

Certified Tester Specialist Automotive Software Tester (CT-AuT) Syllabus

Versione 2018 (2.0.2) 4 Luglio 2018

International Software Testing Qualifications Board

Avviso sul Copyright

Questo documento può essere copiato del tutto o in parte, se la fonte è riconosciuta.

Copyright © International Software Testing Qualifications Board (di seguito chiamata ISTQB®).

Copyright © 2017, German Testing Board e.V. (GTB)

Gli autori e il German Testing Board hanno concordato i seguenti termini di utilizzo:

- Ogni individuo e training provider può utilizzare il syllabus come base per la formazione se i proprietari del copyright sono riconosciuti e menzionati come fonte e proprietari del copyright. Inoltre, il syllabus può essere utilizzato per scopi di marketing solo dopo l'accREDITAMENTO da parte dell'ISTQB® member board.
- Ogni individuo o gruppo di individui può utilizzare il syllabus come base per articoli, libri o altre pubblicazioni che ne derivino se gli autori e il German Testing Board sono menzionati come fonte e proprietari del copyright.
- Il syllabus comprese tutte le sue parti è protetto da copyright. L'uso – se non è esplicitamente permesso dalla German copyright law (UrhG) – è permesso solamente previa approvazione delle persone autorizzate. Questo si applica specificatamente a copie, adattamenti, traduzioni, microfilm, archiviazioni ed elaborazioni in sistemi elettronici, messi a disposizione del pubblico.

Marchi Registrati

- CTFL® è un marchio registrato del German Testing Board (GTB) e.V. solo nell'UE.
- GTB® è un marchio registrato del German Testing Board (GTB) e.V. solo nell'UE.
- ISTQB® è un marchio registrato dell'International Software Testing Qualifications Board
- Automotive SPICE® è un marchio registrato della German Association of the Automotive Industry (VDA)

Avviso di esclusione e limitazione di responsabilità

Non viene fornita alcuna dichiarazione o garanzia che le informazioni siano tecnicamente accurate o sufficienti o siano conformi a qualsiasi statuto, norma o regolamento governativo, e inoltre nessuna dichiarazione o garanzia è fornita sulla commerciabilità o idoneità per qualsiasi scopo particolare o contro la violazione dei diritti di proprietà intellettuale. In nessun caso ISTQB® o GTB® saranno ritenuti responsabili per la perdita di profitti o altri incidenti o danni consequenziali. ISTQB® e GTB® avvertono espressamente che qualsiasi utilizzo o affidamento delle informazioni fornite in questo documento è a rischio dell'utente. Nessuna raccomandazione riguardo a prodotti o venditori è fatta o dovrebbe essere sottintesa.

Storia delle revisioni (riportata in originale)

Versione	Data:	Note
1.0	19.01.2011	Author: Dr. Hendrik Dettmering, developed upon request of gasq GmbH The copyright was fully transferred to German Testing Board e.V.
1.1	14.06.2015	Review of content and comparison to the German ISTQB® Certified Tester Foundation Level syllabus 2011 V1.0.1 and the ISTQB® Glossary V2.2 Release per GTB working group meeting of 15.03.2015 (Munich).
2.0	31.03.2017	Learning objectives and content on the basis of V.1.1. Release of the (corresponding german) edition per GTB WG meeting of 31.03.2017 (Frankfurt a. M.).
2.0.1 (English Edition)	30.06.2017	Only minor changes in the key terms; Rework according the findings (mainly wording) from international reviewers after 1 st internal alpha review in March 2017 (see acknowledgement). Corresponding English References inserted (see references)
2.0.1 (English Edition)	13.08.2017	Fine tuning of the terms after discussion with ISTQB® WG Glossary; Rework according the findings (mainly wording) from international reviewers after 2 nd internal alpha review in July 2017 (see acknowledgement).
2.0.1 (English Edition)	20.08.2017	Fine tuning of the term after another discussion with ISTQB® WG Glossary; findings from late Reviewer integrated;
2.0.1 (English Edition)	15.09.2017	Findings from Reviewer by GTB working group meeting (Munich) integrated.
2.0.1 (English Edition)	16.09.2017	Rework of Chapter 3.2.2
2.0.1 (English Edition)	22.09.2017	Final Edits for GA BETA DRAFT Edition
2.0.2 (English Edition)	28.05.2018	Final Edits from BETA Review and for GA Release
2.0.2 (English Edition)	04.07.2018	Watermark removed and trademark restriction added after GA approval and for ISTQB® publication

Sommario

Storia delle revisioni (riportata in originale)	3
Ringraziamenti.....	6
Storia di questo documento.....	7
Scopo del documento.....	8
ISTQB® CT-Specialist: Automotive Software Tester	8
Valore per il Business.....	8
Obiettivi di apprendimento/Livelli cognitivi di conoscenza	9
Termini.....	9
L'esame Automotive Software Tester	9
Requisiti di ammissione per l'Esame.....	9
Accreditamento dei Corsi	10
Livello di dettaglio del Syllabus	10
Come è organizzato questo Syllabus.....	10
Terminologia di genere neutro	11
1. Introduzione (K2) [30 Min]	12
1.1 Requisiti derivanti da obiettivi di progetto divergenti e crescente complessità del prodotto (K2) [15 Min].....	12
1.2 Aspetti del progetto influenzati dagli standard (K1) [5 Min].....	13
1.3 Le sei fasi generiche nel ciclo di vita del sistema (K1) [5 Min]	13
1.4 Il/la contributo/partecipazione del tester nel processo di rilascio (K1) [5 Min].....	14
2 Standard per il testing di sistemi E/E (K3) [300 Min].....	15
2.1 Automotive SPICE (ASPICE) (K3) [140 Min].....	16
2.1.1 Progetto e struttura degli standard (K2) [25 Min].....	16
2.1.2 Requisiti dello standard (K3) [115 Min].....	18
2.2 ISO 26262 (K3) [125 Min].....	21
2.2.1 Sicurezza funzionale e cultura della sicurezza (K2) [20 Min]	21
2.2.2 Integrazione del tester nel ciclo di vita della sicurezza (K2) [15 min]	21
2.2.3 Struttura e parti di test specifico dello standard (K1) [10 Min]	22
2.2.4 L'influenza della criticità sull'estensione del test (K2) [20 Min].....	23
2.2.5 Applicazione del contenuto dal CTFL® nel contesto della ISO 26262 (K3) [60 Min]	24
2.3 AUTOSAR (K1) [15 Min].....	25
2.3.1 Obiettivi della AUTOSAR (K1) [5 Min]	25
2.3.2 Struttura generale della AUTOSAR (K1) [informativo] [5 Min].....	25
2.3.3 Influenza dell'AUTOSAR sul lavoro del tester (K1) [5 Min].....	26

2.4	Comparazione (K2) [20 Min].....	26
2.4.1	Obiettivi di ASPICE e ISO 26262 (K1) [5 Min].....	26
2.4.2	Comparazione dei livelli di test (K2) [15 Min].....	27
3	Testing in un ambiente virtuale (K3) [160 Min].....	28
3.1	Ambiente di test in generale (K2) [30 Min]	28
3.1.1	Motivazione per un ambiente di test nello sviluppo automotive (K1) [5 Min].....	28
3.1.2	Parti generali di un ambiente di test (K1) [5 Min].....	29
3.1.3	Differenze tra Closed-Loop e Open-Loop (K2) [15 Min]	29
3.1.4	Interfacce essenziali, database e protocolli di comunicazione di un'unità di controllo elettronico (K1) [5 Min].....	30
3.2	Testing in ambienti di test XiL (K3) [130 Min].....	31
3.2.1	Model in the Loop (MiL) (K2) [20 Min].....	31
3.2.2	Software in the Loop (SiL) (K1) [10 Min].....	32
3.2.3	Hardware in the Loop (HiL) (K2) [20 Min]	32
3.2.4	Confronto degli ambienti di test XiL (K3) [80 Min]	33
4	Tecniche di test statiche e dinamiche specifiche dell' automotive [230 Min]	36
4.1	Tecniche di test statico (K3) [75 Min]	36
4.1.1	Le linee guida MISRA-C: 2012 (K2) [15 Min].....	36
4.1.2	Caratteristiche di qualità per la revisione dei requisiti (K3) [60 Min].....	37
4.2	Tecniche di test dinamico (K3) [155 Min]	38
4.2.1	Testing delle condizioni, testing delle condizioni multiple, testing delle decisioni/condizioni modificate (K3) [60 Min]	38
	Risultato della decisione per l'espressione:	39
4.2.2	Testing Back-to-Back (K2) [15 Min]	39
4.2.3	Fault injection testing (K2) [15 Min].....	39
4.2.4	Testing basato sui requisiti (K1) [5 Min].....	40
4.2.5	Selezione delle tecniche di test dipendente dal contesto (K3) [60 Min]	40
5	Allegato.....	42
5.1	Elenco delle tabelle.....	43
5.2	Riferimenti.....	43
5.3	Definizioni	48
5.4	Abbreviazioni	55
5.5	Indice Analitico.....	57

Ringraziamenti

Il German Testing Board (GTB), vorrebbe ringraziare il team di autori e revisori della versione tedesca 2017, V2.0 (in ordine alfabetico):

Graham Bath, André Baumann, Arne Becher, Ralf Bongard (Lead Syllabus e Co-Chair WG), Kai Borgeest, Tim Burdach, Mirko Conrad, Klaudia Dussa-Zieger, Matthias Friedrich, Dirk Gebrath, Thorsten Geiselhart, Matthias Hamburg, Uwe Hehn, Olaf Janßen, Jacques Kamga, Horst Pohlmann (Lead Exam e Chair WG), Ralf Reißing, Karsten Richter, Ina Schieferdecker, Alexander Schulz, Stefan Stefan, Stephanie Ulrich, Jork Warnecke e Stephan Weißleder.

Il German Testing Board (GTB) e il WG Automotive Software Tester vorrebbero ringraziare l'esteso team di revisori della versione inglese 2018 (V.2.0.x): Graham Bath, Thomas Borchsenius, Ádám Bíró, Zsolt Csátári, Attila Farkas, Attila Fekete, Ferenc Hamori, Ádám Jezsoviczki, Gábor Kapros, Miguel Mancilla, Roland Milos, Kenji Onishii, Miroslaw Panek, Mirosław Panek, Barthomiej Predki, Stefan Stefan, Stuart Reid, Ralf Reissing, Hidetoshi Suhara, Tamás Széplakin, Eshraka Zakaria e Csaba Zelei.

Per la traduzione italiana si ringraziano: Marco Sogliani, Alessandro Collino, Salvatore Reale, Raffaele Bozzini, Giorgio Pisani

Storia di questo documento

Il syllabus 1.0 è stato sviluppato dal Dr. Hendrik Dettmering nel 2010/2011 su richiesta della Global Association for Software Quality AISBL (gasq).

Per la revisione del documento sono stati incaricati esperti selezionati da Original Equipment Manufacturers, da cui la qualità e l'obiettivo del syllabus sono stati controllati e valutati come adatti. Pertanto, questo documento costituisce il syllabus per la certificazione dell'automotive software tester ed è allo stesso tempo la base per il materiale di formazione così come per le domande dell'esame per la certificazione.

A partire dal 1 gennaio 2014, il gruppo di lavoro "Certified Automotive Software Tester" del German Testing Board (GTB) ha preso in carico un ulteriore aggiornamento del syllabus per permettere il rapido sviluppo dell'argomento e per soddisfare i requisiti del settore di non disporre solamente del core syllabus indipendente dall'ambito di applicazione, ma anche di aspetti specifici automotive disponibili come specializzazione del ben consolidato ISTQB® Foundation Level.

La versione 1.1 è stata rilasciata il 15 giugno 2015. L'edizione era compatibile con la versione 1.0; di conseguenza le parti ridondanti del syllabus ISTQB® Foundation Level sono state rimosse nella Versione 1.1.

Scopo del documento

Questo syllabus definisce una specializzazione a livello Foundation dei programmi di formazione per software testing dell'International Software Testing Qualifications Board (di seguito abbreviato ISTQB®). Con l'aiuto del syllabus alla mano, i training provider creano il materiale del loro corso e definiscono una metodologia di formazione adatta per l'accreditamento. I corsisti si preparano per l'esame con l'aiuto del syllabus.

Ulteriori informazioni riguardo la storia e i retroscena del syllabus possono essere trovate nella storia di questo documento.

Questo Syllabus è la base per l'International Software Testing Qualification di Livello Foundation (Foundation Level) per Automotive Software Tester.

L'ISTQB® mette a disposizione questo syllabus con le seguenti modalità:

- Ai Membri del Board, per consentire la traduzione nella loro lingua madre e per accreditare i fornitori della formazione. I Board nazionali possono adattare il syllabus alle loro particolari esigenze linguistiche e modificare i riferimenti bibliografici per adattarli alle loro pubblicazioni locali.
- Agli Enti Esaminatori, per consentire la traduzione delle domande d'esame nella loro lingua adattandole agli obiettivi di apprendimento di ogni modulo.
- Ai Fornitori della formazione, per consentire la produzione di materiale didattico e mettere a punto metodologie d'insegnamento appropriate.
- Ai Candidati alla certificazione, per consentire la preparazione all'esame (partecipando a un corso o in modo indipendente).
- Alla comunità che si occupa d'ingegneria dei sistemi e del software, per consentire di migliorare la professione di tester del software e dei sistemi e per fornire una base di riferimento per libri e articoli.

L'ISTQB® permette inoltre ad altre entità di usare questo syllabus per altre finalità, purché ne chiedano ed ottengano anticipatamente un permesso scritto.

ISTQB® CT-Specialist: Automotive Software Tester

Il presente modulo Specialist dei programmi di formazione Certified Tester è diretto a tutte le persone coinvolte nell'argomento del testing del software nell'area automotive. Ciò include persone in ruoli come tester, test analyst, test engineer, test consultant, test manager, release tester e software developer. Il livello base si rivolge anche alle persone nei ruoli di project manager, quality manager, software development manager, system analyst (business analyst), IT manager o management consultant, che desiderano acquisire conoscenza e comprendere alla base l'argomento testing del software nell'area automotive.

Valore per il Business

In questo paragrafo, delineremo il valore per il Business (Business Outcome per ISTQB®) che ci si può aspettare da un candidato con una certificazione come CTFL® Automotive Software Tester.

Un CTFL® Automotive Software Tester (CTFL®-AuT) può ...

AUTFL-BO-01 *Collaborare efficacemente in un team di test. („collaborate“)*

- AUTFL-BO-02 *Adattare* le tecniche di test conosciute dall'ISTQB® Certified Tester Foundation level (CTFL®) agli specifici requisiti di progetto. („*adapt*“)
- AUTFL-BO-03 Considerare i requisiti base degli standard attinenti (Automotive SPICE®, ISO 26262, etc.) per la *selezione* di tecniche di test adatte. („*select*“)
- AUTFL-BO-04 *Supportare* il team di test nella pianificazione orientata al rischio delle attività di test e applicare gli elementi conosciuti di strutturazione e prioritizzazione. („*support & apply*“)
- AUTFL-BO-05 *Applicare* i metodi di test virtuali (es. HiL, SiL, MiL, ecc.) negli ambienti di test. („*apply*“)

Obiettivi di apprendimento/Livelli cognitivi di conoscenza

Ogni paragrafo di questo syllabus è assegnato ad un livello cognitivo:

- K1: ricordare
- K2: comprendere
- K3: applicare
- K4: analizzare

Gli obiettivi di apprendimento definiscono ciò che i corsisti avrebbero dovuto imparare dopo aver completato il corrispondente paragrafo/capitolo/modulo.

Il contenuto degli obiettivi di apprendimento contrassegnati come [informativo] devono essere insegnati dal training provider entro un lasso di tempo adeguato, tuttavia, essi NON sono rilevanti per l'esame.

Esempio: AUTFL-2.2.3.1 ricorda il design e la struttura della ISO 26262. [informativo]

Termini

Il corsista dovrebbe essere in grado di riprodurre tutti i termini menzionati nel paragrafo direttamente sotto il titolo "Termini" (K1), anche se non è esplicitamente menzionato negli obiettivi di apprendimento. Si applicano le definizioni del Glossario ISTQB® e le traduzioni nazionali nelle versioni approvate (incl. i termini aggiuntivi dal presente syllabus).

L'esame Automotive Software Tester

Basato su questo syllabus c'è un esame addizionale per il certificato specifico di dominio Foundation Level Specialist Automotive Software Tester. Una domanda dell'esame può chiedere argomenti contenuti in diversi capitoli del syllabus. In genere, ogni domanda dell'esame è assegnata ad un obiettivo di apprendimento, eccetto per quelle domande che sono assegnate ad un termine chiave. Il formato dell'esame è a risposta multipla. Gli esami possono essere sostenuti direttamente dopo un corso di formazione accreditato o indipendentemente (es. in un centro esami o come esame disponibile pubblicamente). Prendere parte ad un corso non è un requisito per sostenere l'esame.

Requisiti di ammissione per l'Esame

Per sostenere l'esame per Certified Automotive Software Tester i candidati devono possedere il certificato ISTQB® Certified Tester Foundation Level (CTFL®) ed avere interesse nel testing dei progetti di sviluppo automotive.

Tuttavia, è consigliato che il candidato

- abbia almeno una conoscenza minima di base nello sviluppo o nel testing del software (ad esempio un'esperienza di sei mesi da system o acceptance tester o da developer)
- o abbia seguito un corso, che sia accreditato secondo lo standard ISTQB® (da un ISTQB®-member-board) e/o
- abbia acquisito un'esperienza iniziale del testing nei progetti di sviluppo E/E nel settore Automotive.

Accreditamento dei Corsi

Un ISTQB® Member Board può accreditare i training provider il cui materiale del corso segue questo syllabus. I training provider dovrebbero ottenere linee guida per l'accREDITAMENTO dal board o dall'organismo che esegue l'accREDITAMENTO. Un corso accREDITATO è riconosciuto conforme a questo syllabus e può includere un esame come parte separata.

Ulteriori riferimenti per i training provider possono essere trovati nell'allegato.

Livello di dettaglio del Syllabus

Il livello di dettaglio consente una formazione ed un'esaminazione coerenti. Per raggiungere questo obiettivo, questo syllabus contiene i seguenti:

- obiettivi di apprendimento generali, i quali descrivono l'intenzione degli (estesi) livelli base
- contenuto che deve essere studiato, includendo una descrizione e, se necessario, riferimenti ad ulteriore letteratura
- obiettivi di apprendimento per ogni area di conoscenza, i quali descrivono il risultato cognitivo obiettivo della formazione e l'approccio mentale del partecipante che devono essere raggiunti
- una lista di termini che il partecipante dovrebbe essere in grado di riprodurre e comprendere
- una descrizione dei concetti importanti da studiare, includendo fonti come letteratura tecnica ben consolidata, normative

Il contenuto del Syllabus non è una descrizione dell'intera area di conoscenza "Testing per sistemi orientati ai software nei progetti di sviluppo elettronico automotive". Esso riflette semplicemente l'ambito e il livello di dettaglio necessari che sono rilevanti per gli obiettivi di apprendimento.

Come è organizzato questo Syllabus

Il syllabus consiste in quattro capitoli principali. Ogni titolo di un capitolo mostra la categoria più impegnativa degli obiettivi di apprendimento / il livello cognitivo più alto, che deve essere coperto dal rispettivo capitolo e definisce il tempo di formazione, il quale va considerato come minimo per quel capitolo in un corso accreditato.

Esempio:

Introduzione (K2) - [30 minuti]

L'esempio mostra che per il capitolo "Introduzione (K2)" K1¹ e K2 sono previsti (ma non K3) e 30 minuti sono pianificati per la formazione del materiale di questo capitolo.

¹ Un obiettivo di apprendimento di un livello più alto di tassonomia implica gli obiettivi di apprendimento dei livelli più bassi.

Ogni capitolo contiene diversi sottocapitoli. Ogni sottocapitolo può anch'esso definire obiettivi di apprendimento e un arco temporale. Se non è dato nessun tempo per un sottocapitolo, è già incluso nel capitolo principale.

Terminologia di genere neutro

Per motivi di semplificazione della leggibilità ci asterremo dalla differenziazione neutra dal punto di vista del genere, ad es. utenti maschi e femmine. Seguendo un approccio di uguaglianza, tutti i nomi di ruolo sono generalmente da considerare validi per entrambi i generi.

1. Introduzione (K2) [30 Min]

Termini

Nessun termine specifico per il test

Obiettivi di apprendimento

- AUTFL-1.1.1 Spiegare e fornire esempi delle sfide dello sviluppo di prodotti automotive che derivano da obiettivi di progetto divergenti e crescente complessità del prodotto (K2)
- AUTFL-1.2.1 Richiamare gli aspetti del progetto, come tempi, costi, qualità e rischi progetto /prodotto, che sono influenzati dagli standard.. (K1)
- AUTFL-1.3.1 Richiamare le sei fasi generiche nel ciclo di vita del sistema secondo la ISO/IEC 24748-1 [1]. (K1)
- AUTFL-1.4.1 Richiamare il contributo e la collaborazione del tester nel processo di rilascio. (K1)

Introduzione

Uno dei sette principi del testing del software è “Il testing è dipendente dal contesto” [21]. Questo paragrafo delinea l’ambiente dello sviluppo E/E, in cui agisce un “Automotive Software Tester”². Da un lato, obiettivi divergenti, complessità crescente e alta spinta per l’innovazione portano a sfide speciali. Dall’altro, gli standard e il ciclo di vita dei veicoli formano l’ambito operativo nel quale il tester lavora. Alla fine, il tester sta contribuendo con il suo lavoro al rilascio di software e sistemi.

1.1 Requisiti derivanti da obiettivi di progetto divergenti e crescente complessità del prodotto (K2) [15 Min]

I produttori e fornitori di autoveicoli continuano a lanciare nuovi modelli di autoveicoli³ più frequentemente che in passato e sotto la crescente pressione dei costi. I seguenti aspetti influenzano questo processo:

- Numero crescente di modelli e complessità:
Per essere in grado di soddisfare meglio le esigenze individuali e dei clienti, OEMs (Produttori di autoveicoli) offrono sempre più modelli di autoveicoli. Tuttavia, questo riduce le quantità per modello. Per coprire il risultante aumento dei costi di sviluppo e produzione, i produttori sviluppano molti modelli come varianti di una piattaforma comune. Lo sviluppo di una piattaforma, tuttavia, è di gran lunga più complicato dello sviluppo di un singolo modello a causa della necessità di mantenere il controllo sulle molte possibili variazioni.
- Aumento della gamma di funzionalità:
Il cliente finale richiede sempre più innovazioni senza omettere le funzioni esistenti, il che fa aumentare la gamma di funzioni.
- Aumento del numero di configurazioni:
Il cliente finale vuole adattare il suo modello di auto ai suoi desideri individuali. Ciò richiede molte configurazioni possibili per un modello di auto, anche nella gamma della funzionalità.

² In seguito useremo solo il termine „Tester”. Deve essere inteso come la forma abbreviata di “Automotive E/E Software Tester”

³ Esempio da uno studio dei consulenti di management Progenium: “Nel 1990, solo 101 modelli differenti di autoveicoli erano sul mercato..., nel 2014, questo numero era aumentato a 453” [43]

- Aumento dei requisiti di qualità:
Nonostante l'aumento dei livelli di funzionalità e di complessità, il cliente finale si aspetta almeno la stessa o addirittura una qualità superiore del veicolo e delle sue funzioni.

Poiché gli obiettivi del progetto tempo, costi e qualità sono in competizione („Project management triangle“) i produttori di autoveicoli (OEMs) e i fornitori devono battersi per uno sviluppo del sistema più efficiente, che consenta tempi di sviluppo più brevi nonostante l'aumento della complessità, l'aumento dei requisiti di qualità e budget più ridotti.

1.2 Aspetti del progetto influenzati dagli standard (K1) [5 Min]

Gli standard hanno un'influenza su aspetti principali del progetto come tempo, costi, qualità e rischi di progetto e di prodotto:

- Gli standard aumentano l'efficienza dei processi (ad esempio per ridurre il tempo o il costo di sviluppo ad una qualità stabile) attraverso:
 - terminologia uniforme
 - migliore trasparenza
 - collaborazione più facile (interna ed esterna)
 - maggiore riutilizzabilità
 - esperienza consolidata („Best Practice“)
- Con linee guida ben consolidate [21], aiutano a scoprire presto rischi e difetti e a risolverli.
- Gli standard sono la base per le revisioni. Di conseguenza, un revisore può valutare la qualità di un prodotto o di un processo. Allo stesso tempo, il revisore può controllare se questi soddisfano i requisiti [1].
- Gli standard fanno parte delle disposizioni e linee guida contrattuali o regolatorie.

Questo syllabus esaminerà, tra gli altri, i seguenti standard:

- standard, come la ISO 26262 [3] o la Automotive SPICE(ASPICE) [2], che standardizzano processi e metodi.
- standard, come la AUTOSAR [3], che standardizza i prodotti.

1.3 Le sei fasi generiche nel ciclo di vita del sistema (K1) [5 Min]

Il ciclo di vita di un autoveicolo e di tutti i componenti inclusi⁴ inizia con l'idea del prodotto e termina con il ritiro. Durante questo ciclo di vita sono coinvolti processi di sviluppo, processi di business, processi logistici e processi riguardanti la tecnologia di produzione. Milestone con criteri di entrata e uscita precedentemente definiti aiutano a raggiungere processi maturi. Questi separano e sincronizzano il ciclo di vita del sistema⁵ nelle seguenti sei fasi [1]. (attività di test tipiche⁶ tra parentesi):

- idea (pianificazione del test)
- sviluppo (analisi, progettazione, implementazione, esecuzione, valutazione e report del test)
- produzione (end of line test)
- utilizzo (nessuna attività di test)
- supporto (test di manutenzione)

⁴ Unità di controllo elettronico (hardware e software) così come i componenti.

⁵ Il Safety lifecycle della ISO 26262 passa attraverso fasi simili.

⁶ Per le attività di test vedere anche: processi di test fondamentali [2].

- ritiro (test di migrazione)

Il processo di sviluppo del prodotto più diffuso nel settore automotive delinea: concetto, sviluppo e produzione.

1.4 Il/la contributo/partecipazione del tester nel processo di rilascio (K1) [5 Min]

Nell'ambiente automotive, un progetto raggiunge una milestone dichiarandone un rilascio e dopo averne verificato le caratteristiche si decide se gli obiettivi sono stati raggiunti. Da questo momento in poi, l'elemento da rilasciare soddisfa il livello di maturità necessario per il suo utilizzo e scopo.

Il processo di rilascio è previsto conduca al rilascio dell'elemento. L'elemento da rilasciare consiste:

- nell'elemento di test (configurazione del software includendo la parametrizzazione, se necessario anche con l'hardware e le meccaniche)
- nella documentazione di supporto aggiuntiva.

Il tester, attraverso il test report finale, consegna importanti informazioni per il processo di rilascio quali: [3]:

- oggetti testati e le caratteristiche delle prestazioni includendo la loro versione
- difetti noti
- metriche del prodotto
- informazioni per la raccomandazione di rilascio (al raggiungimento dei criteri di uscita del test) basate sulla regole di rilascio, ad esempio fornite da linee guida di Best Practice (cioè: test su terreni chiusi o strade pubbliche, raccomandazione sull'installazione)

Inoltre, il tester partecipa alla creazione di ulteriori risultati rilevanti per il rilascio quali [4]:

- mettere in ordine di priorità e partecipare alle decisioni riguardanti i cambiamenti.
- mettere in ordine di priorità le funzionalità (per la sequenza di implementazione).

2 Standard per il testing di sistemi E/E (K3) [300 Min]

Termini

Automotive SPICE (ASPICE)

Automotive SPICE (ASPICE), testing di qualifica del software (ASPICE), testing di qualifica del sistema (ASPICE)

ISO 26262

Automotive Safety Integrity Level (ASIL), sicurezza funzionale, tabella metodologica (ISO 26262),

AUTOSAR

Nessun termine specifico per il test

Confronto

Nessun termine specifico per il test

Obiettivi di apprendimento

Automotive SPICE (ASPICE)

AUTFL-2.1.1.1 Richiamare le due dimensioni dell'Automotive SPICE (ASPICE). (K1)

AUTFL-2.1.1.2 Richiamare le 3 categorie di processo e gli 8 gruppi di processo di ASPICE [informativo]. (K1)

AUTFL-2.1.1.3 Spiegare i livelli Capability da 0 a 3 di ASPICE. (K2)

AUTFL-2.1.2.1 Richiamare lo scopo dei 5 processi di test importanti di ASPICE. (K1)

AUTFL-2.1.2.2 Spiegare il significato dei quattro livelli di classificazione e degli indicatori di capacità di ASPICE dal punto di vista del testing. (K2)

AUTFL-2.1.2.3 Spiegare i requisiti di ASPICE per la strategia di test includendo quella del test di regressione. (K2)

AUTFL-2.1.2.4 Richiamare i requisiti di ASPICE per la documentazione dei test. (K1)

AUTFL-2.1.2.5 Progettare una strategia di verifica (in contrasto con una strategia di test) e criteri per la verifica di unità. (K3)

AUTFL-2.1.2.6 Spiegare i differenti requisiti di tracciabilità di ASPICE dal punto di vista del testing. (K2)

ISO 26262

AUTFL-2.2.1.1 Spiegare l'obiettivo di sicurezza funzionale per i sistemi E/E. (K2)

AUTFL-2.2.1.2 Richiamare il contributo dei tester per la cultura sulla sicurezza. (K1)

AUTFL-2.2.2.1 Presentare il ruolo del tester nel contesto del ciclo di vita della sicurezza secondo la ISO 26262. (K2)

AUTFL-2.2.3.1 Richiamare il progetto e la struttura della ISO 26262. [informativo]⁷

⁷ Non obbligatorio per gli esami
Italiano V2.0.2

- AUTFL-2.2.3.2 Richiamare il nome delle sezioni (titoli delle parti) della ISO 26262 che sono rilevanti per il tester. (K1)
- AUTFL-2.2.4.1 Richiamare i livelli di criticità dell'ASIL. (K1)
- AUTFL-2.2.4.2 Spiegare l'influenza dell'ASIL sulle tecniche di progettazione dei test e sui tipi di test applicabili a test statici e dinamici e la risultante estensione del test. (K2)
- AUTFL-2.2.5 Essere in grado di interpretare le tabelle metodologiche della ISO 26262. (K3)

AUTOSAR

- AUTFL-2.3.1 Richiamare gli obiettivi della AUTOSAR. (K1)
- AUTFL-2.3.2 Richiamare il progetto generale della AUTOSAR [informativo]⁸. (K1)
- AUTFL-2.3.3 Richiamare l'influenza della AUTOSAR sul lavoro del tester. (K1)

Confronto

- AUTFL-2.4.1 Richiamare i differenti obiettivi di ASPICE e ISO 26262 (K1).
- AUTFL-2.4.2 Spiegare le differenze tra ASPICE e ISO 26262 e CTFL® riguardo i livelli di test (K2).

2.1 Automotive SPICE (ASPICE) (K3) [140 Min]

Introduzione

Il miglioramento del processo segue l'approccio secondo cui la qualità di un sistema dipende dalla qualità del processo di sviluppo. I modelli di processo in questo caso offrono un'opzione per i miglioramenti, misurando la capacità di processo di un'organizzazione rispetto al modello. Inoltre, il modello serve come struttura operativa per il miglioramento dei processi di un'organizzazione utilizzando i risultati della valutazione [5].

Dal 2001 in poi, la SPICE⁹ User Group e la AUTOSIG (Automotive Special Interest Group) ha sviluppato la Automotive SPICE (ASPICE). Dalla sua pubblicazione nel 2005, lo standard si è ben affermato nel settore automotive.

Nel luglio 2015, la German Association dell'Automotive Industry (VDA) ha rilasciato ASPICE Versione 3.0 [9]. Dal 2017 in poi la versione migliorata V.3.1. di ASPICE 3.1 sostituirà [6] la precedente Versione 2.5 [2]. Tutte le affermazioni riportate in questo paragrafo di conseguenza si riferiscono alla Versione 3.1 di ASPICE [47].

2.1.1 Progetto e struttura degli standard (K2) [25 Min]

2.1.1.1 Le due dimensioni di ASPICE

ASPICE definisce un modello di valutazione con due dimensioni:

Nella **dimensione del processo**, ASPICE definisce il modello di riferimento del processo. Questo serve come riferimento per confrontare i processi delle organizzazioni così che possano essere valutati e migliorati. Per ogni processo, ASPICE ne definisce lo scopo e i risultati così come le azioni richieste (pratiche di base) e i risultati del lavoro (prodotti di lavoro). Se un'organizzazione necessita di ulteriori

⁸ Non obbligatorio per gli esami.

⁹ Acronimo per „Software Process Improvement and Capability Determination“

processi di riferimento oltre ad ASPICE, questi possono essere presi per esempio dalla ISO/IEC 12207 [10] o dalla ISO/IEC 15288 [11].

Nella **dimensione delle capacità** ASPICE definisce un numero di attributi di processo. Questi forniscono le caratteristiche misurabili delle capacità di processo. Per ogni processo, ci sono attributi di processo sia specifici che generici. La ISO/IEC 33020 serve come base per la valutazione delle capacità di processo [39].

Con l'aiuto di questo modello è possibile valutare i processi (dimensione del processo) per quanto riguarda le capacità (dimensione delle capacità).

2.1.1.2 Categorie di processo nella dimensione di processo

ASPICE raggruppa i processi in 8 gruppi di processo, poi i gruppi di processo in 3 categorie di processo [9][47]:

I **processi primari** includono tutti i processi che servono come processi chiave dell'azienda:

- Acquisition (ACQ) di prodotti e/o servizi
- Supply (SPL) di prodotti e/o servizi
- System engineering (SYS)
- Software engineering (SWE)

I **processi di supporto** includono tutti i processi che supportano gli altri processi:

- Supporting processes (SUP)

I **processi organizzativi** includono tutti i processi che supportano gli obiettivi dell'azienda:

- Management (MAN) di un progetto o di un processo
- Process improvement (PIM)
- Reuse (REU) di sistemi e componenti

Per il tester, i gruppi di processo system development (SYS) e software development (SWE) sono di particolare interesse. Essi costruiscono i processi dell'Automotive SPICE V-model ([9] Allegato D "Concetti Chiave").

2.1.1.3 Livelli di capacità nella dimensione delle capacità

Il valutatore valuta la capacità di processo con l'aiuto di un sistema di valutazione a sei livelli (display di livelli). ASPICE definisce i livelli di capacità da 0 a 3¹⁰ come segue [9][47]:

- **Livello 0** (processo incompleto): Il processo non esiste o non raggiunge lo scopo del processo. Esempio: Il tester esamina solamente una parte minore dei requisiti.
- **Livello 1** (processo eseguito): Il processo implementato raggiunge il suo scopo di processo (ma forse eseguito in maniera incoerente). Esempio: Non è disponibile una pianificazione completa per il processo di test. Tuttavia, il tester può dimostrare il livello di adempimento dei requisiti.
- **Livello 2** (processo gestito): Il progetto pianifica e supervisiona il processo nella sua esecuzione. In determinate circostanze, adatta il corso dell'azione durante l'esecuzione per raggiungere l'obiettivo. I requisiti per i prodotti di lavoro sono definiti. Un membro del progetto controlla i prodotti di lavoro e li approva. Esempio: Il test manager definisce gli obiettivi di test, pianifica le attività di test e supervisiona il processo. In caso di deviazioni, reagisce di conseguenza.

¹⁰ I livelli di capacità 4 e 5 non sono attualmente al centro del settore automotive.

- **Livello 3** (processo stabilito): Il progetto usa un processo standardizzato, e i risultati sono utilizzati per migliorare costantemente. Esempio: Esiste una strategia di test generale per l'intera organizzazione. Dopo il completamento del test (vedere i processi di test fondamentali) il test manager aiuta a svilupparlo ulteriormente.

2.1.2 Requisiti dello standard (K3) [115 Min]

2.1.2.1 Processi di test specifici

ASPICE definisce i processi di test in base a tutti i processi di sviluppo del software e di sistema [8]:

- Il processo di **verifica dell'unità** del software (SWE.4) richiede un testing sia statico che dinamico. Valuta i componenti del software in base al suo progetto dettagliato (SWE.3).
- Il testing di **integrazione del software** (SWE.5) valuta il software integrato in base all'architettura del software (SWE.2).
- Il testing di **qualifica del software** (SWE.6) valuta il software integrato in base ai requisiti del software (SWE.1).
- Il testing di **integrazione di sistema** (SYS.4) valuta il sistema integrato in base all'architettura del sistema (SYS.3).
- Il testing di **qualifica di sistema** (SYS.5) valuta il sistema integrato in base ai requisiti del sistema (SYS.2).

2.1.2.2 Livelli di valutazione e indicatori di capacità

Un valutatore può analizzare le capacità di un processo attraverso indicatori di capacità. ASPICE li definisce con 9 attributi di processo (PA). I livelli di capacità da 1 a 3, sono definiti come segue (usando l'esempio SWE.6 tra parentesi) [9], [47]:

- PA 1.1: **Performance del processo** (il tester si orienta da solo mediante il processo di test fondamentale).
- PA 2.1: **Gestione della performance** (il tester pianifica, supervisiona e controlla, tra le altre cose, le attività di test).
- PA 2.2: **Gestione del prodotto di lavoro** (il tester controlla, tra le altre cose, la qualità della documentazione di test).
- PA 3.1: **Definizione del processo** (la persona responsabile per il processo di test definisce, fra le altre cose, una strategia di progetto generale).
- PA 3.2: **Implementazione del processo** (Il tester applica la strategia di test definita nel PA 3.1).

Per l'esecuzione del processo (PA 1.1) ASPICE definisce due tipi di indicatori: pratiche di base (BP) e prodotti di lavoro (WP). In aggiunta sono definite pratiche generiche (GP) e risorse. La valutazione degli attributi di processo è basata sul livello di implementazione degli indicatori con quattro livelli di classificazione [9], [47]:

- **N** (None): non soddisfatto (0% fino a \leq 15%)
- **P** (Partly): parzialmente soddisfatto ($>$ 15% fino a \leq 50%)
- **L** (Largely): largamente soddisfatto ($>$ 50% fino a \leq 85%)
- **F** (Fully): pienamente soddisfatto ($>$ 85% fino a \leq 100%)

Affinché un processo raggiunga determinati livelli di capacità, gli indicatori del livello di capacità da raggiungere devono essere "largamente soddisfatto (L)". Gli indicatori dei livelli di capacità inferiori devono essere "pienamente soddisfatto (F)".

2.1.2.3 Strategia di test e strategia del test di regressione

Come pratica base, ASPICE richiede una strategia di test¹¹ per ogni processo di test specifico (vedere 2.1.2.1). Il test manager la sviluppa durante la pianificazione del test. Linee guida dei test, obiettivi di progetto così come requisiti contrattuali e normativi, ne costruiscono la base.

Il tester riconosce, come principio del testing, le verifiche dalle fasi iniziali. Ciò si applica anche al testing del software nell'ambiente automotive. Tuttavia, qui entra in gioco un altro aspetto perché gli ambienti a livelli di test più elevati sono significativamente più costosi. Ad esempio, per il testing a livelli più alti è necessario un hardware appositamente sviluppato ed incorporato (ad esempio come un prototipo o un modello unico). La strategia di test definisce gli ambienti di test specifici per livello, ma anche quali test e in quali ambienti è richiesto siano eseguiti dal tester.

La strategia del test di regressione è una parte essenziale della strategia di test. La sfida qui, sta nella scelta economicamente sensata dei casi di test ("valore aggiunto del testing"). La strategia di regressione definisce l'obiettivo e la tecnica per la scelta dei test di regressione. Ad esempio, la scelta può essere basata sul rischio. Un'analisi d'impatto aiuta a identificare le aree sulle quali il tester si deve concentrare con i test di regressione. Tuttavia, il test manager può anche chiedere al tester di ripetere tutti i casi di test automatizzati per ogni rilascio.

2.1.2.4 Documentazione del test in ASPICE

Per la documentazione delle attività di test, ASPICE richiede molti prodotti di lavoro (WP) che sono noti al tester dal CTFL® [9]:

- WP 08-50: Specifica di test (contenente il progetto di test, il caso di test e la specifica della procedura di test)
- WP 08-52: Piano di test secondo ISO/IEC/IEEE 29119-3 [34] e la strategia inclusa (WP 19-00)
- WP 13-50: Risultato del test, test log, incident/deviation report e test summary report

Per ogni prodotto di lavoro, ASPICE definisce esempi di caratteristiche e contenuto. Un valutatore può valutarli tramite controlli a campione. Per un valutatore servono come indicatore oggettivo per l'esecuzione di un processo.

Per il test plan, ASPICE fa riferimento direttamente a ISO/IEC/IEEE 29119-3¹². Questo standard fornisce anche modelli che possono essere usati per altri prodotti di lavoro richiesti e possono essere adattati per uno scopo particolare. Deve essere garantito che nel contesto contribuisca allo scopo dei processi previsto.

2.1.2.5 Strategia di verifica e criteri per la verifica delle unità (SWE.4)

Per la verifica delle unità del software (SWE.4) ASPICE richiede una strategia di verifica¹³. Nel caso dei processi di test specifici di SWE.5/SWE.6/SYS.4/SYS.5, ASPICE richiede una strategia di test (vedere 2.1.2.3). La strategia di test riguarda "solo" i test dinamici. Questa è un'aggiunta alla strategia di verifica, la quale considera anche la revisione del codice e le analisi statiche (Entrambe le tecniche sono note come "test statici" dal CTFL®).

Il tester verifica la conformità con il progetto dettagliato del software e con i requisiti funzionali e non funzionali secondo la strategia di verifica. La strategia definisce in che modo il tester fornisce le

¹¹ Secondo il CTFL [2] la strategia di test specifica del progetto è conosciuta anche come tecnica di test.

¹² Questa rimpiazza la IEEE 829:1998 e la IEEE 829:2008 che sono ancora utilizzate nei syllabi ISTQB.

¹³ Per i termini „strategia di verifica“ e „strategia di test“, in ASPICE il termine „strategia“ è usato in opposizione a „tecnica“ specifica di progetto in ISTQB.

evidenze. Di conseguenza, il tester può usare combinazioni differenti di tecniche di test statico e dinamico per verificare le unità.

Se uno sviluppatore modifica un'unità software, il tester deve valutare anche questo cambio. Quindi, la strategia per la verifica delle unità include anche una strategia di regressione. Ciò include la verifica del codice cambiato, il testing confermativo così come la verifica ripetuta delle parti non cambiate (test di regressione statici e dinamici).

Nel SWE.4.BP.2 ASPICE richiede lo sviluppo di criteri per la verifica delle unità. Questi criteri definiscono ciò che necessita essere soddisfatto. Perciò, un tester può valutare quanto un'unità soddisfi i requisiti non funzionali e corrisponda al progetto dettagliato. I seguenti criteri sono criteri possibili per la verifica delle unità:

- Test case di unità (inclusi i dati di test)
- Obiettivi per la copertura del test (ad esempio copertura delle decisioni)
- Analisi statica supportata dagli strumenti, la quale valuta la conformità con gli standard di codifica (come MISRA-C, vedere 4.1.1)
- Revisioni del codice per unità o parti di unità, le quali non possono essere valutate da analisi statiche supportate dagli strumenti.

Secondo la Automotive SPICE (ASPICE), la documentazione della strategia di verifica fa parte del piano di test ([13] paragrafo 6.2.7) sul livello dell'unità. Il contenuto è diviso secondo ISO/IEC/IEEE 29119-3 e migliorato dagli aspetti dei test statici.

2.1.2.6 Tracciabilità nell'Automotive SPICE (ASPICE)

Come nel CTFL® Core Syllabus [21], ASPICE richiede anche una tracciabilità bidirezionale¹⁴. Questo permette al tester:

- di analizzare l'impatto
- di valutare la copertura o
- di tenere traccia dello status.

Inoltre, questo consente al tester di assicurare la coerenza tra gli elementi collegati, sia testualmente che semanticamente.

ASPICE differenzia tra tracciabilità verticale e orizzontale [9]:

Verticalmente, ASPICE richiede che i requisiti degli stakeholder siano collegati ai componenti del software. Nel farlo, il collegamento tra tutti i livelli di sviluppo assicura una coerenza tra i prodotti di lavoro correlati.

Orizzontalmente, ASPICE richiede anche tracciabilità e coerenza, in questo caso tra risultati di lavoro dello sviluppo con le corrispondenti specifiche di test e risultati.

In aggiunta, la pratica base SUP.10.BP8 richiede una tracciabilità bidirezionale tra richieste di cambiamento e prodotti di lavoro influenzati da queste ultime. Siccome una richiesta di cambiamento è iniziata da un problema, è richiesta la tracciabilità bidirezionale tra le richieste di cambiamento e il corrispondente rapporto sul problema. A causa del numero occasionalmente elevato di collegamenti, una catena consistente di strumenti può essere utile. Questo permette al tester di creare e gestire efficacemente le dipendenze.

¹⁴ Di seguito, il termine tracciabilità implicherà sempre la tracciabilità bidirezionale.

2.2 ISO 26262 (K3) [125 Min]

2.2.1 Sicurezza funzionale e cultura della sicurezza (K2) [20 Min]

2.2.1.1 Obiettivo della sicurezza funzionale per i sistemi E/E

La complessità tecnica e funzionale dei sistemi integrati è in costante aumento. Allo stesso tempo, potenti sistemi elettrici ed elettronici basati su software permettono nuove complesse funzionalità come l'automazione delle funzioni di guida nell'autoveicolo.

A causa dell'elevata complessità, il rischio che un'azione errata si verifichi durante lo sviluppo è in aumento. La conseguenza può essere un (non rilevato) stato di errore nel sistema. Per i sistemi con un potenziale rischio intrinseco per la vita e per parti del corpo, la persona responsabile per la sicurezza, necessita pertanto di analizzare i potenziali rischi. Se è presente un rischio reale, identifica misure adeguate per mitigare il loro possibile impatto ad un livello di rischio accettabile.

I metodi per l'esecuzione di tali analisi sono riassunti nello standard per la sicurezza funzionale. Lo standard base è lo IEC 61508. La International Organization for Standardization (ISO) ha adattato la ISO 26262 da questo standard.

Secondo la ISO 26262, la Sicurezza Funzionale è definita come assenza di rischi irragionevoli dovuti a pericoli causati da un malfunzionamento dei sistemi E/E. In tal senso, il termine va differenziato da altri termini sulla sicurezza come sicurezza informatica, sicurezza del prodotto o sicurezza del lavoro [ISO 26262] [IEC 61508]. Sicurezza nell'ambiente di lavoro e sicurezza informatica non sono al centro della ISO 26262. La mancanza di sicurezza informatica può danneggiare la Sicurezza Funzionale e la sicurezza informatica contribuisce alla sicurezza del prodotto.

2.2.1.2 Contributo del tester alla cultura della sicurezza

Nello sviluppo del prodotto secondo la ISO 26262 non è abbastanza monitorare i processi della propria organizzazione. Tutti i partecipanti devono assumere un approccio indipendente dal processo. Tutti devono capire il loro impatto sul processo di sviluppo e sulla sicurezza del prodotto finale. Questo include partner esterni e fornitori.

I partecipanti devono comprendere che le proprie azioni non si svolgono indipendentemente da altri processi. Ogni fase dello sviluppo va a costituire un contributo essenziale alla rispondenza di conformità e all'implementazione dei requisiti rilevanti della Sicurezza Funzionale. Questa responsabilità *non* termina con il lancio del prodotto. Tutto questo prosegue fino alla fine del ciclo di vita del sistema.

Il tester contribuisce alla cultura della sicurezza partecipando responsabilmente in tutte le fasi del ciclo di vita di sviluppo del software e svolgendo il proprio lavoro con una visione continua del contesto generale dello sviluppo del prodotto [ISO 26262]

2.2.2 Integrazione del tester nel ciclo di vita della sicurezza (K2) [15 min]

Il ciclo di vita della sicurezza descrive le fasi di uno sviluppo del prodotto orientato alla sicurezza. Inizia con la prima idea del prodotto e la ricerca di possibili rischi. Dopo la specifica dei risultanti requisiti di sicurezza, segue l'implementazione in un prodotto specifico. Il ciclo termina con lo smaltimento del prodotto alla fine della sua vita (vedere anche il capitolo 1.3).

Il ciclo di vita della sicurezza secondo la ISO 26262 passa attraverso le seguenti fasi:

- 1° fase: Concezione del prodotto
- 2° fase: Sviluppo del prodotto

- 3° fase: Produzione e manutenzione del prodotto (dopo il “rilascio per la produzione”)

Il tester presso il fornitore lavora principalmente nelle prime due fasi. Modifiche al prodotto nella terza fase portano a un ritorno alla prima o seconda fase, a seconda della loro estensione. Pertanto, il tester partecipa anche alle modifiche. Basandosi sui requisiti relativi alla sicurezza (vedere il capitolo 2.2.4) progetta i casi di test, seleziona le tecniche di test per la verifica all'interno dello sviluppo del prodotto e la convalida di questi requisiti. Il tester, poi li eseguirà nelle relative sottofasi dello sviluppo del prodotto.

Le attività di pianificazione dei test normalmente si svolgono all'interno della fase di concezione. Rettifiche nei documenti risultanti (ad esempio nel test plan o nelle specifiche del test) possono, tuttavia, essere necessarie in qualsiasi fase. L'esecuzione del test avviene principalmente al trasferimento tra le singole sottofasi dello sviluppo del prodotto. Ad esempio, tra l'implementazione e l'integrazione del software così come successivamente all'integrazione del software con l'hardware. Inoltre, il tester contribuisce significativamente al trasferimento alla terza fase con le sue attività di test [ISO 26262].

2.2.3 Struttura e parti di test specifico dello standard (K1) [10 Min]

2.2.3.1 Progettazione e struttura dello standard [informativo]

La ISO 26262 è costituita da 10 volumi (parti):

- Vocabolario (volume 1),
- Gestione della sicurezza funzionale (volume 2),
- Le fasi del ciclo di vita della sicurezza:
 - Fase di concezione (volume 3)
 - Sviluppo del prodotto per l'intero sistema, hardware e software (volumi 4-6)
 - Produzione e operazione (volume 7)
- Processi di supporto (volume 8)
- ASIL e analisi orientata alla sicurezza (volume 9)
- Linee guida per l'applicazione della ISO 26262 (volume 10).

Eccetto che per il volume 1 ed il volume 10, ogni volume include contenuto normativo. Parte di questo sono:

- Un'introduzione generale,
- L'ambito dell'applicazione,
- Riferimenti normativi e
- Requisiti per la conformità con lo standard.

Questi sono seguiti dagli argomenti specifici del volume corrispondente. La struttura della loro descrizione è la stessa in ogni volume. Le attività da effettuare sono descritte con una struttura simile in tutti i volumi (parti) [ISO 26262]:

- Obiettivo
- Informazioni generali
- Informazioni introduttive
- Prerequisiti
- Ulteriori informazioni di supporto
- Requisiti e raccomandazioni
- Prodotti di lavoro

2.2.3.2 Volumi rilevanti (parti) per il tester

Per il tester del software, la verifica del software e (almeno in parte) anche la validazione del sistema sono di primaria importanza. Oltre al Volume 1 (terminologia), anche molti altri volumi (parti) sono di

particolare interesse: i Volumi 4 e 6 forniscono informazioni dettagliate e requisiti riguardanti le misure raccomandate per la verifica del software. Questo si applica alla selezione, alla progettazione e all'implementazione così come all'esecuzione delle corrispondenti misure di verifica.

In tal modo, questi volumi si concentrano sui test, sugli aspetti di verifica specifici del sistema (Volume 4, compresa la validazione del sistema) e sui livelli del software (Volume 6). Se gli aspetti specifici dell'hardware sono anch'essi rilevanti per il proprio lavoro, il tester li troverà nel Volume 5. Gli aspetti riguardanti l'hardware, ma anche il software, sono considerati nell'ambito dell'interfaccia hardware software (volumi 4, 5 e 6).

Il Volume 8 della ISO 26262 costituisce un caso speciale in quanto descrive le caratteristiche di processo specifiche alla verifica di tutti i livelli di test. In aggiunta, contiene requisiti per importanti processi di supporto al tester, come ad esempio la documentazione e la certificazione degli strumenti. [ISO 26262]

2.2.4 L'influenza della criticità sull'estensione del test (K2) [20 Min]

2.2.4.1 I livelli di criticità di ASIL

L'ASIL ("Automotive Safety Integrity Level") è una misura per la riduzione del rischio ottenuta mediante misure della Sicurezza Funzionale. Tali misure possono ad esempio essere una funzione di sicurezza indipendente per la supervisione di un sistema E/E o per l'implementazione di metodi specificatamente definiti. Per livelli di rischio più elevato, possono essere necessarie misure più elaborate.

All'inizio del progetto, un esperto del team esegue la Hazard analysis e la valutazione di rischio per il prodotto. Per *ogni* rischio identificato da questa analisi, si determina un ASIL con l'aiuto di una delle metodologie definite nello standard. Nel passo successivo, si redigono obiettivi di sicurezza e requisiti di sicurezza. Questi usano lo stesso ASIL del rischio su cui sono basati.

La ISO 26262 definisce quattro livelli: da ASIL A per requisiti di sicurezza inferiori, fino a ASIL D per requisiti di sicurezza elevati.

Se la Hazard Analysis e la Valutazione del rischio portano a requisiti inferiori a ASIL A, secondo lo standard essi *non* sono rilevanti per la sicurezza. Questi requisiti saranno coperti attenendosi all'esistente procedura di quality management (QM). [ISO 26262]

2.2.4.2 Influenza di ASIL su tecniche, tipologie ed estensione dei test

L'ASIL stabilito influenza direttamente l'estensione dei test che vanno implementati dal tester. In base al particolare livello dell'ASIL, la norma ISO 26262 raccomanda l'esecuzione di diverse misure o pacchetti di misure. Così facendo, la regola è che lo standard per ASIL più elevati raccomanda misure più estensive e più dettagliate. Per livelli ASIL più bassi, l'esecuzione delle misure specificate è spesso opzionale.

ISO 26262 specifica tre livelli di raccomandazione: nessuna raccomandazione, raccomandato, e altamente raccomandato. Per "nessuna raccomandazione", lo standard non fornisce nessuna raccomandazione per o contro l'uso della misura corrispondente. Può essere usato come supporto senza alcuna preoccupazione. Tuttavia, la sua esecuzione non sostituisce le misure raccomandate o altamente raccomandate dalla ISO.

Ciò significa che per il tester, lo standard raccomanda tecniche di progettazione specifiche del test e tipi di test per sistemi rilevanti alla sicurezza funzionale a seconda dell'ASIL. Il tester può decidere indipendentemente solo nel contesto dello standard riguardante questo caso speciale. Ad esempio, l'uso della partizione di equivalenza e dell'analisi ai valori limite sono raccomandati per ASIL A. D'altra parte, per un ASIL B o superiore, quelle tecniche sono altamente raccomandate (vedere anche il capitolo 2.2.5).

L'ASIL non è una caratteristica dell'intero prodotto. È connesso ad uno specifico obiettivo di sicurezza e al risultante requisito di sicurezza. Di conseguenza, possono esserci sforzi di test significativamente differenti per i requisiti di sicurezza con differenti ASIL per lo stesso prodotto. Questo deve essere tenuto in considerazione dal tester quando pianifica l'estensione dei test. [ISO 26262]

2.2.5 Applicazione del contenuto dal CTFL® nel contesto della ISO 26262 (K3) [60 Min]

ISO 26262 offre al tester raccomandazioni specifiche per le sue attività di test nella forma di tabelle metodologiche. Queste tabelle possono essere trovate nei volumi (parti) 4, 5, 6 e 8. Oltre alle raccomandazioni specifiche della Sicurezza Funzionale per i processi e le attività, le tabelle includono anche le tecniche che vanno usate dal tester. In questo contesto, lo standard utilizza il termine "metodo" in relazione a tutte le tecniche o le attività applicabili. A questo punto, la terminologia della Sicurezza Funzionale differisce leggermente dai termini di ISTQB®. Per il tester, i seguenti metodi della ISO 26262 sono di particolare interesse:

- Tecniche di progettazione dei test (ad es. partizione di equivalenza, analisi ai valori limite, ...)
- Tecniche di esecuzione del test (ad es. simulazione o prototipo di una parte o dell'intero sistema)
- Tipologie di test (ad es. test non funzionali come il performance test, o il soak test, ..)
- Ambienti di test (ad es. HiL, veicolo, ...)
- Tecniche di test statico (ad es. revisioni, analisi statiche, ...)

La tabella metodologica definisce il metodo raccomandato dallo standard per ogni livello ASIL.

Le tabelle sono sempre progettate con la stessa struttura:

		ASIL A	ASIL B	ASIL C	ASIL D
1	Metodo x	o	+	++	++
2	Metodo y	o	o	+	+
3a	Metodo z1	+	++	++	++
3b	Metodo z2	++	+	o	o

Tabella 1: Esempio di una tabella metodologica

Per ogni metodo, in base al livello ASIL, è indicato se il suo uso è raccomandato (+) o anche altamente raccomandato (++) . Per i metodi indicati come opzionali (o), non ci sono raccomandazioni fornite dallo standard pro o contro i loro usi.

ISO 26262 menziona anche metodi alternativi equivalenti nelle tabelle (nell'esempio qui sopra, le righe 3a e 3b). Qui, il tester necessita di scegliere la combinazione adatta per essere in grado di controllare i requisiti pertinenti in modo conforme all'ASIL. La scelta della combinazione dovrebbe essere spiegata dal tester.

Nel caso di metodi senza alternative (nell'esempio, le righe 1 e 2), questa opzione di scelta non è permessa. In questo caso, il tester deve applicare tutti i metodi che sono altamente raccomandati per il livello ASIL corrispondente.

Dall'esempio sopra, derivano i seguenti metodi per la prova di un requisito secondo ASIL C:

- Metodo x: altamente raccomandato, quindi normamente da applicare se lo sviluppo è conforme con la ISO 26262
- Metodo y: raccomandato, quindi da applicare se utile per fornire evidenza
- Metodi z1 e z2: qui, almeno il metodo z1 va scelto siccome ha il livello più alto per ASIL C.

ISO 26262 permette al tester di usare anche altri metodi oltre a quelli menzionati nelle tabelle. In quel caso, però, si deve spiegare l'utilità e l'adeguatezza del metodo scelto alternativamente. [ISO 26262]

2.3 AUTOSAR (K1) [15 Min]

Introduzione

AUTOSAR è un acronimo per „AUTomotive Open System ARchitecture“ e per le partnership di sviluppo collegato ad esso. Questa partnership è stata stabilita nel 2003 ed include principalmente produttori e fornitori del settore automotive. L'obiettivo della partnership era: “Di creare e stabilire uno standard liberamente disponibile per un'architettura software nel settore dei veicoli”. Pertanto, questo standard è indirizzato ad affrontare l'importanza e la complessità crescenti del software [14]. Oggi, l'AUTOSAR è uno standard stabilito a livello mondiale per i sistemi E/E. Quindi, il tester verrà certamente a contatto con prodotti AUTOSAR. Perciò, è importante per il tester conoscerne gli obiettivi, il progetto base e i punti di contatto con il lavoro del tester.

2.3.1 Obiettivi della AUTOSAR (K1) [5 Min]

I seguenti obiettivi di progetto per l'AUTOSAR sono guidati dal principio “Collaborazione negli standard, competizione nell'implementazione”: [14, 15]:

1. Supportare la trasferibilità (portabilità) del software
2. Supportare la scalabilità a differenti veicoli e varianti di piattaforme
3. Supportare differenti domini funzionali
4. Definizione di un'architettura aperta, che sia manutenibile così come adattabile ed espandibile
5. Supportare lo sviluppo di sistemi affidabili, caratterizzati da disponibilità, affidabilità, sicurezza (funzionale così come per quanto riguarda cybersecurity, “safety & security”), integrità e manutenibilità
6. Supportare un uso sostenibile di risorse naturali
7. Supportare la collaborazione tra vari partner
8. Standardizzazione della funzionalità software di base delle unità di controllo elettronico automotive (ECU)
9. Supporto di standard in ambito automotive, applicabili per veicoli e tecnologie all'avanguardia.

2.3.2 Struttura generale della AUTOSAR (K1) [informativo] [5 Min]

L'architettura della AUTOSAR consiste in tre strati separati:

- Lo strato che è indipendente dall' hardware, contenente i componenti software (SW-C) con l'AUTOSAR.
- Lo strato orientato all'hardware con software base standardizzato (BSW).
- Lo strato di astrazione con l'ambiente di runtime (RTE) dell'AUTOSAR. Questo controlla lo scambio di dati all'interno e all'esterno delle unità di controllo elettronico e lo implementa tra i componenti del software, nonché tra i componenti del software ed il software base.

Un ulteriore aspetto è la metodologia AUTOSAR per lo sviluppo armonizzato dell'unità di controllo del software. In base a questo, l'OEM e i fornitori si scambiano informazioni sui file di descrizione tramite modelli AUTOSAR (i cosiddetti "file arxml"). [14, 16]:

- La "descrizione di configurazione ECU" include i dati per l'integrazione dei SW-C sull'unità di controllo elettronico.
- La "descrizione di configurazione del sistema" include i dati per l'integrazione di tutte le unità di controllo in un veicolo.
- L' "estratto ECU" include i dati dalla "descrizione di configurazione del sistema" per una singola unità di controllo elettronico.

2.3.3 Influenza dell'AUTOSAR sul lavoro del tester (K1) [5 Min]

AUTOSAR influenza il lavoro del tester, specialmente ai seguenti livelli di test¹⁶:

- Test dei componenti del software e test di integrazione del software in un ambiente virtuale (ad esempio software in loop): con l'aiuto di un BSW virtuale e del RTE, un tester può testare i Componenti SW dell'applicazione in fasi iniziali [17, 18].
- Il test del software e i test di integrazione del software nell'unità di controllo reale: qui, il tester ottiene l'accesso alla comunicazione sul RTE. Con ciò, il tester può misurare e stimolare il comportamento del SW-C a runtime [19].
- Il test di accettazione AUTOSAR è un test del software del sistema che assicura la conformità con la funzionalità AUTOSAR ai livelli di comunicazione ed applicazione. L'esecuzione del test di accettazione AUTOSAR è opzionale [20, 21].
- Test di integrazione del sistema: Integrazione funzionale e connessione di unità di controllo elettronico differenti (ad esempio, anche nel veicolo). Simulando funzionalità mancanti, probabilmente distribuite, il tester può valutare precocemente il comportamento del sistema [17].

2.4 Comparazione (K2) [20 Min]

2.4.1 Obiettivi di ASPICE e ISO 26262 (K1) [5 Min]

Esistono molteplici standard che propongono requisiti per lo sviluppo del prodotto. Di solito, questi evidenziano aspetti differenti nello sviluppo. La ISO 26262 e l'ASPICE sono confrontate qui, per quanto riguarda i loro obiettivi.

ISO 26262 [3] ha l'obiettivo di evitare rischi da guasti sistematici nello sviluppo e da guasti dell'hardware nell'esecuzione presentando requisiti e processi adeguati. Per lo sviluppo di sistemi E/E, definisce i requisiti per i processi e i metodi che vanno utilizzati dal tester. Questi dipendono dal livello ASIL dell'elemento.

ASPICE [9] ha lo scopo di determinare la capacità del processo di sviluppo del prodotto nell'ambito delle valutazioni. Per fare ciò, ASPICE definisce criteri valutabili per questi processi. In contrasto con la ISO 26262, questi sono indipendenti dalla criticità e dal livello ASIL dei prodotti.

¹⁶ Livelli di test; vedere anche 2.4.2

2.4.2 Comparazione dei livelli di test (K2) [15 Min]

Sia la ISO 26262 che l'ASPICE descrivono livelli di test. Tuttavia, questi non sono completamente consistenti con i livelli di test del CTFL® [21]. Perciò, per una collaborazione effettiva ed efficiente, i tester dovrebbero avere una comprensione comune di tutti i livelli di test.

Il termine "sistema" usato nell'ASPICE e i termini "sistema" e "elemento" usato nella ISO 26262 si riferiscono ad un prodotto costituito da componenti hardware e software. Il CTFL®, comunque, si riferisce al software quando utilizza il termine "sistema". Pertanto, i livelli di test secondo ISTQB® [21] possono essere mappati ai livelli di test nella ISO 26262 e nell'ASPICE come segue:

ISTQB®	ISO 26262	ASPICE 3.0
Test di accettazione	Validazione sicurezza (4-9) ¹⁷	Nessun equivalente
Test di sistema di sistemi ¹⁸	Integrazione dell'elemento e relativi test (4- 8) ¹⁹	Test di qualifica del sistema (SYS.5)
Test di integrazione di sistema		Test di integrazione dei sistemi (SYS.4)
Test di sistema	Verifica dei requisiti di sicurezza del software (6-11) Integrazione del software e relativi test (6-10)	Test di qualifica del software (SWE.6)
Test di integrazione di componente		Test di integrazione del software (SWE.5)
Test di componente	Software-Unit-Test (6-9)	Verifica di unità del software (SWE.4)

Tabella 2: Assegnazione dei livelli di test

Secondo il ISTQB® CTFL® Core Syllabus ([21], [48]) le tecniche di test sono in gran parte applicabili indipendentemente dai livelli di test. Anche ASPICE generalmente non assegna alcuna tecnica ai livelli di test. Perciò, entrambi lasciano la scelta ai tester. Nella ISO 26262 d'altra parte, ci sono tabelle metodologiche individuali per ciascun livello di test (vedere capitoli 2.2.5 e 2.2.4.2). Queste forniscono raccomandazioni al tester su quali tecniche dovrebbe usare, a seconda del livello ASIL.

¹⁷ La validazione di sicurezza copre solo le parti del test di accettazione secondo ISTQB.

¹⁸ Il testing di molti sistemi distribuiti eterogeneamente [34, 39]

¹⁹ L'integrazione dell'elemento e il relativo test include tre fasi: l'integrazione e il test hardware e software di un elemento, l'integrazione e il test di tutti gli elementi appartenenti all'elemento, e l'integrazione e il test dell'elemento in connessione con altri elementi nel veicolo.

3 Testing in un ambiente virtuale (K3) [160 Min]

Termini

Model in the Loop (MiL), Software in the Loop (SiL), Hardware in the Loop (HiL), Sistema Open-Loop, Sistema Closed-Loop, Modello Ambientale (Automotive)

Obiettivi di apprendimento

- AUTFL-3.1.1 Richiamare lo scopo/la motivazione dietro un ambiente di test nell'ambiente automotive. (K1)
- AUTFL-3.1.2 Richiamare le parti generali di un ambiente di test specifico automotive. (K1)
- AUTFL-3.1.3 Richiamare le differenze tra sistemi Closed-Loop e sistemi Open-Loop. (K2)
- AUTFL-3.1.4 Richiamare le funzioni essenziali, i database e i protocolli di un'unità di controllo automotive. (K1)
- AUTFL-3.2.1.1 Richiamare la struttura di un ambiente di test MiL. (K1)
- AUTFL-3.2.1.2 Spiegare le aree di applicazione e le condizioni di limite di un ambiente di test MiL. (K2)
- AUTFL-3.2.2.1 Richiamare la struttura di un ambiente di test SiL. (K1)
- AUTFL-3.2.2.2 Richiamare le aree di applicazione e le condizioni di limite di un ambiente di test SiL. (K1)
- AUTFL-3.2.3.1 Richiamare la struttura di un ambiente di test HiL. (K1)
- AUTFL-3.2.3.2 Spiegare le aree di applicazione e le condizioni di limite di un ambiente di test HiL. (K2)
- AUTFL-3.2.4.1 Riassumere i vantaggi e gli svantaggi per il testing con l'aiuto dei criteri degli ambienti di test XiL (MiL, SiL e HiL). (K2)
- AUTFL-3.2.4.2 Applicare criteri per l'assegnazione di una determinata estensione del test ad uno o più ambienti di test. (K3)
- AUTFL-3.2.4.3 Classificare i tre ambienti di test XiL (MiL, SiL, HiL) nel V-model. (K1)

3.1 Ambiente di test in generale (K2) [30 Min]

3.1.1 Motivazione per un ambiente di test nello sviluppo automotive (K1) [5 Min]

Il tester affronta sfide particolari. Da un lato, è previsto che inizi il testing il prima possibile per trovare difetti all'inizio del processo di sviluppo. Dall'altro, necessita di un ambiente realistico per testare il sistema e trovare i difetti che si manifesterebbero nel prodotto finale. Il tester può risolvere questo conflitto usando ambienti di test adatti che si combinino con le differenti fasi dello sviluppo. Così facendo, il tester è in grado di implementare ed eseguire le sue attività di test individuali prima che sia disponibile l'unità di controllo elettronico (ECU) completamente prodotta o sviluppata. Utilizzando ambienti di test differenti, può simulare situazioni ed eseguire casi di test che sarebbero difficili da riprodurre nel veicolo finale, ad esempio, cortocircuiti e circuiti aperti nei cablaggi o sovraccarico nelle comunicazioni di rete. [24]

3.1.2 Parti generali di un ambiente di test (K1) [5 Min]

Per far sì che il tester sia in grado di eseguire le sue attività, ha bisogno di un ambiente di test nel quale siano simulate le parti mancanti. Questo ambiente aiuta il tester a stimolare gli input dell'elemento di test e ad osservarne gli output, anche chiamati 'point of control' (PoC) e 'point of observation' (PoO). Secondo ISO/IEC/IEEE 29119, un ambiente di test consiste nelle seguenti parti:

- Hardware dell'ambiente di test (computer, se necessario anche un computer con funzioni in tempo reale, banco di test, kit di sviluppo, ...)
- Software dell'ambiente di test (sistema operativo, software di simulazione, modelli ambientali)
- Strutture di comunicazione (accesso alle reti, data logger)
- Strumenti (oscilloscopio, strumenti di misurazione)
- Laboratorio (protezione da radiazioni elettromagnetiche e rumori)

Una parte importante dell'ambiente di test è il modello ambientale. I modelli sono una parte importante dell'ambiente di test virtuale. Essi rappresentano aspetti del mondo reale come il motore a combustione, le trasmissioni, i sensori del veicolo e le unità di controllo elettronico o persino il guidatore o le condizioni della strada. L'ambiente di test ha anche differenti punti di accesso. Il tester può usarli per misurare e osservare l'elemento di test [25].

3.1.3 Differenze tra Closed-Loop e Open-Loop (K2) [15 Min]

L'ambiente di test è usato per stimolare le interfacce in input del dispositivo in test e monitorarne gli output attraverso le interfacce in output. Successivamente, viene analizzato il comportamento alle interfacce in output. In un test riuscito, il comportamento osservato corrisponde all'output previsto.

In genere, ci sono due tipi di sistema di controllo, closed loop e open loop. La differenza risiede nel modo in cui l'unità di controllo elettronico reagisce all'ambiente, il che genera requisiti di simulazione differenti per l'ambiente di test virtuale.

3.1.3.1 Sistema Open-Loop

In un sistema open-loop, gli output del sistema non hanno relazione con gli input. Il sistema è aperto e non ci sono feedback. In questo caso gli input dell'elemento di test sono direttamente definiti dal tester nella procedura di test.

Il caso di applicazione per sistemi Open-Loop e Closed-Loop dipende fortemente dal principio di funzionamento dell'elemento di test. Se l'elemento di test ha un comportamento reattivo o se rispecchia una macchina a stati, è preferibile un sistema Open-Loop. Nell'elettronica degli inteni e del telaio ci sono molti esempi di sistemi Open-Loop (vedi luci e interruttori).

3.1.3.2 Sistema Closed-Loop

La stimolazione in un sistema Closed-Loop (anche conosciuta in-the-Loop) prende in considerazione l'output dell'elemento di test. Questo viene fatto tramite un modello ambientale, il quale raccoglie gli output e li inoltra direttamente o indirettamente all'input dell'oggetto di test. Pertanto, un loop di controllo viene creato nell'ambiente di test.

Per il testing dei controller, i sistemi Closed-Loop sono usati più spesso. Usandoli, il tester può testare funzioni complesse come i controlli del motore e del cambio così come i sistemi di assistenza alla guida come il sistema di frenatura antibloccaggio (ABS®) o il controllo della dinamica del veicolo (ESP®). [26, 27]

3.1.4 Interfacce essenziali, database e protocolli di comunicazione di un'unità di controllo elettronico (K1) [5 Min]

Un'**unità di controllo** nell'ambiente automotive è un sistema embedded, che è costituito da hardware e software. L'unità di controllo elettronico riceve diversi input analogici e digitali, che raccolgono costantemente dati ambientali sotto forma di tensione, corrente e temperatura. Inoltre, i sistemi di comunicazione Bus forniscono ulteriori informazioni all'unità di controllo. Queste informazioni provengono da sensori o altre unità di controllo elettronico, che raccolgono ed elaborano le informazioni stesse o ne generano di nuove. L'oggetto di test gestisce i dati nella memoria per elaborare l'azione, le informazioni o i dati in output. Gli output generati sono anche eseguiti tramite pin output analogici e digitali, sistemi bus o interfacce di diagnosi.

I database sono Data Warehouse (contenitori di dati) e definiscono i segnali di input e output dell'unità di controllo. Questi dati includono anche descrizioni, unità e formule di conversione dei segnali.

I protocolli di comunicazione descrivono lo scambio di dati tramite le corrispondenti interfacce fisiche. Questi protocolli definiscono quale tensione o sequenza di bit rappresenta un corrispondente valore del segnale.

La scelta del database e del protocollo di comunicazione dipende dalla funzione dell'unità di controllo elettronico. Ad esempio, per accedere alle funzioni di diagnosi nell'unità di controllo, il tester necessita dell'informazione sul database utilizzato (ad esempio ASAM MCD2 D; o anche "Open Diagnostic Data Exchange") e sul protocollo di comunicazione ("Unified Diagnostic Services" secondo la ISO 14229). Ulteriori database specifici automotive sono definiti ad esempio nello standard ASAM [27, 28].

3.2 Testing in ambienti di test XiL (K3) [130 Min]

Nel settore automotive, sono usati i seguenti tipi di ambiente di test XiL:

- Model in the Loop (MiL),
- Software in the Loop (SiL),
- Processor in the Loop²⁰ (PiL),
- Hardware in the Loop (HiL) e
- Vehicle in the Loop²¹ (ViL)

Qui, il tester dovrebbe familiarizzare con gli ambienti di test (MiL, SiL e HiL) e comprenderli. I seguenti paragrafi approfondiscono la struttura e le aree di applicazione dei diversi ambienti di test. XiL in tal senso si pone come termine generico per i differenti ambienti di test.

3.2.1 Model in the Loop (MiL) (K2) [20 Min]

3.2.1.1 Struttura di un ambiente di test MiL

In un ambiente di test MiL, l'elemento di test è disponibile come modello. Questo modello è eseguibile ma non compilato per un hardware speciale. Tali modelli sono plasmati dagli sviluppatori utilizzando strumenti di modellazione speciali. Per far sì che il tester sia in grado di eseguire e testare quei modelli, necessita di un ambiente di test. Ciò è implementato per la maggior parte nello stesso ambiente di sviluppo dell'elemento di test stesso. Questo ambiente di test può contenere inoltre un modello ambientale. Il tester può stimolare e osservare l'elemento di test tramite punti di accesso. I punti di accesso possono essere posti arbitrariamente nel modello dell'elemento di test e anche nel modello ambientale. Il modello dell'elemento di test è connesso al modello ambientale e può essere implementato facilmente e utilizzato come sistema Closed-Loop.

3.2.1.2 Aree di applicazione e condizioni di limite di un ambiente di test MiL

Con un ambiente di test MiL, il tester è in grado di testare la progettazione del sistema funzionale. Durante lo sviluppo (seguendo il V-model generale) il tester può anche testare i singoli componenti sino anche ad un intero sistema. Per eseguire il test, il tester ha bisogno del computer e della corrispondente simulazione software includendo il modello ambientale. Il modello ambientale diventa più complesso man mano che l'ambito delle funzioni dell'elemento di test aumenta. Gli aspetti della realtà e dei fattori ambientali sono molto complessi. Anche i tempi di esecuzione per i modelli aumentano in modo sproporzionato. Pertanto, lo sforzo per implementare un ambiente di test MiL diventa inutile da una certa fase dello sviluppo in poi.²²

Usando un ambiente di test MiL, il tester può testare la funzionalità dei modelli su tutti i livelli di sviluppo nella fase iniziale dello stesso (lato sinistro del V-model). Ma non è comune abilitare il modello ambientale per simulare bus, funzioni di diagnosi o comportamento fisico (come rotture di cavi o cortocircuiti). Queste attività possono essere eseguite più facilmente e a costi inferiori con altri ambienti di test.

In un ambiente di test MiL, si deve notare che l'esecuzione dei test non avviene in tempo reale. Poiché tutti i componenti sono disponibili come modello, l'esecuzione dei test avviene in tempo di simulazione. Più un sistema è complesso, maggiore è il tempo di esecuzione o la potenza di cui il computer necessita per fornire tutte le informazioni necessarie. La durata della simulazione in sistemi più piccoli è minore

²⁰ Questo ambiente di test non è considerato in questo syllabus ed è puramente informativo.

²¹ Questo ambiente di test non è considerato in questo syllabus ed è puramente informativo.

²² Questo vale anche per tutti gli altri ambienti XiL.

dell'esecuzione in tempo reale. Tuttavia, un grande vantaggio è che il tester può mettere in pausa la simulazione in qualsiasi momento per eseguire analisi e valutazioni dettagliate.

3.2.2 Software in the Loop (SiL) (K1) [10 Min]

3.2.2.1 Struttura di un ambiente di test SiL

L'elemento di test viene compilato per uno specifico ambiente di test SiL. Ciò significa che il codice sorgente è stato compilato con uno strumento software per una determinata architettura di computer. Questo codice macchina è leggibile (solo) dall'ambiente di test poiché è costituito da set di dati binari. Affinché l'ambiente di test possa accedere ai segnali, è necessario un wrapper. Un wrapper è un software aggiuntivo che crea un'interfaccia di accesso specifica per il codice macchina. Perciò, il tester può stimolare segnali software e osservarli. Il wrapper definisce i punti di accesso all'elemento di test ma non ne esegue le attività funzionali.

Per la simulazione, è necessario un modello ambientale. L'elemento di test è collegato all'ambiente di test con l'aiuto del wrapper. L'esecuzione del test viene eseguita su un computer senza un hardware speciale. Il tester, però ha bisogno di uno strumento software che sia in grado di creare un wrapper per l'oggetto di test con punti di accesso all'ambiente di test.

3.2.2.2 Aree di applicazione e condizioni limite di un ambiente di test SiL

Se lo sviluppatore genera codici sorgente basati su un modello, il comportamento reale del software può essere diverso dal comportamento previsto. Ciò può essere causato da differenti tipologie di dati nel modello (principalmente virgola mobile) e nel codice software compilato (principalmente virgola fissa) ma anche da diversi spazi di memoria. Queste anomalie nel comportamento previsto possono essere testate per la prima volta in un ambiente di test SiL. Il tester può usare tecniche come il Testing back-to-back (vedere anche 4.2.2) per confrontare i comportamenti.

Il tester esegue i test, analogamente all'ambiente di test MiL, nel tempo di simulazione. In base alla tecnica di calcolo e alla complessità del modello ambientale, questo tempo di simulazione può essere più lungo o più corto di quello in tempo reale. Il tester può mettere in pausa l'esecuzione in qualunque momento per eseguire analisi e valutazioni dettagliate. I test funzionali, di interfaccia e di regressione sono tipi di test molto comuni che possono essere valutati in un ambiente di test SiL. D'altra parte, i performance test e i test di affidabilità sono inusuali. Queste caratteristiche del software sono influenzate principalmente dall'hardware di destinazione.

3.2.3 Hardware in the Loop (HiL) (K2) [20 Min]

3.2.3.1 Struttura dell'ambiente di test HiL

Se l'elemento di test è disponibile come prototipo o è già stato completamente sviluppato, il tester può usare un ambiente di test HiL per eseguire i test. Le parti tipiche di un ambiente di test HiL sono:

- Un alimentatore per impostare diverse tensioni di alimentazione
- Un computer con capacità elaborative in tempo reale in grado di eseguire il modello ambientale
- Diverse parti reali che non sono implementate nel modello ambientale
- Un'elaborazione del segnale del tipo e dell'ampiezza del segnale stesso
- Una fault insertion unit (FIU, vedere anche 4.2.3) per la simulazione di interruzioni e cortocircuiti
- Una breakout box come interfaccia di accesso aggiuntiva nel cablaggio
- Una simulazione bus rimanente per la simulazione dei partecipanti bus inesistenti

3.2.3.2 Aree di applicazione e condizioni limite di un ambiente di test HiL

I punti di accesso in un ambiente di test HiL sono diversi. Il tester deve essere consapevole del fatto che l'utilizzo di punti di accesso errati all'elemento di test può rendere inutili i risultati del test stesso. La conoscenza dei diversi punti di accesso e delle loro connessioni nell'ambiente di test HiL consente di implementare, eseguire e valutare test efficaci.

L'ambiente di test HiL è più complesso degli ambienti di test menzionati precedentemente (MiL e SiL) a causa delle sue molteplici parti. Il tester deve padroneggiare questa complessità per affrontare le sue attività di test. L'ambiente di test HiL può essere utilizzato per test di componente, test di integrazione e test di sistema. L'obiettivo è, tra le altre cose, di trovare difetti funzionali e non funzionali nel software e nell'hardware.

Con l'aiuto degli ambienti di test HiL, possono essere analizzati diversi livelli di test. Se l'elemento di test è un'unità di controllo elettronico (ECU), è chiamato componente²³ HiL, mentre se è una combinazione di molte unità di controllo elettronico, è chiamato sistema HiL. Il tester usa il componente HiL per testare le funzioni dell'unità di controllo. Nel sistema HiL, il focus è sul testing dello scambio di dati tra le unità di controllo elettronico e sul test di sistema dell'intero sistema.

A differenza degli ambienti di test menzionati precedentemente (MiL e SiL) il tempo di simulazione in un ambiente di test HiL viene eseguito sempre in tempo reale. Il motivo di ciò è che il software viene eseguito su un hardware reale. La pausa o l'arresto non sono più possibili in questo ambiente di test. Pertanto, l'ambiente di test include un computer con capacità tali che, in tempo reale, sia in grado di raccogliere e fornire tutti i segnali rilevanti entro un periodo di tempo predeterminato.

3.2.4 Confronto degli ambienti di test XiL (K3) [80 Min]

3.2.4.1 Vantaggi e svantaggi del testing negli ambienti di test XiL

Il tester deve conoscere gli attributi dei diversi ambienti di test. Così facendo, può comprendere e valutare i vantaggi e gli svantaggi del testing in ciascun ambiente. I criteri sono mostrati nella tabella 3.

Criteri	Ambiente di test MiL	Ambiente di test SiL	Ambiente di test HiL
Vicinanza alla realtà	Basso	Medio-basso	Alto
	La realtà è simulata, molte caratteristiche sono astratte, il focus è sulle strutture e sulla logica	Il software reale compilato può essere eseguito (senza Hardware)	Sistema integrato, in grado di funzionare
Tempo e difficoltà nel debugging	Basso	Medio	Alto
	Difetti riscontrati nel modello dell'elemento di test (regolazione del modello)	Difetti riscontrati nel software programmato (regolazione del software)	Difetti riscontrati a livello di sistema (regolazione del sistema)
Sforzo per l'implementazione e la manutenzione	Basso	Medio	Alto
	Creazione del modello ambientale	Creazione del modello ambientale e del wrapper	Creazione del modello ambientale e cablatura dei componenti hardware
	Basso	Medio	Alto

²³ In questo caso il termine „componente“ è usato per un unità di controllo elettronico (ECU) nel contesto di un sistema E/E.

Sforzo per la preparazione dei test	L'ambiente può essere impostato velocemente	L'ambiente può essere impostato velocemente	Progetto, implementazione e valutazione dei test richiedono uno sforzo elevato
Livello di maturità necessario dell'elemento di test	Basso	Medio	Alto
	I modelli di sistema sono simulati	Le funzioni iniziali sono testate con il software target	Una o più unità di controllo elettronico eseguibili o sistemi parziali sono testati il più interamente possibile
Livello di dettaglio necessario per la base di test (specifico)	Medio	Medio-alto	Alto
	Senza specifica completa vengono testati modelli che contribuiscono anche parzialmente all'accertamento della specifica	Le informazioni rilevanti a livello SW devono essere disponibili (specifica dettagliata dei componenti)	I requisiti possono essere testati a livello di sistema (specifica completa di sistema)
Accesso all'elemento di test	Alto	Medio	Basso
	Tutti i segnali in un modello possono essere osservati e controllati.	Solo i segnali disponibili nel wrapper possono essere osservati e controllati.	Solo i segnali disponibili nell'hardware o nei protocolli di comunicazione possono essere osservati e controllati.

Tabella 3: I criteri e il loro impatto negli ambienti di test MiL, SiL e HiL

3.2.4.2 Assegnazione dei casi di test ad uno o più ambienti di test

Nella seguente tabella gli obiettivi di test sono descritti in modo più dettagliato e sono assegnati all'ambiente di test più adatto.

Tipo di test	Descrizione tramite esempi	MiL	SiL	HiL
Requisiti di test del cliente	Fornitura corretta della funzionalità richiesta. Ciò include la corretta elaborazione dell'input, la corretta reazione all'input così come il corretto output dei dati al punto di uscita.	O	O	+
Meccanismi di test per il rilevamento dei difetti e il loro trattamento	<ul style="list-style-type: none"> Rilevamento e gestione di guasti casuali dell'hardware Rilevamento e gestione di guasti software Trasferimento ad uno stato sicuro dopo che i difetti sono rilevati (ad esempio la disattivazione di un sistema) 	+	+	+
Reazione del test alla configurazione dei dati	Controllare l'influenza dei dati di Configurazione (come insiemi di parametri o codice variante) al comportamento dell'oggetto di test.	O	+	+
Funzioni della diagnosi di test	Fornitura corretta della funzionalità di diagnosi richiesta, come il rilevamento di difetti, la loro impostazione e il requisito di	-	+	+

	ripristino, l'impostazione dei difetti nella memoria dei difetti (ad esempio diagnosi On-Board o nel garage)			
Interazione del test con le interfacce	Controllare le interfacce interne ed esterne dell'elemento di test	O	+	+
Dimostrare l'usabilità	L'oggetto di test osservato dovrebbe essere usabile come richiesto e previsto dall'utente.	-	O	+
<u>Legenda:</u> + raccomandato, o possibile, - non sensato				

Tabella 4: Confronto dei tipi di test negli ambienti di test MiL, SiL e HiL

Questa tabella mostra che gli ambienti di test possono essere adatti a determinati obiettivi di test. Questo approccio diversificato diventa evidente soprattutto nel testing dei meccanismi per il rilevamento dei difetti e il loro trattamento. In conformità con il principio del "Front-loading"²⁴ la conclusione generale è che i requisiti di base e i difetti di progettazione siano già stati rilevati precocemente attraverso il testing. Pertanto, MiL è utilizzato per il rilevamento di difetti di progettazione generali, SiL principalmente per difetti tecnici del software e HiL per difetti tecnici hardware/software. Inoltre, è importante notare che, a parte le verifiche di stabilità e affidabilità, efficienza e prestazioni nonché di usabilità, tutti i tipi di test si concentrano sull'idoneità funzionale dell'elemento di test.

Nella strategia di test, il tester (nel ruolo di test manager) assegna l'ambito del testing a molti ambienti di test diversi. Combinando i criteri delle tabelle 3 e 4 il test manager può scegliere l'ambiente di test ottimale.

3.2.4.3 Classificazione degli ambienti di test XiL (MiL, SiL, HiL) nel modello generale a V

La progettazione tecnica del sistema si trova sul lato sinistro del V-model. Il tester può testare questa progettazione con un ambiente di test MiL. Se l'elemento di test e l'ambiente di test MiL sono sviluppati ulteriormente, il tester può anche eseguire i test di componente e di integrazione con questo ambiente di test.

Il tester può utilizzare un ambiente di test SiL se i singoli componenti dell'elemento di test sono programmati e compilati. Test tipici per un ambiente di test SiL sono test di componente e di integrazione. Questi possono essere trovati sul lato destro del V-model.

Nei test del sistema, alcune funzionalità dell'elemento di test sono state interamente sviluppate. Il tester può eseguire il test del sistema con un ambiente di test HiL. [24]

Con una corretta assegnazione degli ambienti di test ai livelli di test, l'intero processo di test può essere ottimizzato in base ai seguenti tre aspetti:

- Minimizzare i rischi del prodotto
Individuare tipi di errori specifici a livello di test (ad esempio performance test a livello di sistema in un ambiente HiL)
- Minimizzare il costo del test
Per ogni tipo di test sono richiesti gli adeguati livelli di test
Trasferimento dei test a livelli di test precedenti, meno costosi e virtuali
- Conformità agli standard
Nelle tabelle metodologiche dello standard ISO 26262, gli ambienti di test sono raccomandati a seconda dell'ASIL.

²⁴ Prima viene rilevato un difetto, meglio è
Italiano V2.0.2

4 Tecniche di test statiche e dinamiche specifiche dell'automotive [230 Min]

Termini

Standard di codifica, testing back-to-back

Obiettivi di apprendimento

Tecniche di test statico

- AUTFL-4.1.1 Spiegare lo scopo e i requisiti delle linee guida MISRA-C:2012 con l'aiuto di esempi. (K2)
- AUTFL-4.1.2 Applicare una revisione dei requisiti utilizzando le caratteristiche di qualità dello standard ISO/IEC 29148 che sono rilevanti per i tester. (K3)

Tecniche di test dinamico

- AUTFL-4.2.1 Creare casi di test per ottenere la copertura del testing delle decisioni/condizioni modificate. (K3)
- AUTFL-4.2.2 Spiegare l'uso del testing back-to-back fornendo degli esempi. (K2)
- AUTFL-4.2.3 Spiegare il principio del fault injection testing fornendo degli esempi. (K2)
- AUTFL-4.2.4 Richiamare i principi del testing basato sui requisiti. (K1)
- AUTFL-4.2.5 Applicare criteri dipendenti dal contesto per la scelta di tecniche di progettazione dei test adeguate e necessarie. (K3)

4.1 Tecniche di test statico (K3) [75 Min]

Introduzione

Il testing statico consiste nell'esaminare i prodotti di lavoro dello sviluppo del software senza eseguirli. Ciò include sia la valutazione svolta da persone (revisione) che l'analisi statica supportata da strumenti specifici.

4.1.1 Le linee guida MISRA-C: 2012 (K2) [15 Min]

Oggi è allo stato dell'arte che lo sviluppatore rispetti le linee guida di codifica durante la programmazione. Lo standard ISO 26262 lo raccomanda anche per il software di particolare rilievo in ambito sicurezza²⁵. Gli standard di codifica aiutano ad evitare anomalie nel software, le quali possono eventualmente portare a difetti. Allo stesso tempo, supportano lo sviluppatore nel migliorare la manutenibilità e la portabilità del suo software.

Le Linee guida MISRA-C:2012 [15] includono guide per il linguaggio di programmazione C. Esso definisce due tipi di linee guida:

- Le regole sono generalmente verificabili mediante strumenti analitici statici. Ad esempio, che il codice sorgente non includa commenti nidificati.
- Le direttive non sono interamente verificabili con strumenti analitici statici. La ragione è che si riferiscono piuttosto a dettagli del processo di sviluppo o a documenti al di fuori del software.

²⁵ Vedere anche [ISO 26262:2011] Parte 9 Tabella 6

Ad esempio, se lo sviluppatore ha documentato sufficientemente il comportamento implementato.

Ogni linea guida è classificata come uno dei seguenti tre livelli di obbligatorietà:

- Le linee guida “**suggerite**” dovrebbero essere seguite dallo sviluppatore se lo sforzo per rispettarle è appropriato.
- Le linee guida “**richieste**” possono essere trascurate dallo sviluppatore solo se è in grado di spiegarne il motivo in modo esaustivo
- Le linee guida “**obbligatorie**” devono essere seguite dallo sviluppatore. Non sono ammesse eccezioni.

Le organizzazioni possono intensificare individualmente il requisito di una regola o direttiva, ma non possono mai diminuirlo.

4.1.2 Caratteristiche di qualità per la revisione dei requisiti (K3) [60 Min]

Le specifiche sono la base per lo sviluppo e per il testing. Perciò, difetti in tali specifiche portano ad attività di follow-up che richiedono molto tempo e comportano ulteriori costi. Ciò vale soprattutto se i difetti vengono rilevati solo in fasi di sviluppo avanzate, come nel testing di accettazione, o in opera. Le revisioni sono una misura efficace per individuare tempestivamente i difetti nelle specifiche e di conseguenza essere in grado di risolverli in anticipo e a basso costo.

Durante l'analisi dei test, il tester deve controllare le specifiche relative all'elemento di test [21]. Così facendo, le specifiche vengono controllate in particolare per quanto riguarda la loro idoneità come base di test. Le caratteristiche di qualità aiutano il tester durante le revisioni delle specifiche a focalizzare la sua attenzione e a trovare il maggior numero possibile di difetti. ISO/IEC/IEEE 29148:2011[37] include le caratteristiche di qualità per i singoli requisiti così come per i gruppi di requisiti.

Caratteristiche dei requisiti secondo ISO/IEC/IEEE 29148:2011 rilevanti per i tester

Caratteristiche per requisiti individuali rispetto ad una serie di requisiti:

- **Verificabile:** Ogni requisito può essere verificato mediante test statici o dinamici.
- **Non ambiguo:** Ogni requisito contiene condizioni di test chiare.
- **Coerente:** Ogni requisito è coerente in sé e con altri requisiti.
- **Completo:** Ogni requisito considera tutti i possibili casi (anche scenari di errore, interruzione ed eccezione). Allo stesso tempo, tutte le tabelle e i diagrammi utilizzati vengono etichettati e sono definite le abbreviazioni e i termini utilizzati.
- **Tracciabile:** Ogni requisito è contrassegnato chiaramente (ad esempio da un ID). Ciò consente un'analisi dell'impatto e la copertura dei casi di test è trasparente.
- **Delimitato (per una serie di requisiti):** È chiaramente definito qual è l'ambito da sviluppare e quindi da testare.
- **Singolare:** Nessun requisito può essere suddiviso in requisiti parziali ragionevoli.

Come strumento per la revisione, il tester può ad esempio ricavare checklist di revisione dalle caratteristiche. Queste checklist di revisione includono quindi domande adeguate per le caratteristiche precedentemente menzionate. Il tester deve rispondere al meglio delle sue conoscenze e convinzioni. La lista seguente include un estratto delle possibili domande a cui è necessario rispondere per ogni requisito:

- **Verificabile:** Il requisito è verificabile mediante test statici o dinamici al livello di test corrispondente?
- **Esplicito:** Il requisito impedisce qualsiasi spazio all'interpretazione e non si basa sulla conoscenza implicita o data dall'esperienza?

- **Coerente:** Il requisito è coerente con se stesso e rispetto ad altri requisiti?
- **Singolare:** Il requisito non può essere suddiviso in ulteriori requisiti parziali, ad es. Risolvendo collegamenti logici come costrutti “if-then-else” all’interno dei requisiti e annotando separatamente i risultanti requisiti parziali?

Secondo Hobbs: Embedded Software Development for Safety-Critical Systems [30], i requisiti dovrebbero anche essere fattibili, indipendenti dall’implementazione e necessari. Di solito per il tester è difficile valutare queste caratteristiche che tuttavia, influenzano marginalmente la progettazione dei test.

4.2 Tecniche di test dinamico (K3) [155 Min]

4.2.1 Testing delle condizioni, testing delle condizioni multiple, testing delle decisioni/condizioni modificate (K3) [60 Min]

Le tecniche descritte qui sono parte delle tecniche di progettazione di test white-box (per ulteriori dettagli vedere anche il syllabus CTAL-TTA). Il tester ricava i casi di test direttamente dalla struttura dell’elemento di test (ad esempio dal codice sorgente).

Rispetto al testing delle decisioni, nel quale il tester progetta i casi di test con attenzione alla copertura delle decisioni nel codice (vedere [21]), il testing delle condizioni si riferisce alle condizioni individuali all’interno di una decisione. Pertanto, queste tecniche affrontano il modo *in cui* viene presa una decisione: ogni decisione consiste in una o più condizioni “atomiche”. Se il tester esegue un caso di test, ognuna di queste condizioni può avere il valore “vero” o “falso”. Il valore complessivo della decisione risulta quindi dalla combinazione logica di questi valori individuali [7].

Se una decisione consiste solamente in una condizione individuale, queste tecniche sono identiche al testing delle decisioni. Altrimenti, esse differiscono come segue [7]:

- **Testing (semplice) delle condizioni** (tecnica A nella tabella 5): Il tester progetta i casi di test con l’obiettivo di coprire i risultati vero/falso di ogni condizione individuale. Con una scelta imprudente dei dati di test (vedere tabella 5), è possibile raggiungere una copertura (semplice) delle condizioni del 100%, ma non la piena copertura dei risultati delle decisioni. Nella tabella sottostante, le condizioni individuali B1 e B2 vengono esercitate sia come vero che come falso, ma il risultato della decisione per entrambi i casi di test viene valutato “falso”.
- **Testing delle condizioni multiple** (tecnica B nella tabella sottostante): Il tester progetta i casi di test con l’obiettivo di coprire tutte le combinazioni di valori relative alle condizioni individuali. Se tutte le combinazioni di valori vengono testate, viene testato anche ogni risultato delle decisioni.
- **Testing delle decisioni/condizioni modificate (MC/DC)** (tecnica C nella tabella sottostante): Questo è simile al testing delle condizioni multiple (B). Tuttavia, la tecnica considera solo combinazioni nelle quali le condizioni individuali (B1, B2) influenzano in modo indipendente il risultato della decisione. Nel caso del test TC4, cambiare B1 o B2 da “falso” a “vero” non risulta in un cambio del risultato della decisione (cioè rimane “falso”). La copertura MC/DC al 100% può essere raggiunta da TC1, TC2 e TC3; non è necessario considerare TC4.

La tabella 5 mostra, con l’aiuto di un esempio, i casi di test necessari per la copertura al 100% a seconda della tecnica di test scelta:

Casi di test	Condizioni individuali		Risultato della decisione per l'espressione: E = B1 AND B2	Tecnica		
	B1	B2		A	B	C
TC 1	B1=VERO	B2=FALSO	E =FALSO	X	X	X
TC 2	B1=FALSO	B2=VERO	E=FALSO	X	X	X
TC 3	B1=VERO	B2=VERO	E=VERO		X	X
TC 4	B1=FALSO	B2=FALSO	E=FALSO		X	

Tabella 5: Confronto delle tecniche di testing delle condizioni (A), testing delle condizioni multiple (B) e testing delle decisioni/condizioni modificate (test MC/DC) (C)

L'esempio mostra i limiti delle tecniche: Nel caso del testing (semplice) delle condizioni (A), nonostante una copertura delle condizioni del 100%, il tester si assume il rischio di coprire solo *un* risultato delle decisioni. Una scelta migliore dei casi di test lo correggerebbe (nell'esempio TC 3 e TC 4).

Con l'uso del testing delle condizioni multiple (B) il tester può coprire tutti i possibili input e output. Tuttavia, il numero di test che vanno eseguiti è il più alto per questa tecnica.

Usando il testing delle decisioni/condizioni modificate (C), il tester può raggiungere una copertura completa di tutte le singole condizioni e di tutte le decisioni con un numero minore di test rispetto al testing delle condizioni multiple.

4.2.2 Testing Back-to-Back (K2) [15 Min]

Il testing back-to-back (anche detto: test comparativo [32]) è più un approccio al testing che una tecnica (progettazione) di test. Esso confronta due o più varianti di un elemento di test. Per fare ciò, il tester esegue lo stesso caso di test su tutte le varianti e ne confronta i risultati. Se i risultati sono identici, il test è stato superato. Se i risultati differiscono, viene analizzata la causa delle differenze rilevate.

Gli elementi di test devono essere basati sugli stessi requisiti dal punto di vista del contenuto. Solo in questo modo possono mostrare un comportamento comparabile. I requisiti non servono come base di test per la loro progettazione. Al contrario, il test back-to-back è previsto mostri le più piccole differenze non volute tra gli elementi di test o l'ambiente di test. Questo test, quindi, non sostituisce il test basato sui requisiti.

Nel caso più semplice, gli elementi di test di un test back-to-back sono versioni differenti dello stesso software. In questo caso, ad esempio una versione precedente dell'elemento di test serve come oracolo per il test back-to-back (simile ad un test di regressione) [33]. Un'altra alternativa è il confronto di un modello eseguibile con il codice generato (manualmente o automaticamente) [32]. In questo caso, è una forma di testing basato sul modello, nel quale il modello eseguibile serve anche come oracolo del test [34]. Questa tecnica è quindi molto adatta per la progettazione di test automatizzati. Qui, il tester deriva dal modello non solo il risultato previsto, ma anche casi di test automatizzati.

4.2.3 Fault injection testing (K2) [15 Min]

Il fault injection testing è più un approccio al testing di robustezza, che una tecnica (progettazione) di test speciale. Le tecniche di programmazione come la gestione degli errori hanno lo scopo di far reagire il sistema a difetti interni ed esterni in maniera robusta e sicura. Per testare queste tecniche, il tester può inserire selettivamente difetti nel sistema ai seguenti punti [34]:

- Difetti in componenti esterni: Ad esempio se il sistema deve rilevare in maniera sicura valori non plausibili dai sensori.
- Difetti nelle interfacce: Se ad esempio la funzione del sistema non deve essere danneggiata da cortocircuiti o messaggi persi.
- Difetti nel software: Se il sistema dovrebbe rilevare e gestire difetti interni.

Nel classico fault injection, il tester inserisce un difetto manipolando il componente reale.

I difetti esterni (detti anche difetti di interfaccia) possono essere simulati dal tester nel tempo di esecuzione. Il fault injection di solito si svolge in un ambiente di test HiL. Qui, un'unità fault insertion (FIU) [35] funge da driver per i difetti fisici. Tra questi difetti si annoverano in particolare i cortocircuiti e i circuiti aperti. La simulazione dei difetti di interfaccia basati sul software spesso può essere eseguita in un ambiente di test SiL.

I difetti nel software spesso possono essere inseriti solo nell'ambiente di sviluppo, ad esempio tramite debugger o XCP. Quindi, in pratica, l'esecuzione spesso richiede molto tempo.

4.2.4 Testing basato sui requisiti (K1) [5 Min]

Il testing basato sui requisiti è più un approccio (una pratica) per il testing [22] e meno una tecnica (progettazione) di test. L'approccio mira a coprire i requisiti con i casi di test. Pertanto, il tester decide se l'elemento di test soddisfa i requisiti.

In questo approccio, il tester analizza i requisiti, ricava le condizioni di test, progetta i casi di test e li esegue. Basandosi poi sull'analisi dei risultati dei test, li affina. Così facendo, il tester può anche creare ulteriori casi di test. Inoltre, applica altre pratiche di test (come il testing basato sull'esperienza). Perciò, può ridurre il rischio di difetti, ad esempio tramite test di regressione sotto forma di test esplorativi.

Se i requisiti sono incompleti o inconsistenti, i test progettati su tale base risentono degli stessi problemi. D'altra parte, il tester può non essere in grado di testare tutti i requisiti se sono molto dettagliati. Qui, è obbligatoria una assegnazione di priorità ai casi di test. [3]

4.2.5 Selezione delle tecniche di test dipendente dal contesto (K3) [60 Min]

Lo standard ISO 26262 (Volume 6) suggerisce che il tester applichi tecniche di progettazione dei test (vedere il capitolo 2.2) a seconda del livello ASIL. Queste, includono tra le altre tecniche menzionate nel CTFL® e precedentemente nel capitolo 4.2:

- Testing basato sui requisiti
- Partizionamento di equivalenza
- Analisi ai valori limite
- Testing delle istruzioni
- Testing delle decisioni
- Testing delle decisioni/condizioni modificate
- Error guessing
- Fault injection
- Testing back-to-back

Tuttavia, la decisione su quale tecnica usare dipende, tra gli altri, dai seguenti fattori:

Stato dell'arte

La tecnica rappresenta lo stato dell'arte attuale per questo scopo? Qui, sono d'aiuto standard come la ISO/IEC/IEEE 29119 e la ISO 26262. Lo standard ISO 26262 suggerisce anche le tecniche applicabili

a seconda del livello ASIL. Deviazioni dalle raccomandazioni dello standard sono discusse nel capitolo 2.2 riguardante la ISO 26262.

Base di test

La base di test fornisce le condizioni di test adatte per la tecnica? Ad esempio, il tester può formare classi equivalenti solo se la base di test include parametri o variabili. Deve essere in grado di raggruppare i loro valori in classi equivalenti ragionevoli. Condizioni simili si applicano ai valori limite. Il tester può testarli solo se l'intervallo dei valori è definito in modo lineare.

Testing basato sul rischio

Per testing basato sul rischio si intende l'identificazione dei prodotti di rischio e la considerazione del livello di rischio per la selezione delle tecniche. Ad esempio, il test di un valore limite ha senso solo se esiste il rischio che si verifichino violazioni di limite e se l'impatto di tali violazioni costituisce un rischio.

Livello di test

La tecnica può essere usata ragionevolmente su un livello di test? I test white-box sono particolarmente adatti se il codice sorgente o la struttura interna fungono da base di test. Nel caso ideale, si può misurare il grado strutturale di copertura. Per i test black-box, l'elemento di test è necessario sia disponibile e osservabile. Ad esempio, il testing di una classe equivalente di un sensore nel test del sistema può essere più efficiente che nel test di componente. Se una tecnica di progettazione dei test non è usabile su un livello di test, il tester dovrebbe sceglierne uno differente a seconda della strategia di test.

Esempio della selezione delle tecniche di test

La seguente tabella contiene un elenco di tecniche di progettazione dei test, supportate da un esempio della valutazione di un utente per quanto riguarda molti fattori menzionati precedentemente e la selezione della tecnica di progettazione dei test basata su tale valutazione.

	Tecnica di progettazione dei test	Consigliata per l'uso con ASIL A?	Base di test adatta?	Rischio, se il difetto non viene rilevato?	Livello di test „test di sistema“ ragionevole ?	Selezione
1	Testing basato sui requisiti	++	SI	++	SI	X
2	Partizione di equivalenza	+	SI	++	SI	X
3	Analisi ai valori limite	+	NO	-	SI	
4	Testing delle istruzioni	++	SI	++	NO	
5	Testing delle decisioni	+	SI	++	NO	
6	MC/DC	+	SI	+	NO	
7	Error guessing	+	NO	++	SI	
8	Fault injection	+	SI	+	NO	
9	Testing back-to-back	+	NO	++	SI	

Tabella 6: Esempio della selezione di una tecnica di test

5 Allegato

Data Base e Protocolli di Comunicazione Automotive

Interfacce	database	Protocolli di comunicazione
Memoria	ASAM MCD-2 MC (anche ASAP2 o A2L)	ASAM MCD-1 XCP (Universal Measurement and Calibration Protocol) ASAM standard CCP (CAN Calibration Protocol)
Bus	ASAM MCD2 NET standard (anche <i>FIBEX - Field Bus Exchange Format</i>)	FlexRay (ISO 17458) CAN (Controller Area Network secondo la ISO 11898-2)
	DBC (communication database for CAN)	CAN (Controller Area Network secondo la ISO 11898-2)
Diagnosi	ASAM MCD2 D (anche ODX) CDD (CANdelaStudio diagnostic description)	<i>KWP2000 (ISO 14230)</i> <i>ISO-OBD (ISO 15031)</i> <i>UDS (ISO 14229)</i>

Tabella 7: Database comuni e protocolli di comunicazione dal settore automotive

AUTOSAR si è standardizzato su un formato XML, che integra i database di un veicolo completo. È il formato ARXML (AUTOSAR Integrated Master Table of Application Interfaces, XML scheme R3.0).

ASAM sta per "Association for Standardization of Automation and Measuring Systems"

5.1 Elenco delle tabelle

Tabella 1: Esempio di una tabella metodologica	24
Tabella 2: Assegnazione dei livelli di test	27
Tabella 3: I criteri e il loro impatto negli ambienti di test MiL, SiL e HiL	34
Tabella 4: Confronto dei tipi di test negli ambienti di test MiL, SiL e HiL	35
Tabella 5: Confronto delle tecniche di testing delle condizioni (A), testing delle condizioni multiple (B) e testing delle decisioni/condizioni modificate (test MC/DC) (C).....	39
Tabella 6: Esempio della selezione di una tecnica di test.....	41
Tabella 7: Database comuni e protocolli di comunicazione dal settore automotive.....	42

5.2 Riferimenti

- [1] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), *ISO/IEC TS 24748-1:2016 Systems and software engineering - Life cycle management - Part 1: Guide for life cycle management*, 2016.
- [2] Verband der Automobilindustrie e.V. (VDA) / QMC Working Group 13 / Automotive SIG, *Automotive SPICE Process Assessment Model*, Berlin: Verband der Automobilindustrie e. V. (VDA), 2008.
- [3] AUTOSAR, „<http://www.autosar.org/specifications/>,“ [Online]. [Zugriff am 04 04 2016].
- [4] ZVEI, *Best Practice Guideline - Software Release*, Frankfurt am Main,: ZVEI, 2016.
- [5] International Software Testing Qualifications Board (ISTQB) / German Testing Board e.V. (GTB), *ISTQB/GTB Certified Tester Advanced Level (CTAL) Syllabus - Technical Test Analyst (TTA) - Deutsche Ausgabe*, German Testing Board e.V. (GTB), 2012.
- [6] Verband der Automobilindustrie e.V. (VDA) / QMC Working Group 13, „Status and outlook VDA QMC working group 13 - Automotive SPICE 3.0, Blue-Gold Volume,“ in *Sixth VDA Automotive SYS Conference*, Berlin, 2016.
- [7] Verband der Automobilindustrie e.V. (VDA) / QMC Working Group 13 / Automotive SIG, *Automotive SPICE Process Assessment / Reference Model*, <http://www.automotivespice.com/download/>, 2015 Version 3.0.
- [8] International Organization for Standardization (ISO), *ISO 26262:2011 Road Vehicles - Functional Safety*, Genf, 2011.
- [9] AUTOSAR, „Glossary AUTOSAR Release 4.2.2,“ [Online]. Available: http://www.autosar.org/fileadmin/files/releases/4-2/main/auxiliary/AUTOSAR_TR_Glossary.pdf. [Zugriff am 03 03 2016].

- [10] H. Wallentowitz, Handbuch Kraftfahrzeugelektronik : Grundlagen, Komponenten, Systeme, Anwendungen ; mit zahlreichen Tabellen, Wiesbaden: Vieweg, 2016.
- [11] K. Borgeest, *Elektronik in der Fahrzeugtechnik*, Springer Vieweg, 2014.
- [12] MISRA Electrical Group MIRA Ltd., *MISRA-C:2012-Programmierrichtlinien – Version 3.*, UK, Warwickshire, 2013.
- [13] 754-2008 - IEEE Standard for Floating-Point Arithmetic, *754-2008 - IEEE Standard for Floating-Point Arithmetic*, 2008.
- [14] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers (IEEE), *ISO/IEC/IEEE 15288:2015 Systems and software engineering - System life cycle processes*, 2015-15-05.
- [15] Verband der Automobilindustrie e.V. (VDA), *Entwicklung softwarebestimmter Systeme - Forderungen an Prozesse und Produkte*, Bd. 13, Verband der Automobilindustrie e.V. (VDA), 2004.
- [16] Measuring, Association for Standardization of Automation and, „<http://asam.net/>,“ 2016. [Online]. [Zugriff am 2016].
- [17] K. Hoermann, M. Mueller, L. Dittmann und J. Zimmer, *Automotive SPICE in Practice in der Praxis – Interpretationshilfe für Anwender und Assessoren*, Heidelberg: dpunkt verlag GmbH, 2. Auflage, 2016.
- [18] National Instruments Germany GmbH, „Einsatz von Fault Insertion Units (FIUs) für die Überprüfung elektronischer Steuergeräte,“ Nr. 25. Juni, 2015.
- [19] Patzer und Zaiser, „Einsatzgebiete für XCP,“ in *XCP-Das Standardprotokoll für die Steuergeräte Entwicklung*, Stuttgart, Vector Informatik GmbH, 2014.
- [20] International Software Testing Qualifications Board (ISTQB) / German Testing Board e.V. (GTB), *ISTQB/GTB Certified Tester Foundation Level (CTFL) Syllabus - Version 2011 1.0.1 - Deutsche Ausgabe*, German Testing Board e.V. (GTB), 2011.
- [21] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers (IEEE), *ISO/IEC/IEEE 29119-1:2013 Software and systems engineering - Software testing - Part 1: Concepts and definitions*, 2013-09-01.
- [22] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), *ISO/IEC 12207:2008 Systems and software engineering - Software life cycle processes*, International Organization for Standardization (ISO), 2008-02-01.

- [23] A. Spillner, T. Roßner, M. Winter und T. Linz, Praxiswissen Softwaretest Testmanagement: Aus- und Weiterbildung zum Certified Tester - Advanced Level nach ISTQB-Standard, Heidelberg: dpunkt.verlag, 2008.
- [24] Verband der Automobilindustrie e.V. (VDA), *Sicherung der Qualität in der Prozesslandschaft*, Bd. Band 4, Verband der Automobilindustrie e.V. (VDA), 2011.
- [25] dpa, „www.motor-talk.de,“ 24 02 2015. [Online]. Available: <http://www.motor-talk.de/news/die-zahl-der-modelle-waechst-der-absatz-nicht-t5219608.html>. [Zugriff am 12 12 2016].
- [26] AUTOSAR, „Requirements on Acceptance Test AUTOSAR TC Release 1.1.0,“ [Online]. Available: http://www.autosar.org/fileadmin/files/standards/tests/tc-1-1/general_auxiliary/AUTOSAR_ATR_Requirements.pdf. [Zugriff am 2016 12 12].
- [27] R. Schönfeld, *Regelungen und Steuerungen in der Elektrotechnik*, Verlag Technik GmbH, 1993.
- [28] AUTOSAR, „Project Objectives AUTOSAR Release 4.2.1,“ [Online]. Available: http://www.autosar.org/fileadmin/files/releases/4-2/main/auxiliary/AUTOSAR_RS_ProjectObjectives.pdf. [Zugriff am 03 03 2016].
- [29] AUTOSAR, „Main Requirements AUTOSAR Release 4.2.1,“ [Online]. Available: http://www.autosar.org/fileadmin/files/releases/4-2/main/auxiliary/AUTOSAR_RS_Main.pdf. [Zugriff am 03 03 2016].
- [30] G. Baumann, „Was verstehen wir unter Test? Abstraktionsebenen, Begriffe und Definitionen,“ FKFS 1. AutoTest; Fachkonferenz zum Thema Test und Diagnose in der Automobilentwicklung, Stuttgart, 2006.
- [31] C. Hobbs, *Embedded Software Development for Safety-Critical Systems*, Taylor & Francis Group, 2016.
- [32] AUTOSAR, „AUTOSAR - The worldwide Automotive Standard for E/E systems,“ *ATZ extra*, p. 5, 2013.
- [33] A. Spillner und T. Linz, *Basiswissen Softwaretest [Elektronische Ressource] : Aus- und Weiterbildung zum Certified Tester - Foundation Level nach ISTQB-Standard*, Heidelberg: dpunkt.verlag, 2012.
- [34] International Software Testing Qualifications Board (ISTQB) / German Testing Board e.V. (GTB), *ISTQB/GTB Standardglossar der Testbegriffe Version 3.1*, Erlangen: German Testing Board e.V. (GTB), 13. April 2016.
- [35] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers (IEEE), *ISO/IEC/IEEE 29119-3:2013 Software and systems engineering - Software testing - Part 3: Test documentation*, 2013-09-01.

- [36] AUTOSAR, „Acceptance Test Main Requirements AUTOSAR TC Release 1.1.0,“ [Online]. Available: http://www.autosar.org/fileadmin/files/releases/tc-1-1/general_auxiliary/AUTOSAR_ATR_Main.pdf. [Zugriff am 2016 03 03].
- [37] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers (IEEE), *ISO/IEC/IEEE 29119-4:2015 Software and systems engineering - Software testing - Part 4: Test techniques*, Bd. 4, 2015.
- [38] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers (IEEE), *ISO/IEC/IEEE 29148:2011 - Systems and software engineering - Life cycle processes - Requirements engineering*, 2011-12-01.
- [39] M. Winter, M. Eksir-Monfared, H. M. Sneed, R. Seidl und L. Borner, *Der Integrationstest: Von Entwurf und Architektur zur Komponenten- und Systemintegration*, München: Carl Hanser Verlag GmbH & Co. KG, 2012.
- [40] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), *ISO/IEC 33020-03:2015 Informationstechnik – Prozessbewertung – Rahmenwerk für Prozessmessungen zur Beurteilung der Prozessfähigkeit*, 01-03-2015.
- [41] M. Conrad und G. Sandmann, „A Verification and Validation Workflow for IEC 61508 Applications,“ *SAE International*, 2009.
- [42] H.-W. Wiesbrock, M. Conrad, I. Fey und H. Pohlheim, „Ein neues automatisiertes Auswertverfahren für Regressions- und Back-to-Back-Tests eingebetteter Regelsysteme,“ *Softwaretechnik-Trends*, Bd. 22, 2002.
- [43] U. Freund, V. Jaikamal und J. Löchner, „Multilevel System Integration of Automotive ECUs based on AUTOSAR,“ [Online]. Available: <http://papers.sae.org/2009-01-0918/>. [Zugriff am 27 09 2016].
- [44] T. Ringler, C. Dziobek und F. Wohlgemuth, „Tagungsband Modellbasierte Entwicklung eingebetteter Systeme - Chancen und Herausforderungen bei der virtuellen Absicherung verteilter Body&Comfort-Funktionen auf Basis von AUTOSAR - S.83 - 93,“ [Online]. Available: <https://www.in.tu-clausthal.de/fileadmin/homes/GI/Documents/MBEES15Proceedings.pdf>. [Zugriff am 27 09 2016].
- [44] Bakshi, U.A.; Baksi; V.U.: *Control Systems*, Edition 2010. (English edition)
- [45] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), *ISO/IEC 2382:2015-05 Information technology - Vocabulary*, 2015-05.
- [46] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), *ISO/IEC 24765:2017-09 Information technology - Vocabulary*, 2017-09.



- [47] German Association of the Automotive Industry (VDA) / QMC Working Group 13 / Automotive SIG, *Automotive SPICE Process Assessment / Reference Model*, <http://www.automotivespice.com/download/>, 2017, Version 3.1.
- [48] International Software Testing Qualifications Board (ISTQB®) – *ISTQB® Certified Tester Foundation Level (CTFL®) Syllabus - Version 2018*.

5.3 Definizioni

I seguenti termini specifici del syllabus (non in grassetto) vengono utilizzati in aggiunta al Glossario ISTQB® [ISTQB®2016]. Questi termini dovrebbero essere usati come definiti qui:

Termini	Definizione / Significato	Termini Chiave Glossario ²⁶	Riferimento
Ambiente di runtime (AUTOSAR)	Il livello di astrazione, che controlla e implementa lo scambio di dati tra i componenti del software AUTOSAR così come tra l'applicazione e il Base Software (BSW), all'interno e all'esterno delle unità di controllo.		[9]
Ambiente di test XiL	Un termine generalizzato per il testing dinamico in diversi ambienti di test virtuali. Vedere anche Hardware in the Loop, Software in the Loop, Model in the Loop	X	
Attributo di processo	Caratteristiche misurabili di un processo per la valutazione della capacità di processo.		[7]
Automotive Open System Architecture (AUTOSAR)	Partnership di sviluppo fondata nel 2003 con l'obiettivo di creare e stabilire uno standard industriale aperto per un'architettura software nel settore automotive.		
Automotive Safety Integrity Level	Uno dei quattro livelli che specificano i requisiti necessari della norma ISO per il componente o l'elemento 26262 e le misure di sicurezza da adottare per evitare un rischio residuo non razionale.	X	[8]
Automotive SPICE	Un modello di riferimento di processo e un modello associato di valutazione del processo nell'industria automotive conforme ai requisiti di ISO/IEC 33002:2015.	X	[9]
Breakout box	Un'unità di misura per analizzare, interrompere o manipolare segnali fisici nei fili.		[10]
Ciclo di vita del Prodotto	Vedere ciclo di vita del Sistema		
Ciclo di vita del Sistema	Le fasi di sviluppo e implementazione di un sistema oltre il PEP fino al suo ritiro.		[15, 1, 14]

²⁶ I termini in grassetto verranno trasferiti nel Glossario ISTQB dopo GA Release.

Termini	Definizione / Significato	Termini Chiave Glossario ²⁶	Riferimento
Ciclo di vita della sicurezza	Ciclo di vita del prodotto di un sistema rilevante per la sicurezza. Inizia con l'idea del prodotto e termina con il suo smaltimento alla fine del suo ciclo di vita.		[8]
Componente software	(AUTOSAR): livello software indipendente dall'hardware che include le singole applicazioni e funzionalità.		[9]
Componenti HiL	Un ambiente di test per l'immagine di una singola Unità di controllo elettronico (ECU).		[10]
Computer con capacità (di elaborazione) in tempo reale	Un'unità di calcolo che garantisce l'elaborazione dei segnali in una finestra temporale definita.		[10]
Copertura delle condizioni	Vedere il Glossario ISTQB® 3.1		
Criteri di verifica	Criteri di verifica che definiscono criteri qualitativi e quantitativi che devono essere soddisfatti per verificare con successo un elemento di test.		[7]
Criteri per la verifica	Una serie di casi di test e di criteri per la verifica del software.		[9]
Cultura della sicurezza	L'attitudine a livello aziendale di sviluppare comunemente un prodotto funzionalmente sicuro.		[8]
Descrizione della configurazione del sistema	I dati utilizzati nell'integrazione di tutte le unità di controllo elettronico in un veicolo.		[9]
Descrizione della configurazione ECU	Include i dati per l'integrazione dei componenti SW sull'Unità di controllo elettronico.		[9]
Dimensione del processo	Tutti i processi rilevanti sono definiti e combinati in categorie di processo e ad un secondo livello in gruppi di processo.		[7]
Dimensione di capacità	Viene definito un numero di attributi di processo divisi per livello di capacità. Gli attributi di processo forniscono le caratteristiche misurabili della capacità di processo.		[7]
Direttiva (MISRA)	Linee guida di programmazione MISRA-C:2012 che non sono completamente verificate dagli strumenti di analisi statica.		[12]

Termini	Definizione / Significato	Termini Chiave Glossario ²⁶	Riferimento
Documentazione del test	Documentazione che descrive i piani o i risultati del testing di un sistema o di un componente. [ISO/IEC/IEEE 24765]		[46]
Electrical Error Simulation	Vedere Fault insertion unit		
Elemento di rilascio	Elemento identificabile in modo univoco con funzioni, proprietà e scopo dichiarati. [15]		[6]
Elemento di test	<ol style="list-style-type: none"> Vedere il Glossario ISTQB® L'elemento di test nel contesto automotive consiste in una configurazione software che include la parametrizzazione di base e di solito anche un hardware ed elementi meccanici. [6] 		[4]
Elenco dei difetti	Una lista di difetti risolti e non risolti. Di solito è una parte del test report.		[4]
Estratto ECU	Include i dati per un'Unità di controllo elettronico dalla descrizione della configurazione del sistema.		[9]
Fault injection	Vedere il Glossario ISTQB® 3.1		
Fault insertion unit	Una parte di un ambiente di test che è in grado di simulare i difetti alle interfacce di un componente o di un sistema.		
Hardware in the Loop	Testing dinamico eseguito utilizzando hardware reale con software integrato in un ambiente simulato.	X	[4]
Indicatore di capacità	Indicatori che possono essere utilizzati per l'esecuzione e la spiegazione di una valutazione della capacità di processo.		[7]
Lista di funzioni	Le funzioni da implementare per un rilascio sono specificate durante la sua pianificazione e indicate nell'elenco delle funzioni.		[4]
Livello di capacità	Uno o più attributi di processo che, se sufficientemente soddisfatti, forniscono un miglioramento significativo della capacità di processo.		[7]
Miglioramento di processo	Vedere il Glossario ISTQB® 3.1		

Termini	Definizione / Significato	Termini Chiave Glossario ²⁶	Riferimento
Model in the Loop	Il testing dinamico eseguito utilizzando un modello di simulazione del sistema in un ambiente simulato.	X	[4]
Modello ambientale (Automotive)	Un'astrazione dell'ambiente reale di un componente o sistema, che include altri componenti, processi e condizioni ambientali, in una simulazione real-time.	X	[10]
Modello di processo	Vedere il Glossario ISTQB® 3.1		
Original equipment manufacturer (OEM)	Nel settore automotive, questo termine è utilizzato per descrivere i produttori di autoveicoli. Vedere anche "Tier 1... n"		[2]
Processo di riferimento	Vedere il Glossario ISTQB® 3.1		
Processo di rilascio	Processo che porta al rilascio.		[4]
Processo di sviluppo del prodotto	Processo che comprende tutte le attività dalla prima idea di prodotto fino alla produzione.		[15]
Produzione	Produzione del prodotto sviluppato. In PEP nell'ambiente automotive noto anche come fabbricazione / fabbricazione in serie.		[14, 1]
Raccomandazione di rilascio	Raccomandazione del tester o del test manager di rilasciare (o non rilasciare) l'elemento di rilascio in base ai risultati del test.		[4]
Raccomandazione per l'installazione	Un'aggiunta al rilascio SW con il quale il fornitore conferma all'OEM che l'elemento ha un rilascio illimitato per le strade pubbliche e può essere usato/testato lì.		
Regola (MISRA)	Linea Guida di Programmazione in MISRA-C:2012 che è verificabile mediante strumenti di analisi statica.		[12]
Revisione del codice	Un controllo di idoneità del codice rispetto allo scopo pianificato e un'analisi della deviazione delle specifiche e degli standard forniti.		[7]
Rilascio	Dichiarazione sulle funzioni implementate, sulle proprietà e sull'uso previsto per un elemento di rilascio. [15]		[15]
Scopo di rilascio	Scopo, per il quale l'elemento di rilascio può o potrebbe essere utilizzato.		[4]

Termini	Definizione / Significato	Termini Chiave Glossario ²⁶	Riferimento
Sicurezza funzionale	L'assenza di un rischio irragionevole a causa di pericoli (hazard) causati dal comportamento errato di Sistemi Elettrici/Elettronici (E/E).	X	[3]
Sicurezza informatica (Automotive)	La condizione di essere al sicuro dalla criminalità elettronica e le misure adottate per raggiungerla.		TBD
Simulazione rest bus	Virtualizzazione dell'interfaccia di comunicazione bus delle unità di controllo elettroniche non esistenti.		
Sistema bus	Rete di molte unità di controllo elettronico che scambiano informazioni tramite le stesse connessioni.		[11]
Sistema closed-loop	Un sistema in cui l'attività di controllo o l'input dipende dall'output o dai cambiamenti nell'output. Vedere anche sistema open-loop.	X	[44]
Sistema E/E	Sistema funzionale di elementi elettrici o elettronici.		[8]
Sistema HiL	Un ambiente di test per l'immagine di un gruppo di unità di controllo elettronico fino all'intero veicolo.		[10]
Sistema open-loop	Un sistema in cui l'azione di controllo o l'input è indipendente dall'output o da modifiche nell'output. Vedere anche sistema closed-loop.	X	[44]
soak test	Un soak test è simile ai test derivati dall'esperienza sul campo, ma utilizza un campione di dimensioni maggiori, utenti normali come tester e non è vincolato a scenari di test specificati in precedenza, ma viene eseguito in condizioni di vita quotidiana. Questo test può avere limitazioni, se necessario, per garantire la sicurezza dei tester, ad es. con misure di sicurezza aggiuntive o attuatori disabilitati.		[8]
Software base	(AUTOSAR): Componenti software standardizzati e orientati all'hardware.		[9]
Software in the loop (SiL)	Testing dinamico eseguito utilizzando software reale in un ambiente simulato o con hardware sperimentale.	X	[4]
Standard di codifica	Uno standard che descrive le caratteristiche di una progettazione o una descrizione di progettazione di dati o componenti di un programma.	X	[46]

Termini	Definizione / Significato	Termini Chiave Glossario ²⁶	Riferimento
Strategia del test di regressione	La strategia del test di regressione definisce quali criteri vengono utilizzati per selezionare i casi di test di regressione quando si verifica una modifica all'elemento di test.		
Strategia di test	Vedere il Glossario ISTQB® 3.1		
Strategia di verifica	Un piano di alto livello per la verifica di un elemento contenente criteri di verifica, attività di verifica con metodi, tecniche e strumenti associati e prodotti di lavoro o processi sottoposti a verifica.		[7]
Tabella metodologica (Automotive)	Una tabella che contiene differenti approcci di test, tecniche di test e tipi di test che sono richiesti in base ad Automotive Safety Integrity Level (ASIL) e al contesto dell'oggetto di test.	X	[8]
Tempo di simulazione	Il lasso di tempo, che è valido per una simulazione al computer.		[10]
Tempo reale	Funzionamento di un sistema informatico in cui i programmi per l'elaborazione dei dati sono costantemente pronti per il funzionamento in modo tale che i risultati dell'elaborazione siano disponibili entro un periodo di tempo predeterminato. A seconda dell'applicazione, i dati possono essere generati secondo una distribuzione temporalmente casuale o in tempi predeterminati.		[45]
Test di integrazione del sistema (ASPICE)	Testing riguardo alla progettazione architettonica del sistema per fornire evidenza della conformità degli elementi del sistema integrato con la progettazione architettonica dello stesso, comprese le interfacce tra gli elementi del sistema.		[9]
Testing back-to-back	Il testing per confrontare due o più varianti di un elemento di test o di un modello di simulazione dello stesso elemento di test, eseguendo gli stessi test case su tutte le varianti e confrontando i risultati. Vedere anche il testing comparativo.	X	[32]
Testing del sistema di sistemi	Testare un sistema di sistemi per verificare che soddisfi i requisiti specificati.		
Testing delle condizioni	Vedere il Glossario ISTQB® 3.1		

Termini	Definizione / Significato	Termini Chiave Glossario ²⁶	Riferimento
Testing delle condizioni multiple	Vedere il Glossario ISTQB® 3.1		
Testing delle decisioni/condizioni modificate (Test MC/DC)	Vedere il Glossario ISTQB® 3.1.		
Testing di qualifica del sistema (ASPICE)	Testing eseguito sul sistema completo e integrato di componenti software, di componenti hardware e di elementi meccanici, per fornire evidenza che il sistema è conforme con i requisiti di sistema e che il sistema completo è pronto per la consegna.	X	[9]
Testing di qualifica del software (ASPICE)	Testing eseguito su software completo e integrato, per fornire evidenza della conformità ai requisiti software.	X	[9]
Tier 1...n	I fornitori nella catena di fornitura ai diversi livelli sono denominati Tier 1...n. I fornitori diretti dell'OEM sono chiamati Tier 1, i fornitori di un Tier 1 sono chiamati Tier 2, ecc.		[2]
Tracciabilità	Vedere il Glossario ISTQB® 3.1		
Virgola fissa	Un numero composto da un numero fisso di cifre. La posizione della virgola è fissa.		
Virgola mobile	Una rappresentazione approssimativa di un numero reale.		[13]

5.4 Abbreviazioni

Le seguenti abbreviazioni sono usate in questo syllabus:

Abbreviazione	Definizione / Significato	Riferimento
ACQ	Acquisition	[7]
ASIL	Automotive Safety Integrity Level	[8]
ASAM	Association for Standardisation of Automation and Measuring Systems	[18]
ASPICE	Automotive SPICE	
AUTOSAR	Automotive Open System Architecture	[9]
AUTOSIG	Automotive Specific Interest Group	[17]
BP	Pratica di Base	[7]
BSW	Base Software	[9]
CTFL®	Certified Tester Foundation Level	
E/E	Elettrico / Elettronico	
ECU	Unità di Controllo Elettronico	
EES	Electrical Error Simulation	[16]
EOP	End-of-Production (fine della produzione)	
FIU	Fault Insertion Unit	[18]
GP	Pratica Generica	[7]
HiL	Hardware-in-the-Loop	
IEC	International Electrotechnical Commission	
ISO	International Organization for Standardization	

Abbreviazione	Definizione / Significato	Riferimento
ISTQB®	International Software Testing Qualifications Board	
MAN	Management (ASPICE)	[7]
MC/DC	Copertura delle Decisioni/Condizioni Modificate	
MIL	Model in the loop	
MISRA	Motor Industry Software Reliability Association	
OEM	Original Equipment Manufacturer	
PA	Attributo di Processo	[7]
PEP	Product Evolution Process	[15]
PIM	Process Improvement (ASPICE)	[7]
QM	Quality Management	
REU	Reuse (ASPICE)	[7]
RTE	Run Time Environment (ambiente di runtime)	[9]
SIL	Software in the Loop	
SOP	Start-of-Production (inizio della produzione)	
SPICE	Software Process Improvement and Capability Determination	[7]
SPL	Supply (ASPICE)	[7]
SUP	Support (ASPICE)	[7]
SW	Software	
SW-C	Componente Software	[9]
SWE	Software Engineering (ASPICE)	[7]
SYS	System Engineering (ASPICE)	[7]
VDA	German Association of the Automotive Industry	

Abbreviazione	Definizione / Significato	Riferimento
WP	Prodotto di Lavoro	[7]
XCP	Universal Measurement and Calibration Protocol	[19]
XIL	Acronimo generico per diversi In the Loop	

5.5 Indice Analitico

A

Ambienti di test XiL 33
ASIL 24
Automotive SPICE 16
AUTOSAR 26

C

Caratteristiche di qualità 39
Categoria di processo 17
Ciclo di vita del sistema 13
Ciclo di vita safety 21
Copertura delle condizioni 40
Criteri di verifica 20

D

Display di livelli 17
Documentazione del test 19

E

Elemento di rilascio 14

F

Fault injection 41

G

Gruppo di processo 17

H

Hardware in the Loop 34

I

Integrazione 26

L

Linee guida di codifica 39
Livelli di test 23, 27, 28

M

Miglioramento di processo 16, 17
Model in the Loop 33
Modelli di processo 16
Modello ambientale 31, 33

P

Processi di riferimento 16

R

Rilascio 14



Software. Testing. Excellence.



S

Safety funzionale 21
Sistema Closed-Loop 31
Sistema Open-Loop 31
Software in the Loop 34
Strategia del test di regressione 19
Strategia di test 19
Strategia di verifica 19

T

Tabelle metodologiche 24, 29
Test delle condizioni multiple 40
Test di accettazione 28
Test di componente 28
Test di integrazione 28

Test di integrazione del sistema 18, 27, 28
Test di qualifica del sistema 18
Test di qualifica del software 18
Test di sistema 27, 28
Test MC/DC 40
Test multi-sistema 28
Testing back-to-back 41
Testing basato sui requisiti 42
Testing delle condizioni 40
Testing delle decisioni/condizioni modificate 40
Tracciabilità 20

V

Verifica 22, 23, 28
Verifica dei componenti software 18