte 1: Principi generali per la progettazione*, page 42. UNI, 2023.

[8] CEN. Requisiti di sicurezza del software. In *UNI EN ISO 13849-1:2023 - Sicurezza del macchinario - Parti dei sistemi di comando legate alla sicurezza - Parte 1: Principi generali per la progettazione*, pages 22–28. UNI, 2023.

[9] CEN. *UNI EN ISO 13849-1:2023 - Sicurezza del macchinario - Parti dei sistemi di comando legate alla sicurezza - Parte 1: Principi generali per la progettazione*. UNI, 2023.

[10] IEC CENELEC. 7.18 verification. In *BS EN 61508-1:2010 - Functional safety of electrical/electronic/programmable electronic safety-realated systems - Part 1: General Requirements*, pages 49–50. BS, 2010.

[11] IEC CENELEC. 7.3 validation plan for software aspects of system safety. In *BS EN 61508-3:2010 - Functional safety of electrical/electronic/programmable electronic safety-realated systems - Part 3: Software requirements*, page 24. BS, 2010.

[12] IEC CENELEC. 7.7 software aspects of system safety validation. In *BS EN 61508-3:2010 - Functional safety of electrical/electronic/programmable electronic safety-realated systems - Part 3: Software requirements*, pages 37–38. BS, 2010.

[13] IEC CENELEC. 7.7 system validation. In *BS EN 61508-2:2010 - Functional safety of electrical/electronic/programmable electronic safety-realated systems - Part 2: Requirements for electrical/electronic/ programmable electronic safety-related systems*, page 42. BS, 2010.

[14] IEC CENELEC. 7.9 software verification. In *BS EN 61508-3:2010 - Functional safety of electrical/electronic/programmable electronic safety-realated systems - Part 3: Software requirements*, pages 41–43. BS, 2010.

[15] IEC CENELEC. 7.9 verification. In *BS EN 61508-2:2010 - Functional safety of electrical/electronic/programmable electronic safety-realated systems - Part 2: Requirements for electrical/electronic/ programmable electronic safety-related systems*, page 44. BS, 2010.

[16] IEC CENELEC. *BS EN 61508-1:2010 - Functional safety of electrical/electronic/programmable electronic safety-realated systems - Part 1: General Requirements*. BS, 2010.

[17] IEC CENELEC. *BS EN 61508-2:2010 - Functional safety of electrical/electronic/programmable electronic safety-realated systems - Part 2: Requirements for electrical/electronic/ programmable electronic safety-related systems*. BS, 2010.

[18] IEC CENELEC. *BS EN 61508-3:2010 - Functional safety of electrical/electronic/programmable electronic safety-realated systems - Part 3: Software requirements*. BS, 2010.

[19] IEC CENELEC. 5 specification of a safety function. In *CEI EN IEC 62061:2021 - Sicurezza del macchinario - Sicurezza funzionale dei sistemi di comando e controllo relativi alla sicurezza*, pages 34–37. IEC, 2021.

[20] IEC CENELEC. 8 software. In *CEI EN IEC 62061:2021 - Sicurezza del macchinario - Sicurezza funzionale dei sistemi di comando e controllo relativi alla sicurezza*, pages 62–77. IEC, 2021.

[21] IEC CENELEC. 8 software. In *CEI EN IEC 62061:2021 - Sicurezza del macchinario - Sicurezza funzionale dei sistemi di comando e controllo relativi alla sicurezza*, pages 77–87. IEC, 2021.

[22] IEC CENELEC. Allegato F: regole di codifica. In *CEI EN IEC 62061:2021 - Sicurezza del macchinario - Sicurezza funzionale dei sistemi di comando e controllo relativi alla sicurezza*. IEC, 2021.

[23] IEC CENELEC. *CEI EN IEC 62061:2021 - Sicurezza del macchinario - Sicurezza funzionale dei sistemi di comando e controllo relativi alla sicurezza*. IEC, 2021.

[24] Giacomo Comunian. *Progetto e realizzazione di un sistema di test delle funzioni di sicurezza*. Tesi di laurea triennale, Università degli studi di Padova, Dipartimento di Tecnica e Gestione dei sistemi Industriali, 2023.

[25] Parlamento Europeo e Consiglio Europeo. Allegato 1, parte b, paragrafo 17. In *Regolamento Macchine UE 2023/1230*, page 41. Gazzetta ufficiale dell'Unione Europea, 2023.

[26] Parlamento Europeo e Consiglio Europeo. Allegato 2, paragrafo 18. In *Regolamento Macchine UE 2023/1230*, page 43. Gazzetta ufficiale dell'Unione Europea, 2023.

[27] Parlamento Europeo e Consiglio Europeo. Allegato 4, comma m. In *Regolamento Macchine UE 2023/1230*, page 83. Gazzetta ufficiale dell'Unione Europea, 2023.

[28] Parlamento Europeo e Consiglio Europeo. Atti legislativi, paragrafo 19. In *Regolamento Macchine UE 2023/1230*, page 3. Gazzetta ufficiale dell'Unione Europea, 2023.

[29] Parlamento Europeo e Consiglio Europeo. *Regolamento Macchine UE 2023/1230*. Gazzetta ufficiale dell'Unione Europea, 2023.

[30] Manuel Josef Gerold. *Softwareanforderungen" funktionaler Sicherheit" auf Anwenderebene: inklusive Analyse eines Softwareassistenten*. PhD thesis, FH CAMPUS 02 (CAMPUS 02 Fachhochschule der Wirtschaft), 2020.

Appendice A

Nella tabella seguente è riportata la checklist sviluppata per condurre la validazione di un software applicativo relativo alla sicurezza nelle ipotesi esplicitate al capitolo 3.Si raccomanda di seguire l'ordine delle attività suggerito per garantire una conduzione ottimale del processo.

ID	Descrizione	Data	Responsabile	Eseguito
A	**FASE PRELIMINARE**	06/08/2024	Riccardo Palmarin	☐
A.1	*Impostazione del progetto in SOFTEMA*	06/08/2024	Riccardo Palmarin	☐
A.1.a	Designazione dei ruoli e del team di lavoro	06/08/2024	Riccardo Palmarin	☐
A.1.b	Riferimento alla documentazione principale	06/08/2024	Riccardo Palmarin	☐
A.2	*Individuazione e studio delle funzioni di sicurezza*	06/08/2024	Riccardo Palmarin	☐
A.2.a	Associazione tra nomenclatura degli schemi e software	06/08/2024	Riccardo Palmarin	☐
A.3	*Sviluppare la lista dei guasti*	06/08/2024	Riccardo Palmarin	☐
A.3.a	Considerazione delle liste normate	06/08/2024	Riccardo Palmarin	☐
A.3.b	Sviluppo interno di scenari di guasto aggiuntivi	06/08/2024	Riccardo Palmarin	☐
A.4	*Definizione delle regole di codifica*	06/08/2024	Riccardo Palmarin	☐
A.5	*Aggiornamento delle tabelle in SOFTEMA*	06/08/2024	Riccardo Palmarin	☐
A.5.a	A1 - funzioni di sicurezza	06/08/2024	Riccardo Palmarin	☐
A.5.b	A2.4 - lista I/O	06/08/2024	Riccardo Palmarin	☐
A.5.c	A3 - misure	06/08/2024	Riccardo Palmarin	☐
A.5.d	A4 - requisiti	06/08/2024	Riccardo Palmarin	☐

ID	Descrizione	Data	Responsabile	Eseguito
B	**PIANO DI VALIDAZIONE**	12/08/2024	Riccardo Palmarin	☐
B.1	*Normale funzionamento*	12/08/2024	Riccardo Palmarin	☐
B.1.a	Pianificazione delle analisi ANF	12/08/2024	Riccardo Palmarin	☐
B.1.b	Pianificazione delle verifiche VNF	12/08/2024	Riccardo Palmarin	☐
B.1.c	Pianificazione dei test TNF	12/08/2024	Riccardo Palmarin	☐
B.2	*Funzionamento in caso di guasto*	12/08/2024	Riccardo Palmarin	☐
B.2.a	Pianificazione delle analisi AGF	12/08/2024	Riccardo Palmarin	☐
B.2.b	Pianificazione delle verifiche VGF	12/08/2024	Riccardo Palmarin	☐
B.2.c	Pianificazione dei test TGF	12/08/2024	Riccardo Palmarin	☐
B.3	*Pianificazione verifiche normate*	12/08/2024	Riccardo Palmarin	☐
C	**ANALISI E VERIFICHE**	13/08/2024	Riccardo Palmarin	☐
C.1	*Analisi normale funzionamento*	13/08/2024	Riccardo Palmarin	☐
C.2	*Verifiche normale funzionamento*	13/08/2024	Riccardo Palmarin	☐
C.3	*Analisi in caso di guasto*	16/08/2024	Riccardo Palmarin	☐
C.4	*Pianificazione verifiche normate*	16/08/2024	Riccardo Palmarin	☐
C.5	*Verifiche normate*	17/08/2024	Riccardo Palmarin	☐
C.6	*Aggiornamento tabelle in SOFTEMA*	19/08/2024	Riccardo Palmarin	☐
C.6.a	A3 - misure	19/08/2024	Riccardo Palmarin	☐
C.6.b	B3 - architettura dei moduli	19/08/2024	Riccardo Palmarin	☐
C.6.c	A2.4 - lista I/O	19/08/2024	Riccardo Palmarin	☐
C.6.e	B4 - matrice C+E	19/08/2024	Riccardo Palmarin	☐
C.6.f	B4 - matrice compatta	19/08/2024	Riccardo Palmarin	☐
C.6.g	C1 - revisione del codice	19/08/2024	Riccardo Palmarin	☐
C.7	*Revisione della documentazione prodotta*	12/08/2024	Riccardo Palmarin	☐

ID	Descrizione	Data	Responsabile	Eseguito
D	**TEST**	03/09/2024	Riccardo Palmarin	☐
D.1	*Test in normale funzionamento*	03/09/2024	Riccardo Palmarin	☐
D.1.a	Test e validazione degli I/O	03/09/2024	Riccardo Palmarin	☐
D.1.b	Test e validazione delle funzioni di sicurezza	03/09/2024	Riccardo Palmarin	☐
D.2	*Test in scenario di guasto*	03/09/2024	Riccardo Palmarin	☐
D.2.a	Test e validazione del guasto dei componenti	03/09/2024	Riccardo Palmarin	☐
D.3	*Aggiornamento tabelle in software*	03/09/2024	Riccardo Palmarin	☐
D.3.a	A2.4 - lista I/O	03/09/2024	Riccardo Palmarin	☐
D.3.b	A4 - requisiti	03/09/2024	Riccardo Palmarin	☐
D.3.c	B4 - matrice C+E	03/09/2024	Riccardo Palmarin	☐
D.3.d	B4 - matrice compatta	03/09/2024	Riccardo Palmarin	☐
D.3.e	A1 - funzioni di sicurezza	03/09/2024	Riccardo Palmarin	☐
D.3.f	D1 - validazione	03/09/2024	Riccardo Palmarin	☐
D.4	*Report di validazione*	03/09/2024	Riccardo Palmarin	☐
D.5	*Stesura verbale di validazione*	03/09/2024	Riccardo Palmarin	☐

Ringraziamenti

Vorrei esprimere la mia più profonda riconoscenza a tutte le persone che hanno contribuito, con il loro supporto, alla realizzazione del mio percorso universitario e alla stesura di questa tesi.

In primo luogo, desidero ringraziare il mio relatore, Diego Dainese. La sua passione e la sua competenza hanno saputo guidarmi e ispirarmi, non solo nella stesura di questa tesi, ma anche nella mia carriera accademica e professionale. Sono grato per la fiducia dimostrata durante tutto il percorso di ricerca e scrittura.

Esprimo la mia profonda gratitudine alla mia famiglia per il sostegno incondizionato che mi ha permesso di affrontare e superare i momenti più difficili di questo percorso. Un sentito ringraziamento va ai miei genitori e a mio fratello, il cui amore e incoraggiamento incessanti hanno fornito la forza e la motivazione necessarie per perseverare nei momenti più ardui e che hanno condiviso con me anche i momenti migliori. Un ringraziamento speciale è dovuto ai miei nonni, che hanno rappresentato il faro del mio percorso accademico, offrendo incoraggiamento e sostegno costanti e dimostrando sempre una profonda fiducia nelle mie capacità. Un grazie di cuore va ai miei zii e ai miei cugini, il cui sostegno è stato cruciale per rialzarmi dopo ogni caduta e il cui entusiasmo ha contribuito a rendere questo percorso ancora più gioioso e significativo.

Un pensiero speciale va anche ai miei due amici, Adrian e Mattia, con cui ho condiviso i momenti più significativi e memorabili della mia carriera accademica. Sono profondamente grato per la loro presenza costante e per aver reso questo percorso ancora più speciale con la loro sincera amicizia.

Vorrei anche esprimere un ringraziamento speciale a coloro che hanno cercato di ostacolarmi in ogni modo e a chi ha deciso di andarsene. Le loro parole e le loro azioni hanno alimentato la mia determinazione e trasformato le mie insicurezze in energia per raggiungere il massimo.

Infine, desidero esprimere un sincero ringraziamento a me stesso. Voglio ringraziarmi per aver sempre creduto in me stesso e per aver affrontato con carattere ogni sfida. Grazie a me stesso per aver dedicato tanto impegno e per aver lavorato instancabilmente, senza mai concedermi pause. Apprezzo la mia resilienza e il mio rifiuto di arrendermi dinanzi alle difficoltà. Sono grato per il mio impegno costante nel cercare di fare le scelte giuste e per mantenere la mia integrità. Voglio esprimere la mia gratitudine per essere stato autentico e fedele a me stesso in ogni momento di questo percorso.

Elenco delle figure

Elenco delle tabelle