МАГІСТЕРСЬКА РОБОТА

МР. ІПм - 09.00.00.000 ПЗ

Група ІПм-23-1

Вістовський Микола

2024

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Вістовський Микола Миколайович

(прізвище, ім'я, по батькові)

УДК 004.942

(індекс)

МАГІСТЕРСЬКА РОБОТА

Моделі розробки програмних систем на основі концепцій безпеки та захисту

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Вістовський М.М.

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник Крихівський Михайло Васильович, к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту

Завідувач кафедри

доц. Бандура В.В.

(посада) (підпис) (дата) (ініціали та прізвище)

Нормоконтроль

доц. Вовк Р.Б.

(посада) (підпис) (дата) (ініціали та прізвище)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Івано-Франківськ – 2024

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
Зав. кафедрою **ІПЗ**

доц. В.В. Бандура

“ 04 ” вересня 2024 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Вістовському Миколі Миколайовичу

(прізвище, ім’я, по-батькові)

1. Тема магістерської роботи “ Моделі розробки програмних систем на основі концепцій безпеки та захисту ”

керівник проекту (роботи) Крихівський Михайло Васильович, к.т.н., доцент

затверджені наказом закладу вищої освіти від “ 22 ” листопада 2024 р. № 781/7

2. Строк подання студентом проекту (роботи) 15 грудня 2024 р.

3. Вихідні дані до проекту (роботи) Теоретичні концепції та формальні моделі побудови та функціонування інформаційних технологій безпеки та захисту даних

4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)

1. Дослідження предметної області розробки програмних систем на основі концепцій безпеки
2. Методи і стандарти розробки програмних систем з точки зору їх безпеки
3. Принцип використання стандартів для розробки системи
4. Моделі та методології розробки програмних систем на основі концепцій безпеки та захисту

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

1. Огляд процесу формування коду за допомогою Motar toolbox (рис. 1.1)
2. Діаграма контексту проекту (рис. 1.2)
3. Діаграма визначення задач дослідження (рис. 1.3)
4. Контекстна діаграма системи (рис. 1.4)
5. Робочий процес проекту, що містить різні фази проекту та результати (рис. 1.5)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата
Перевірка на плагіат	доц., к.т.н. Вовк Р.Б.	

7. Дата видачі завдання 04 вересня 2024 р.

Керівник _____
(підпис)

Завдання прийняв до виконання_____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури по темі магістерської роботи	15.09.2024	виконано
2	Аналіз концепцій та алгоритмів предметної області	29.09.2024	виконано
3	Дослідження предметної області розробки програмних систем на основі концепцій безпеки	15.10.2024	виконано
4	Методи і стандарти розробки програмних систем з точки зору їх безпеки	08.11.2024	виконано
5	Принцип використання стандартів для розробки системи	20.11.2024	виконано
6	Моделі та методології розробки програмних систем на основі концепцій безпеки та захисту	01.12.2024	виконано
7	Затвердження пояснювальної записки роботи завідувачем кафедри	15.12.2024	виконано

Студент – магістр _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

Магістерська робота: 76 с., 21 рис., 4 табл., 50 джерел.

Тема: Моделі розробки програмних систем на основі концепцій безпеки та захисту

Об'єкт дослідження: процеси розробки програмних систем, засновані на концепціях безпеки та захисту.

Мета роботи: розробка комбінованого процесу розробки програмних систем на основі стандартів безпеки та захисту, а також створення робочого процесу, що забезпечує інтеграцію цих стандартів на кожному етапі розробки.

Предмет дослідження: методології та підходи до інтеграції стандартів безпеки та захисту у V-цикл розробки програмних систем.

Результати дослідження

В роботі розроблено комбінований процес розробки систем безпеки, що інтегрує стандарти FuSa, SOTIF і CE в єдиний V-цикл. Запропонований підхід забезпечує комплексне управління безпекою та захистом на всіх етапах розробки системи.

Висновок

Розроблений комбінований процес розробки надає ефективний інструмент для інтеграції стандартів безпеки в процес створення програмного забезпечення автомобільних систем. Запропонований робочий процес дозволяє підвищити якість розробки, зменшити кількість помилок і прискорити вихід продукту на ринок.

ФУНКЦІОНАЛЬНА БЕЗПЕКА, КІБЕРБЕЗПЕКА, V-ЦИКЛ, АВТОМОБІЛЬНІ СИСТЕМИ, ІНТЕГРАЦІЯ СТАНДАРТІВ, ПРОЦЕС РОЗРОБКИ, ПРОГРАМНА СИСТЕМА

ABSTRACT

Master Thesis: 76 pp., 21 fig., 4 tab., 50 sources.

Thesis Subject: Software system development models based on security and protection concepts

Research object: software systems development processes based on the concepts of security and protection.

The goal of the work: development of a combined process of developing software systems based on safety and security standards, as well as creating a workflow that ensures the integration of these standards at each stage of development.

The subject of research: methodologies and approaches to the integration of security and protection standards into the V-cycle of developing software systems.

Research results

In the work, a combined process of development of safety systems was developed, which integrates FuSa, SOTIF and CE standards into a single V-cycle. The proposed approach provides comprehensive security and protection management at all stages of system development.

Conclusion

The developed combined development process provides an effective tool for integrating safety standards into the software development process of automotive systems. The proposed workflow allows you to improve the quality of development, reduce the number of errors and accelerate the product's release to the market.

FUNCTIONAL SECURITY, CYBER SECURITY, V-CYCLE, AUTOMOBILE SYSTEMS, STANDARDS INTEGRATION, DEVELOPMENT PROCESS, SOFTWARE SYSTEM

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ADAS - Advanced Driver Assistance System

ADS - Automated Driving System

AEB - Automatic Emergency Braking

ASIL - Automotive Safety Integrity Level

AUTOSAR - Automotive Open System Architecture

BRA - Binary Risk Analysis

C - Controllability

CAL - Cybersecurity Assurance Level

CE - Cybersecurity Engineering

DDT - Dynamic Driving Task

E - Exposure

E/E - Electrical/Electronic

ECU - Electronic Control Unit

ESC - Electronic Stability Control

ESCL - Electrical Steering Column Lock

EVITA - E-Safety Vehicle Intrusion Protected Applications

FMEA - Failure Mode and Effects Analysis

FSC - Functional Safety Concept

FSD - Full Self-Driving

FuSa - Functional Safety

HARA - Hazard Analysis and Risk Assessment

HAZOP - Hazard and Operability Study

HEAVENS - HEaling Vulnerabilities to ENhance Software Security and Safety

HW - Hardware

LKA - Lane Keeping Assistance

OCTAVE - Operationally Critical Threat, Asset, and Vulnerability Evaluation

OEM - Original Equipment Manufacturer

S - Severity

SAE - Society of Automotive Engineers

SAHARA - Security-Aware Hazard Analysis and Risk Assessment

SOTIF - Safety Of The Intended Functionality

STPA-SafeSec - Systems Theoretic Process Approach-Safety and Security

SW - Software

TARA - Threat Analysis and Risk Assessment

TSC - Technical Safety Concept

ВСТУП

Актуальність теми.

У сучасних автомобільних системах безпека та захист є основними критеріями успішної розробки та експлуатації. Інтеграція стандартів функціональної безпеки (FuSa), безпеки очікуваного функціонування (SOTIF) та відповідності стандартам кібербезпеки (CE) у процес розробки є необхідною для забезпечення надійності й захищеності автомобільних систем в умовах постійно зростаючої автоматизації та складності програмно-апаратних рішень. Це дослідження спрямоване на вдосконалення підходів до розробки програмних систем шляхом інтеграції зазначених стандартів, що є актуальним завданням для автомобільної промисловості.

Зростання популярності систем допомоги водію (ADAS), систем автоматизованого водіння (ADS) та інших систем безпеки призводить до нових викликів. Однак процес розробки систем, таких як ADAS та ADS, повинен відповідати правилам та нормам, визначеним ISO 26262, ISO 21448 та ISO 21434. Дослідження спочатку зосередилося на отриманні інформації про використання клієнтами процесу розробки та набору інструментів Motar. Наступним кроком у цьому дослідженні було збирання інформації про включення ISO 26262, ISO 21448 та ISO 21434 у процес розробки. Відповідна інформація була зібрана за допомогою огляду літератури. Огляд літератури зрештою дав цінну інформацію, яка була розроблена в концептуальну основу для комбінованого процесу розробки, який інтегрує стандарти ISO.

Отже, пропозиція, яка була в подальшому названа "комбінований процес розробки систем на основі безпеки та захисту", використовувала результати з різних джерел для створення повного робочого процесу, який вирішував усі аспекти стандартів ISO для кожної фази розробки. Зрештою, комбінований процес розробки систем на основі безпеки та захисту був оцінений за допомогою дослідження випадку, яке оцінило, чи може пропозиція допомогти клієнтам у розробці таких систем, як ADAS та ADS.

Зрештою, дослідження зробило висновок, що комбінований процес розробки систем на основі безпеки та захисту є корисним для розробки таких систем, як ADAS та ADS. Тому, за допомогою комбінованого процесу розробки систем на основі безпеки та захисту, може допомогти клієнтам у розробці систем безпеки.

У сучасних умовах розвитку автомобільної індустрії важливим аспектом є забезпечення безпеки та захисту систем, особливо з огляду на швидкий розвиток технологій автономного водіння, інтеграції мережевих технологій і зростання автоматизації. Стандарти функціональної безпеки (FuSa), безпеки очікуваного функціонування (SOTIF) та кібербезпеки (CE) відіграють ключову роль у запобіганні технічним несправностям, мінімізації ризиків та захисті від кіберзагроз. Проте інтеграція цих стандартів у єдиний процес розробки досі залишається викликом для багатьох автовиробників, оскільки вони охоплюють різні аспекти безпеки та мають специфічні вимоги.

Зокрема, актуальність дослідження зумовлена потребою в розробці узгодженого підходу, що дозволяє одночасно враховувати функціональну безпеку (FuSa), запобігання небажаним функціям (SOTIF) та відповідність вимогам кібербезпеки (CE). Оскільки сучасні автомобільні системи стають дедалі складнішими, особливо в умовах зростаючого використання штучного інтелекту та сенсорних технологій, інтеграція цих стандартів у процес розробки є критично важливою для забезпечення безпеки на всіх етапах життєвого циклу продукту.

Дослідження також є актуальним через зростаючі вимоги регуляторів до безпеки автомобілів, а також підвищення уваги з боку споживачів до надійності та захисту їхніх транспортних засобів. Впровадження комбінованого V-циклу розробки, що інтегрує стандарти FuSa, SOTIF та CE, дасть змогу оптимізувати процеси розробки, прискорити вихід на ринок нових продуктів і підвищити загальну безпеку та захищеність автомобільних систем.

Мета дослідження - розробка комбінованого процесу розробки програмних систем на основі стандартів безпеки та захисту, а також створення робочого процесу, що забезпечує інтеграцію цих стандартів на кожному етапі розробки.

Об'єкт дослідження - процеси розробки програмних систем, засновані на концепціях безпеки та захисту

Предмет дослідження - методології та підходи до інтеграції стандартів безпеки та захисту у V-цикл розробки програмних систем.

Відповідно до мети роботи було сформовано наступні **задачі:**

- провести аналіз літератури щодо застосування стандартів FuSa, SOTIF та CE у процесах розробки систем.
- дослідити сучасні методи та інструменти, що використовуються в автомобільній промисловості для забезпечення безпеки систем.
- розробити комбінований V-цикл для інтеграції стандартів безпеки та захисту в процес розробки.
- оцінити ефективність запропонованого підходу на основі літературних та практичних результатів.
- розробити робочий процес на основі отриманого V-циклу для використання в автомобільних системах.

Методи дослідження.

У дослідженні використовувалися такі методи, як аналіз літературних джерел, моделювання процесів розробки, структурований підхід до інтеграції стандартів у V-цикл, а також оцінка розроблених методологій на основі практичного застосування та теоретичних даних

Наукова новизна отриманих результатів.

Розроблено комбінований процес розробки систем безпеки, що інтегрує стандарти FuSa, SOTIF і CE в єдиний V-цикл. Запропонований підхід забезпечує комплексне управління безпекою та захистом на всіх етапах розробки системи, що є інноваційним у сфері автомобільних технологій.

Практичне значення магістерської роботи полягає в ….

Розроблений комбінований процес розробки надає ефективний інструмент для інтеграції стандартів безпеки в процес створення програмного забезпечення автомобільних систем. Запропонований робочий процес дозволяє підвищити якість розробки, зменшити кількість помилок і прискорити вихід продукту на ринок. Окрім того, його можна застосовувати в реальних проєктах для підвищення надійності та безпеки.

Структура магістерської роботи. Робота складається зі вступу, трьох розділів та висновків. Загальний обсяг роботи становить 76 сторінок, і містить 21 рисунок, 4 таблиці, список використаних джерел із 50 найменувань.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗРОБКИ ПРОГРАМНИХ СИСТЕМ НА ОСНОВІ КОНЦЕПЦІЙ БЕЗПЕКИ

1.1. Опис передумов та інструментаріїв дослідження

Нещодавнє опитування, проведене Consumer Reports, показує підвищення рівня цікавості клієнтів щодо передових технологій, таких як Advanced Driver Assistance System (ADAS) і Automated Driving System (ADS) протягом останніх кількох років [1]. Крім того, місцеві органи влади докладають зусиль, щоб зобов'язати виробників транспортних засобів включати ці нові системи безпеки у свої транспортні засоби як стандарт [2]. Таким чином, автомобільний сектор продовжить інвестувати в нові технології безпеки для підвищення загальної безпеки дорожнього руху.

Товариство автомобільних інженерів (SAE) опублікувало стандарт SAE J3016, намагаючись стандартизувати різні класи автоматизації водіння [3]. Цей стандарт спрямований на класифікацію кожного типу системи автоматизації водія, починаючи від повної автоматизації керування водінням. Стандарт SAE J3016 надає виробникам структуру щодо специфікацій і пояснює роль людини під час роботи [4]. Крім того, він також забезпечує ясність щодо нових законів і політики щодо автономного водіння. Отже, SAE J3016 є цінним стандартом, оскільки він може допомогти виробникам під час розробки системи безпеки.

Однак можуть виникнути нові проблеми, якщо збільшиться кількість систем безпеки всередині автомобіля. Наприклад, транспортні засоби сьогодні будуються за допомогою інтеграції нових компонентів, які розробляються та будуються постачальниками [5]. Ці компоненти також постачаються з власним електронним блоком керування (ECU) і програмним забезпеченням, що, зрештою, призведе до збільшення загальної складності. У свою чергу, збільшення загальної складності систем безпеки призведе до збільшення критичних для безпеки функцій і, наприклад, перевантаження

комунікаційних шин [6]. Тому Міжнародна організація зі стандартизації (ISO) розробила три стандарти, які вирішують вищезазначені проблеми та забезпечують керівництво щодо розробки електричних/електронних (E/E) систем для автомобільного застосування в цілому. Ці стандарти, відомі як ISO 26262, ISO 21448 та ISO 21434, застосовуються до певної області, оскільки ISO 26262 фокусується на функціональній безпеці (FuSa) [7], ISO 21448 — на безпеці передбачуваної функціональності (SOTIF) [8] та ISO 21434 щодо розробки кібербезпеки (CE) [9]. Наприклад, стандарт FuSa надає вказівки щодо пом'якшення потенційних ризиків, які виникають через системні збої та випадкові апаратні збої систем E/E [7]. Однак, оскільки стандарти ISO є досить новими, виробники комплектного обладнання та інші компанії, пов'язані з автомобільною промисловістю, ще не повністю усвідомлюють важливість і потенційний вплив.

Таким чином, ICT Group [10] допомагає своїм клієнтам відповідати цим стандартам, оскільки одним із основних видів діяльності ICT Group є надання рішень щодо розробки програмного забезпечення для автомобільних застосувань. Наприклад, IC T Group працює над розробкою свого Motar toolbox, який може бути корисним для розробки систем безпеки для автомобільних додатків. Набір інструментів Motar дозволяє автоматично перетворювати моделі керування в програмний код. Це означає, що клієнту не потрібно писати кожен рядок коду вручну, що в кінцевому підсумку може прискорити процес розробки. Однак, як зазначалося раніше, розробка систем безпеки та інших систем E/E має відповідати вказівкам, викладеним у стандартах ISO. Тому ICT Group хоче допомогти клієнтам у розробці систем безпеки відповідно до принципів і вказівок стандартів FuSa, SOTIF і CE.

Набір інструментів Motar є розширенням MATLAB/Simulink, і його можна використовувати для автоматизації процесу розробки коду. Традиційно процес розробки програмного забезпечення займає багато часу, але за допомогою інструментарію Motar його можна скоротити. Крім того, інструментарій Motar також усуває потребу у висококваліфікованих

програмістах, оскільки код може бути згенерований із графічної моделі автоматично. Це означає, що поки модель правильна, код також має бути правильним. Як було зазначено, код генерується з графічної моделі, тобто моделі Simulink. Спрощений огляд цього процесу можна побачити на рисунку 1.1.

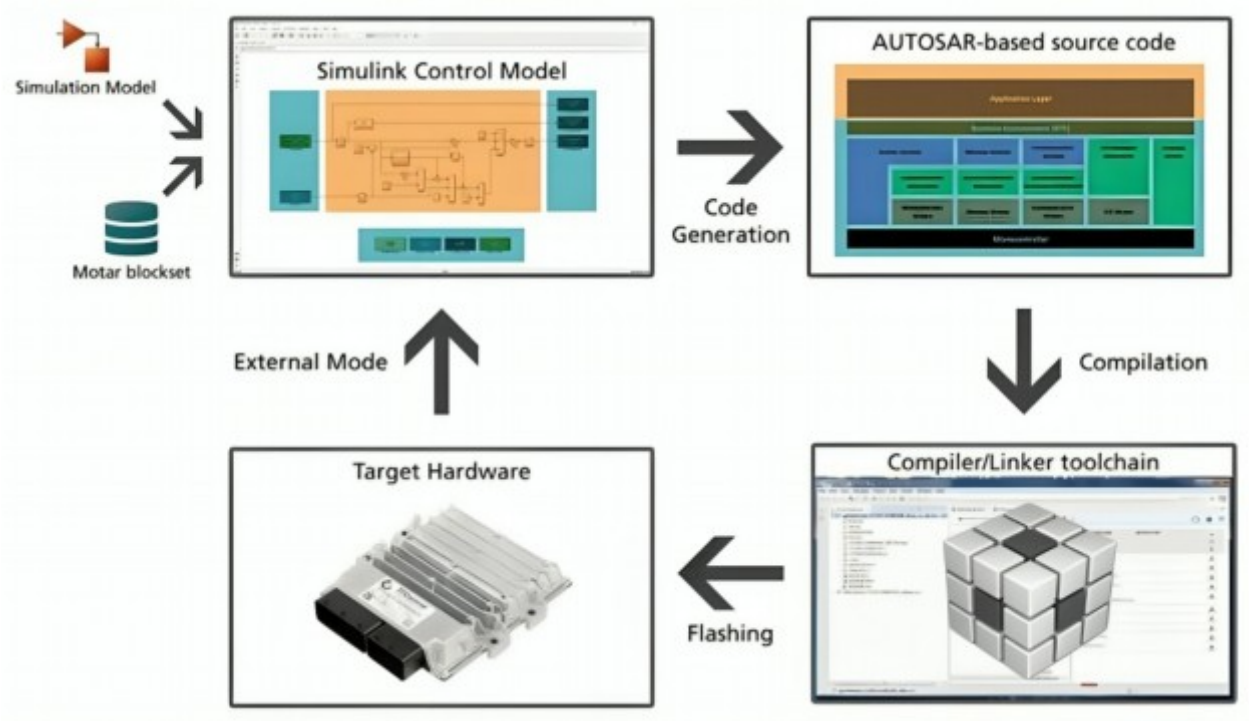

Рис. 1.1. Огляд процесу формування коду за допомогою Motar toolbox [11]

Як показано на рисунку 1.1 модель Simulink поєднана з набором інструментів Motar. Сам інструментарій Motar складається з чотирьох частин які також містять блоки S- Function. Ці блоки S-Function, у свою чергу, надають можливість описувати функції блоку в С-коді. Наступним кроком у процесі є створення всієї моделі. У процесі збирання файли С-коду компілюються. Завдяки компіляції С-коду файли записуються на машинній мові, що дає можливість безпосередньо завантажувати файли коду на цільове обладнання. Після завершення цього кроку весь процес завершено, і апаратне забезпечення має містити функціональні можливості оригінальної моделі.

Загальна якість Motar toolbox може бути впевнена, оскільки згенеровані вихідні файли базуються щодо архітектури відкритої автомобільної системи (AUTOSAR), яка є стандартом, який широко використовується в автомобільній промисловості. AUTOSAR в основному застосовується до ECU, які, у свою чергу, використовуються для керування різними функціями безпеки [12].

За допомогою інструментарію Motar, ICT Group прагне допомогти іншим компаніям у створенні коду. Як було зазначено вище, автомобільна промисловість зараз розробляє такі системи, як ADAS і ADS. Таким чином, ICT Group може надати виробникам оригінального обладнання (OEM) можливість автоматично генерувати код для систем безпеки. Однак розробка автомобільних систем безпеки вимагає ретельного розгляду різних аспектів конструкції. Це означає , що компанія, яка має намір розробити автомобільні системи безпеки, повинна відповідати правилам і нормам, як зазначено в стандартах FuSa, SOTIF і CE. Оскільки стандарти ISO містять вказівки для кожного кроку в процесі розробки, цих процесів слід дотримуватися відповідним чином.

Оскільки загальна складність автомобіля в цілому зросла, важливо належним чином проектувати нові системи, оскільки будь-яка помилка в процесі проектування може мати серйозні наслідки та потенційно поставити під загрозу безпеку. Тому ICT Group прагне допомогти клієнтам у розробці цих нових систем. Отже, життєво важливо правильно розуміти міркування, що стоять за кожним аспектом стандартів ISO. Під час розробки необхідно виконати різні процеси, щоб відповідати кожному стандарту. Цей процес можна спростити, щоб клієнти могли використовувати набори інструментів Motar відповідно до нового процесу розробки, який включає всі аспекти стандартів ISO. Зрештою, це означало б визначення нового процесу розвитку. Незважаючи на те, що FuSa, SOTIF і CE не розрізняють системи рівня SAE 1-5, ISO зараз розробляє новий стандарт ISO 5083. Цей стандарт спрямований на об’єднання всіх стандартів, що стосуються безпеки для автоматизованого

керування [13]. Крім того, він зосереджений саме на системах рівня SAE 3-4 [14]. Отже, це може надати додаткові міркування, які можна впровадити в процес розробки.

1.2. Середовище та методології дослідження

Згідно з визначенням проблеми, розробка автомобільних систем безпеки повинна відповідати стандартам FuSa, SOTIF і CE. Тому створено новий процес розробки, який відповідає стандартам ISO, а також інтегрує використання Motar toolbox. У кінцевому підсумку це призводить до наступної постановки основного питання дослідження:

Як можна інтегрувати стандарти автомобільної безпеки в процес розробки системи для автомобільних додатків?

Щоб знайти відповідь на досліджуване питання, проект розділено на різні етапи. Для кожного етапу дослідження також визначаються окремі дослідницькі питання, щоб загальне дослідження було сформульовано належним чином.

На етапі визначення основним завданням є визначення поточного процесу розробки системи. Крім того, необхідно оцінити, чи клієнти Motar toolbox вже використовують певну форму техніки безпеки під час розробки системи. Фаза аналізу буде зосереджена на дослідженні того, як три стандарти ISO можуть бути включені в процес розробки системи. На етапі реалізації процес розробки системи розроблено для врахування, якщо можливо, всі три стандарти. Необхідно оцінити новостворений процес розробки комбінованої системи, щоб перевірити, чи досягаються загальні цілі. Як було зазначено, мета ICT Group полягає в тому, щоб допомогти клієнтам Motar toolbox у розробці автомобільних систем безпеки, таких як ADAS і ADS. Таким чином, слід оцінити, чи може комбінований процес розробки бути корисним для клієнтів і допомогти їм під час розробки автомобільних систем безпеки.

Необхідно також розглянути граничні умови, методи та можливі припущення цього проекту. Таким чином, дослідження щодо вищезазначених аспектів було виконано на попередніх етапах цього проекту.

Крім того, що стосується граничних умов і обсягу, проект буде зосереджений на створенні комбінованого процесу розробки системи, який інтегрує та дотримується вищезазначених стандартів, а також інтегрує набори інструментів Motar. Будь-яка відповідність набору інструментів Motar щодо стандартів ISO не входить у вказаний обсяг. Нарешті, проект буде зосереджений лише на дорожніх транспортних засобах, оскільки зазначені стандарти ISO застосовуються лише до дорожніх транспортних засобів. Повний зміст проекту показано на рисунку 1.2, де визначено відповідні граничні умови та обмеження щодо обсягу. Крім того, також перераховані відповідні зацікавлені сторони щодо цього проекту.

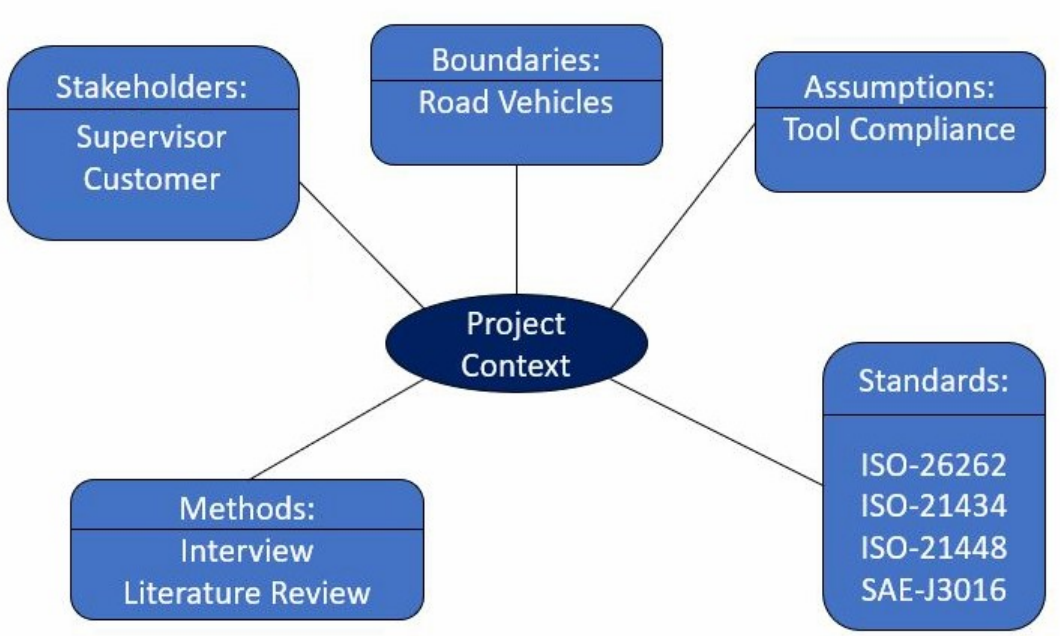

Рис. 1.2. Діаграма контексту проекту

Дослідження поділяється на чотири різні етапи, де кожен етап стосується іншого типу дизайну дослідження. Таким чином, кожна фаза використовує інший тип методу дослідження для проведення зазначеного

дослідження. Типи дизайну дослідження, а також методи перераховані в таблиці 1.1.

Таблиця 1.1.

Перелік етапів проекту та використаних методів дослідження

Phase:	**Research Design:**	**Research Method:**
Definition	Exploratory Research	Interviews
Analysis	Explanatory Research	Literature Review
Realization	Conceptual Research	Literature Review
Evaluation	Evaluative Research	Interviews

Перший етап цього дослідження – етап визначення. Під час цієї фази дослідницький тип дизайну має пошуковий характер. Фокус дослідницького дослідження полягає в отриманні знань про сам предмет [15].

Друга фаза – фаза аналізу. Метою фази аналізу є звернення до інтеграції стандартів ISO у процес розробки системи. Таким чином, прийнятий дослідницький підхід має більше пояснювальний характер, оскільки він зосереджений на забезпеченні кращого розуміння проблеми, що розглядається [17]. Методом дослідження, який використовується на етапі аналізу, є огляд літератури, який зазвичай використовується для пояснювального дослідження.

Третя фаза цього дослідження – фаза реалізації. На цьому етапі будується структура, спираючись на інтерпретацію ідей, отриманих у результаті огляду літератури. Отже, етап реалізації відноситься до концептуального дослідження, оскільки наявна інформація використовується для розробки концептуальної основи [18]. У цьому випадку концептуальна основа є пропозицією щодо процесу розробки, який інтегрує стандарти FuSa, SOTIF і CE. Пропозиція також має включати інтеграцію інструментарію Motar, і пропозиція також має бути розроблена в робочий процес.

Нарешті, етап оцінки зосереджується на оцінці результатів проведеного дослідження. Основне завдання етапу оцінки – оцінити, чи відповідають результати поставленим цілям. Отже, дослідження є оціночним, оскільки

воно спрямоване на оцінку результату дослідження проти очікувань [15]. Щоб визначити, чи це справді так, дослідники можуть вибрати різні методи збору даних [19].

Інструментом, який зазвичай використовується для аналізу наявної проблеми та пошуку творчих рішень для неї, є теорія яка використовується для заохочення користувачів розглянути проблему з різних точок зору. Для цього використовується діаграма з дев'ятьма коробками і створюється відповідно до визначення проблеми, яка показана на рисунку 1.3. Діаграма з дев'ятьма прямокутниками відображає функціональні можливості системних рівнів на горизонтальній осі та часові рамки на вертикальній осі. У цьому випадку підсистемою є Motar toolbox, системою є автомобільні системи безпеки, а надсистемою є процес розробки.

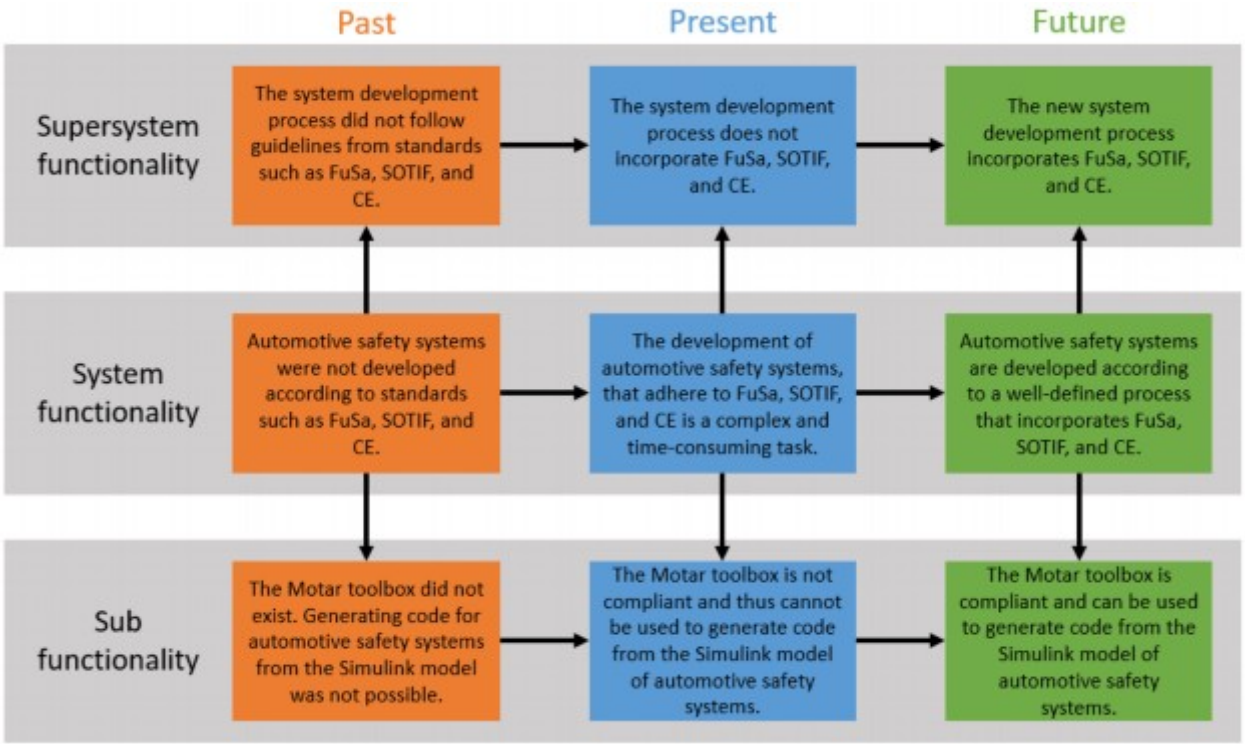

Рис. 1.3. Діаграма визначення задач дослідження

Основна проблема, яка вказана в центрі рисунку 1.3, полягає в тому, що розробка автомобільних систем безпеки є складним і трудомістким процесом. Таким чином, провідні компанії галузі можуть допомогти

виробникам комплектного обладнання з генерацією коду для автомобільних систем безпеки за допомогою Motar toolbox, оскільки це може прискорити процес. Подібним чином, визначивши комбінований процес розробки, який інтегрує FuSa, SOTIF і CE, також може надати клієнтам чітко визначений процес розробки для автомобільних систем безпеки. Однак розробка коду для автомобільних систем безпеки за допомогою Motar toolbox може бути виконана лише в тому випадку, якщо Motar toolbox є сумісним як інструмент, що зараз не так. Таким чином, Motar toolbox також має бути сумісним, але це виходить за рамки цього проекту.

Оскільки дослідження в основному зосереджено на інтеграції стандартів ISO у процес розробки системи, функціональність суперсистеми є найбільш актуальною. Як показано на рисунку 1.3 у минулому розробка автомобільних систем безпеки не мала відповідати правилам, методам і вказівкам, визначеним трьома стандартами безпеки ISO. Однак на даний момент процес розробки системи має відповідати вищезазначеним стандартам, але проблема полягає в тому, що сам процес не включає зазначені вказівки. Тому в майбутньому процес розробки системи повинен містити всі аспекти стандартів безпеки, щоб розробка зазначених систем могла бути виконана точно.

1.3. Представлення контекстної діаграми проекту

Ґрунтуючись на результатах інструменту вирішення проблем, розглянутого в підрозділі 1.2 та діаграму контексту проекту також визначено контекст системи. Контекст системи включає в себе межі та різні сутності, які взаємодіють із системою. Щоб підвищити загальну ясність, система та сутності обробляються на діаграмі системного контексту.

Контекстна діаграма є першим і найвищим рівнем деталізації в моделюванні систем. Вона відображає систему як єдиний блок і показує, як ця система взаємодіє зі своїм зовнішнім середовищем. Іншими словами, вона

визначає межі системи і демонструє, які зовнішні фактори впливають на систему і які результати вона виробляє.

Основні елементи контекстної діаграми:

- Система: Представлена як прямокутник з назвою.

- Зовнішні сутності: Інші системи, люди або процеси, які взаємодіють з системою. Представлені прямокутниками з назвою.

- Потоки даних: Стрілки, які показують, як інформація передається між системою і зовнішніми сутностями.

- Мета системи: Короткий опис того, для чого призначена система.

Контекстна діаграма є потужним інструментом для візуалізації і розуміння системи. Вона допомагає забезпечити єдине бачення проекту всім учасникам команди і сприяє успішній розробці програмного забезпечення.

Контекстну діаграму системи можна побачити на рисунку 1.4 де показано різні сутності, які взаємодіють із процесом розробки об’єднаної системи.

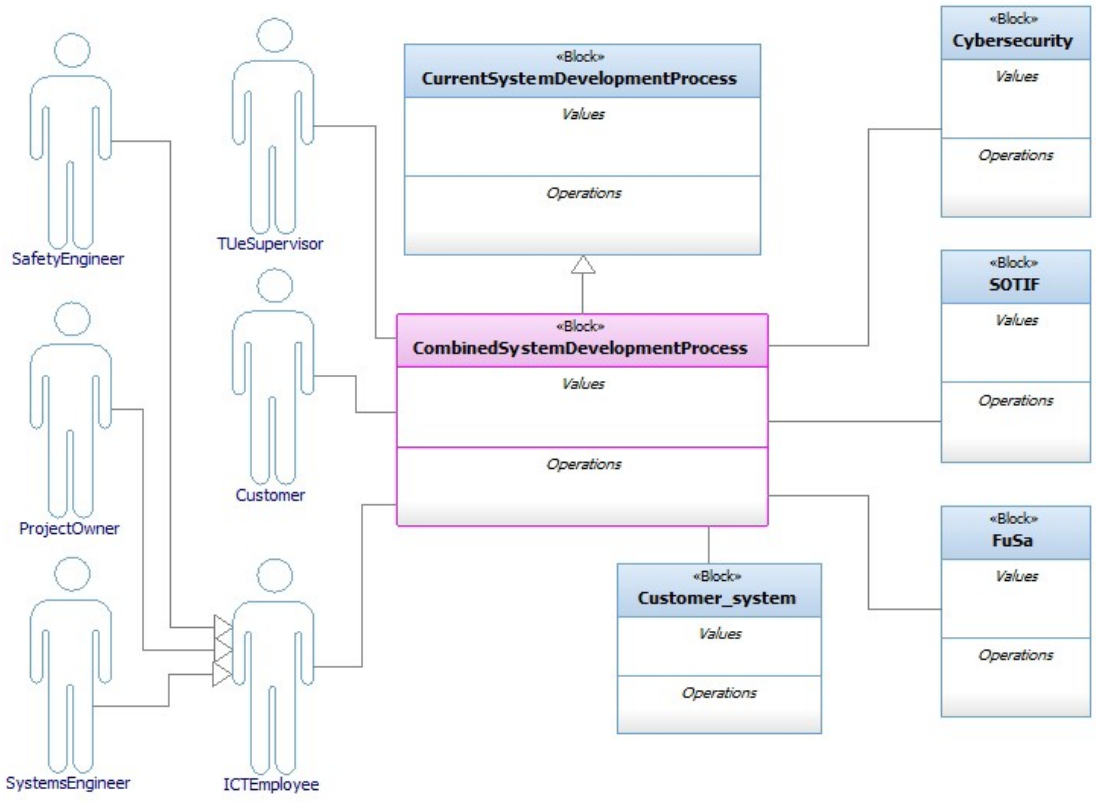

Рис. 1.4. Контекстна діаграма системи

Стандарти ISO, керівник і співробітники є найважливішими суб'єктами, оскільки вони забезпечують значний внесок у процес розробки комбінованої системи. Крім того, клієнт також вказано, оскільки він буде тим, хто зрештою використовуватиме комбінований процес розробки системи. Нарешті, як було зазначено раніше, обсяг проекту спрямований виключно на створення нового комбінованого процесу розробки системи, який відповідає вищезгаданим стандартам, і інтеграції Motar toolbox у цей процес. Таким чином, будь-яка відповідність інструменту Motar toolbox вищезазначеним стандартам безпеки не вважається актуальною для цього проекту.

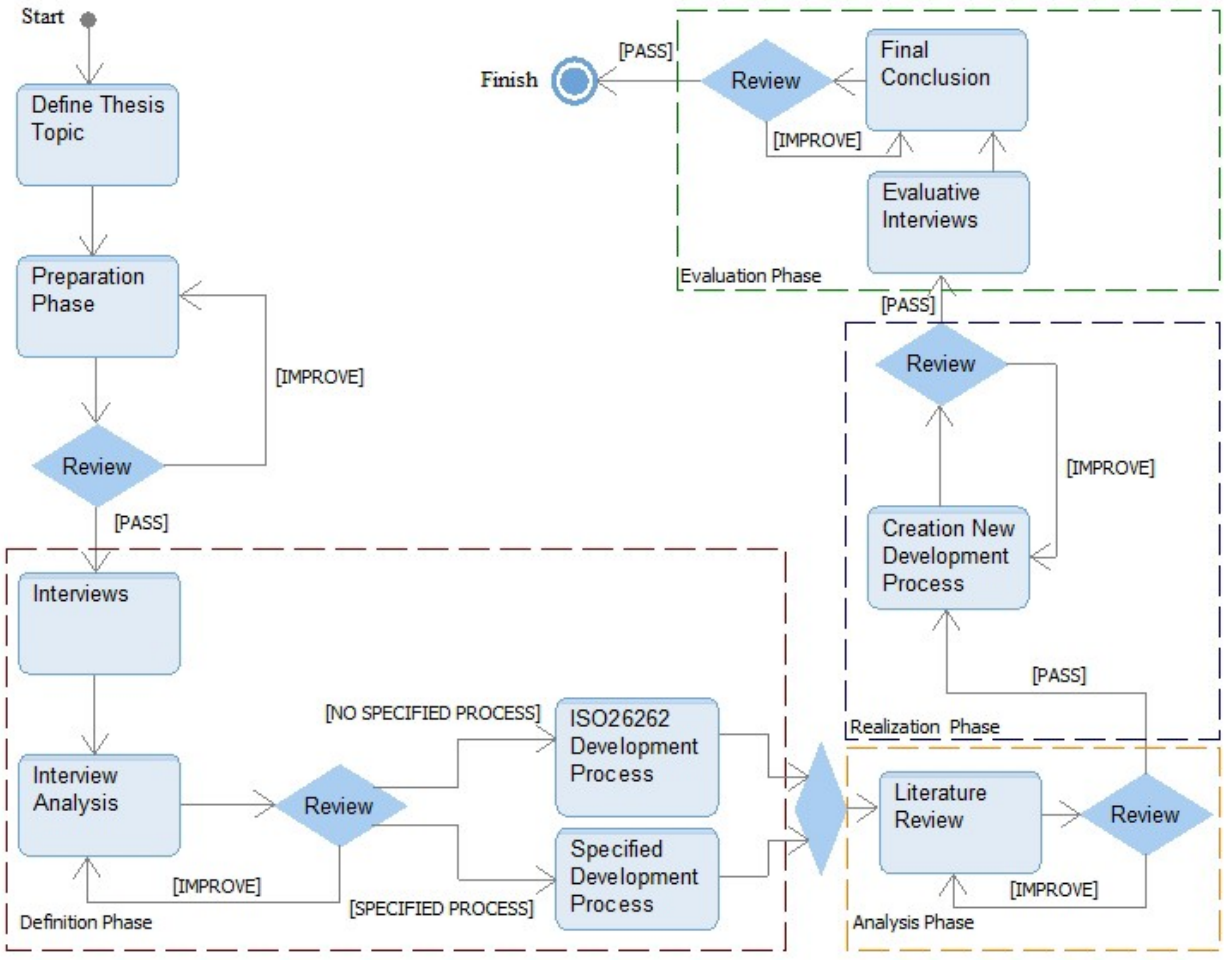

Рис. 1.5. Робочий процес проекту, що містить різні фази проекту та результати

Як було зазначено вище, проект розділений на чотири різні фази. Для кожного етапу є певні результати. Однак результати повинні бути спочатку переглянуті, щоб оцінити, чи відповідають вони поставленим цілям. Якщо це так, то можна переходити до наступного етапу. Робочий процес показано на рисунку 1.5 де представлено перебіг проекту, починаючи від визначення теми до етапу підготовки та закінчуючи етапом оцінювання. Підготовчий етап проводиться перед самим проектом.

Огляд літератури є важливою частиною цього дослідження, тому рекомендується дотримуватися належної методології під час пошуку відповідної літератури.

Етапи проведення огляду літератури:

- Формулювання дослідницького питання: Чітко визначте, на які питання ви хочете отримати відповіді в ході дослідження.

- Вибір джерел: Визначте релевантні бази даних і ключові слова для пошуку літератури.

- Відбір джерел: Виберіть джерела, які найкраще відповідають вашому дослідницькому питанню.

- Аналіз джерел: Проаналізуйте кожне джерело, виділивши ключові ідеї, методи дослідження та результати.

- Синтез результатів: Об'єднайте отриману інформацію і сформулюйте власні висновки.

Щоб знайти відповідну літературу, був використаний наступний підхід [23]:

1. Пошук за ключовими словами

2. Відбір відповідних документів

3. Рецензія на реферат

4. Огляд повної статті

5. Аналіз результатів

Першим кроком було визначення правильних ключових слів. Крім того, використовувалися такі логічні оператори, як І, НЕ та АБО. Другим

кроком був відбір відповідних документів. У цьому випадку статті фільтрувалися за роком публікації, мовою тощо. Після того, як відповідні документи були отримані, необхідно було прочитати анотацію статті, щоб можна було визначити, чи була стаття актуальною чи ні. Якщо це так, то статтю було прочитано повністю, щоб ще раз оцінити її актуальність. Останнім кроком в огляді літератури був аналіз результатів відповідних робіт. Аналіз результатів є важливим, оскільки ці результати в кінцевому рахунку надали інформацію щодо поточного дослідницького проекту.

Висновки до розділу

Проведене в першому розділі дослідження дозволило сформулювати чітке уявлення про предметну область та визначити основні напрямки подальшого дослідження. Описані передумови та інструментарій створюють необхідний фундамент для проведення експериментів та аналізу результатів. Контекстна діаграма, представлена в розділі, наочно демонструє, як розроблювана система вписується в загальний контекст і які фактори впливають на її функціонування. Отримані результати першого розділу мають важливе практичне значення, оскільки дозволяють розробити ефективні та безпечні програмні системи.

РОЗДІЛ 2. МЕТОДИ І СТАНДАРТИ РОЗРОБКИ ПРОГРАМНИХ СИСТЕМ З ТОЧКИ ЗОРУ ЇХ БЕЗПЕКИ

2.1. Дослідження стандартів предметної області

Перед проведенням дослідження було проведено аналіз, який додатково уточнював стандарти FuSa, SOTIF, CE та SAE J3016. Крім того, дослідження також було зосереджено на важливості поєднання FuSa, SOTIF і CE, а також на включенні вищезгаданих стандартів у процес розробки.

2.1.1. Опис стандарту SAE J3016

SAE J3016 «Таксономія та визначення термінів, пов’язаних із системами автоматизації водіння для дорожніх транспортних засобів» [3], спрямований на надання таксономії, яка описує всі рівні автоматизації водіння. SAE J3016 – це міжнародний стандарт, розроблений Об’єднанням автомобільних інженерів (SAE International), який встановлює спільну мову для опису та класифікації систем автоматизації водіння (ADAS) у транспортних засобах. Цей стандарт має на меті забезпечити чітке розуміння рівнів автоматизації, а також визначити ключові терміни та поняття, що використовуються в цій галузі.

Стандарт містить детальні визначення таких термінів, як "автономний автомобіль", "система допомоги водію", "динамічний об'єкт" тощо. Для кожного рівня автоматизації стандарт визначає функціональні вимоги, які повинні бути виконані системою

Стандарт SAE J3016 визначає шість рівнів автоматизації, від 0 (без автоматизації) до 5 (повна автоматизація):

- Рівень 0: Водій повністю контролює транспортний засіб за всіх умов.

- Рівень 1: Система може автоматично виконувати одну з двох функцій: керування рулем або контроль над прискоренням/гальмуванням.

- Рівень 2: Система одночасно виконує дві функції: керування рулем і контроль над прискоренням/гальмуванням, але водій завжди готовий взяти на себе контроль.

- Рівень 3: Система може керувати транспортним засобом у певних ситуаціях, але водій повинен бути готовий взяти на себе контроль за попередженням системи.

- Рівень 4: Система може керувати транспортним засобом у більшості ситуацій, але водій може бути необхідним у деяких випадках.

- Рівень 5: Система повністю керує транспортним засобом за всіх умов без необхідності втручання водія.

Визначення рівня автоматизації виконується відповідно до ролі, яку виконує користувач, система автоматизації водіння та інші компоненти автомобіля [3]. Інфографіка, показана на рисунку 2.1 є прикладом цього.

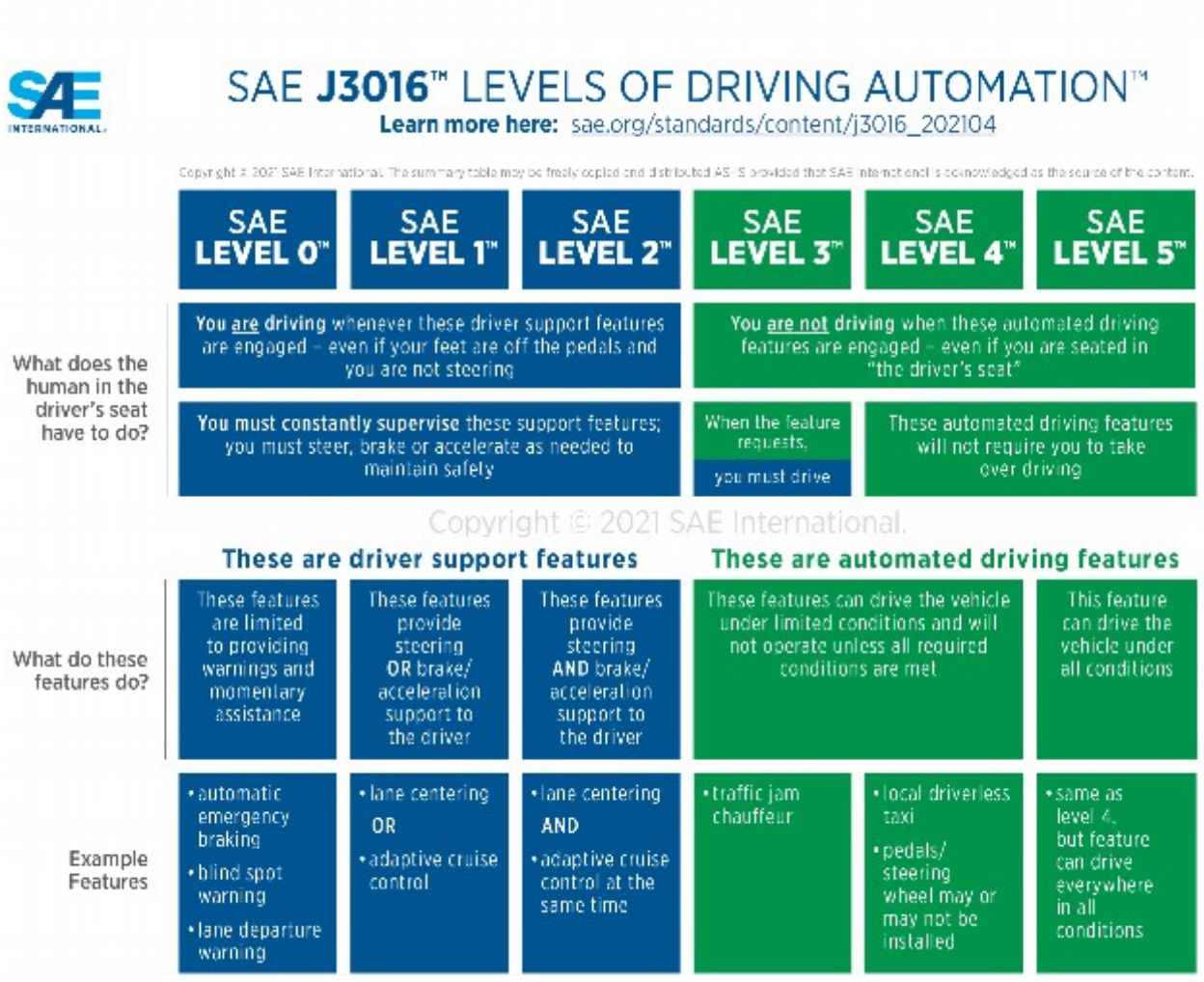

Рис. 2.1. Інфографіка рівнів автоматизації SAE

Як показано на інфографіці, рівень SAE від 0 до рівня SAE 2 вимагає від водія керування автомобілем, тоді як системи рівнів SAE від 3 до 5 беруть на себе керування. Однак існує ще одна чітка відмінність між рівнями SAE 0-2 і рівнями SAE 3-5. Прикладом цієї відмінності є функція повного автономного керування (FSD) Tesla. У цьому випадку назва функції Tesla FSD означає, що, придбавши функцію FSD, автомобіль може їздити самостійно, і тому його слід розглядати як систему SAE рівня 3-5. Однак експерти галузі чітко заявляють, що це вводить в оману, і посилаються на вимоги Tesla, які стверджують, що потрібен активний нагляд водія [24]. Тому за замовчуванням FSD вважається системою SAE рівня 2, оскільки для рівнів 3 і вище активний нагляд не потрібен.

В даний час розробка систем автономного водіння все ще триває. Виробники здатні керувати робототаксі, які оцінюються як автономні системи SAE рівня 4 [24]. Крім того, на даний момент Mercedes є єдиним виробником, який зміг сертифікувати свій варіант автономного водіння під назвою Drive Pilot як систему SAE рівня 3 [26].

Сфера застосування стандарту SAE J3016 обмежена системами автоматизації водіння транспортних засобів, які виконують завдання динамічного водіння (DDT) або принаймні його частину [3]. Крім того, стандарт поширюється на «дорожні» транспортні засоби, які обладнані системами автоматизації транспортних засобів. Однак деякі системи активної безпеки виключаються, оскільки сфера застосування SAE J3016 включає лише системи, які виконують частину або DDT на тривалій основі [3]. Прикладами систем, які виключаються, є електронний контроль стійкості (ESC), система підтримки смуги руху (LKA) і автоматичне екстрене гальмування (AEB).

Стандарт SAE J3016 відіграє важливу роль у розвитку технологій автоматизованого водіння. Він забезпечує спільну мову для всіх учасників ринку, сприяє розробці безпечних і надійних систем, а також сприяє розвитку нормативно-правової бази в цій галузі.

2.1.2. Стандарт функціональної безпеки

FuSa, або стандарт ISO 26262, опублікований ISO, визначає принципи функціональної безпеки для електронних систем для автомобільного застосування. Стандарт забезпечує структуру, яка розглядає пов'язані з безпекою системи на основі інших технологій [7]. Крім того, стандарт:

• Надає довідковий матеріал життєвого циклу безпеки автомобіля, який дає змогу адаптувати дії для кожного етапу життєвого циклу.

• Забезпечує специфічний для автомобіля підхід, заснований на ризиках, який можна використовувати для визначення рівня цілісності автомобільної безпеки (ASIL).

• Використовує ASIL для специфікації вимог ISO 26262, щоб уникнути невиправданих ризиків.

• Забезпечує вимоги до валідації та верифікації і таким чином досягає прийнятного рівня безпеки.

• Забезпечує вимоги до визначення відносин між замовником і постачальником.

Сфера застосування стандарту ISO 26262 в основному обмежена системами, пов'язаними з безпекою, які поєднані принаймні з однією системою E/E. Крім того, стандарт поширюється на системи безпеки, які використовуються у серійному автомобілі [7]. Стандарт використовується OEM-виробниками для усунення можливих небезпек, які можуть виникнути через неправильну роботу системи, пов'язаної з безпекою, або через помилкову взаємодію самої системи. Стандарт розділений на різні етапи, які стосуються розробки продукту, і для кожного етапу необхідно виконати певні дії.

Фази [7]:

• Фаза концепції: фаза концепції починається з визначення елемента, за яким слідує аналіз небезпеки та оцінка ризику (HARA). Під час HARA цілі безпеки повинні бути визначені на основі результатів визначення ASIL. Після визначення цілей безпеки необхідно також розробити Концепцію

функціональної безпеки (FSC). По суті, FSC визначає відповідні вимоги безпеки на основі вищезазначених цілей безпеки. Ці новостворені вимоги безпеки потім розподіляються між відповідними елементами системи.

• Розробка продукту на системному рівні: Розробка продукту на системному рівні починається зі створення Концепції технічної безпеки (TSC). TSC містить технічні вимоги безпеки та архітектурний проект системи. Крім того, розробка продукту на етапі системного рівня також зосереджується на інтеграції та тестуванні системи та елементів, а також перевірці безпеки.

• Розробка продукту на апаратному рівні (HW): Розробка продукту на апаратному рівні зосереджується на специфікації вимог щодо безпеки апаратного забезпечення, а також на дизайні апаратного забезпечення, інтеграції та перевірці апаратного забезпечення.

• Розробка продукту на рівні програмного забезпечення (ПЗ): Розробка продукту на рівні програмного забезпечення зосереджується на специфікації вимог безпеки програмного забезпечення, архітектурному дизайні програмного забезпечення, проектуванні, реалізації, перевірці програмного блоку, а також інтеграції та перевірці програмного забезпечення.

• Виробництво, експлуатація, обслуговування та виведення з експлуатації: Нарешті, етап виробництва, експлуатації, обслуговування та виведення з експлуатації зосереджується на плануванні виробництва, експлуатації, обслуговування та виведення з експлуатації, а також на самому виробництві та експлуатації, обслуговуванні та виведенні з експлуатації.

Функціональна безпека (FuSa) є невід’ємною частиною процесу розробки будь-якої автомобільної електричної та електронної системи для забезпечення безпечної та надійної роботи системи. Таким чином, FuSa має на меті застосування систематичного підходу до виявлення, оцінки та розробки способів пом’якшення ризиків/потенційних небезпек, які можуть виникнути.

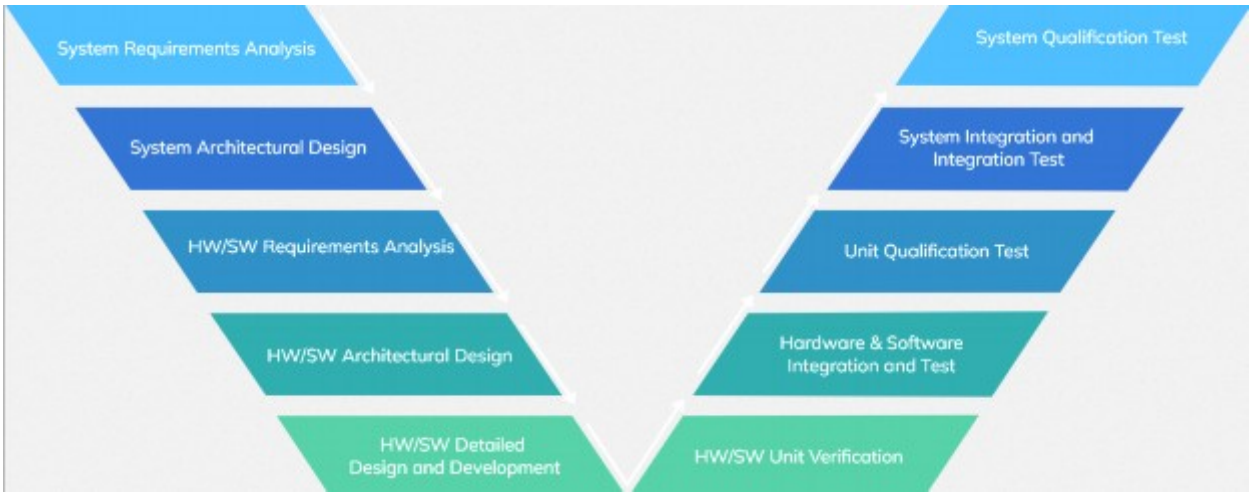

Рис. 2.2. Фази стандарту функціональної безпеки

На початковому етапі дослідження проводиться деталізований аналіз небезпек і ризиків (HARA) з метою ідентифікації та класифікації потенційних загроз. Отримані дані дозволяють визначити рівень автоматизації безпеки (ASIL) в діапазоні від A до D. Далі, за допомогою методів DFMEA та FMEDA здійснюється детальна оцінка можливих відмов та їхніх наслідків. На основі отриманих результатів формується концепція функціональної безпеки, яка визначає конкретні вимоги до системи. Ці вимоги пронизують усі етапи розробки, від системного рівня до розробки програмного та апаратного забезпечення. Для підтвердження відповідності системи вимогам безпеки розробляються та реалізуються комплексні процедури тестування та верифікації. Весь процес розробки базується на вимогах стандарту ISO 26262, що гарантує передбачуваність поведінки системи та її здатність адекватно реагувати на нестандартні ситуації.

2.1.3. ISO 21448: Безпека передбачуваної функціональності

Подібно до FuSa, SOTIF або стандарт ISO 21448 [8] також використовується виробниками та OEM. Однак ключовою відмінністю є те, що SOTIF зосереджується не на небезпеці, яка виникає через несправність систем/компонентів, а на потенційних небезпеках під час запланованої роботи. Тому основна увага приділяється ненавмисному неправильному

використанню водієм, неправильній поведінці через умови навколишнього середовища та обмеження продуктивності самої системи.

Сфера застосування SOTIF обмежена системами, де ситуаційна обізнаність є критичною для забезпечення безпечної роботи. Таким чином, SOTIF зосереджується на системах, які використовують датчики та алгоритми для створення згаданої раніше ситуаційної обізнаності. Особливо системи, які можна віднести до категорії SAE рівні від 1 до 5 підходять [8].

Стандарт SOTIF складається з наступних дій [8]:

• Специфікація та дизайн: Специфікація та дизайн зосереджені на забезпеченні достатньої інформації для здійснення діяльності SOTIF. Крім того, специфікації та дизайн повинні оновлюватися після кожної ітерації.

• Ідентифікація та оцінка небезпек: ідентифікація та оцінка небезпек зосереджуються на визначенні небезпек та визначенні ризиків і сценаріїв, які можуть призвести до шкоди. Крім того, для результуючих залишкових ризиків необхідно визначити критерії прийнятності.

• Ідентифікація та оцінка потенційних функціональних недоліків і потенційних ініціюючих умов: під час ідентифікації та оцінки потенційних функціональних недостатностей і ініціюючих умов необхідно ідентифікувати недостатності та ініціюючі умови, а також визначити ті, що призводять до шкоди. Крім того, відповідь системи повинна бути оцінена на основі прийнятності.

• Функціональні модифікації, що стосуються ризиків, пов’язаних із SOTIF: під час функціональних модифікацій, спрямованих на ризики SOTIF, мають бути визначені та остаточно застосовані відповідні заходи. Крім того, специфікація та дизайн повинні оновлюватися після кожної модифікації SOTIF.

• Визначення стратегії верифікації та валідації: під час визначення верифікації та валідації стратегія повинна бути визначена разом із цілями щодо валідації.

• Оцінка відомих сценаріїв: Під час оцінки відомих сценаріїв необхідно оцінити визначені небезпечні сценарії.

• Оцінка невідомих сценаріїв: під час оцінки невідомих сценаріїв валідація повинна довести, що залишковий ризик відповідає визначеним критеріям прийнятності.

• Оцінка досягнення SOTIF: Оцінка досягнення SOTIF зосереджується на наданні інформації, яка підтверджує, що діяльність SOTIF завершена та правильна. Таким чином, аргумент досягнення повинен бути наданий і остаточно оцінений.

• Діяльність на етапі експлуатації: нарешті, діяльність на етапі експлуатації зосереджена на визначенні процесу моніторингу для забезпечення SOTIF під час роботи.

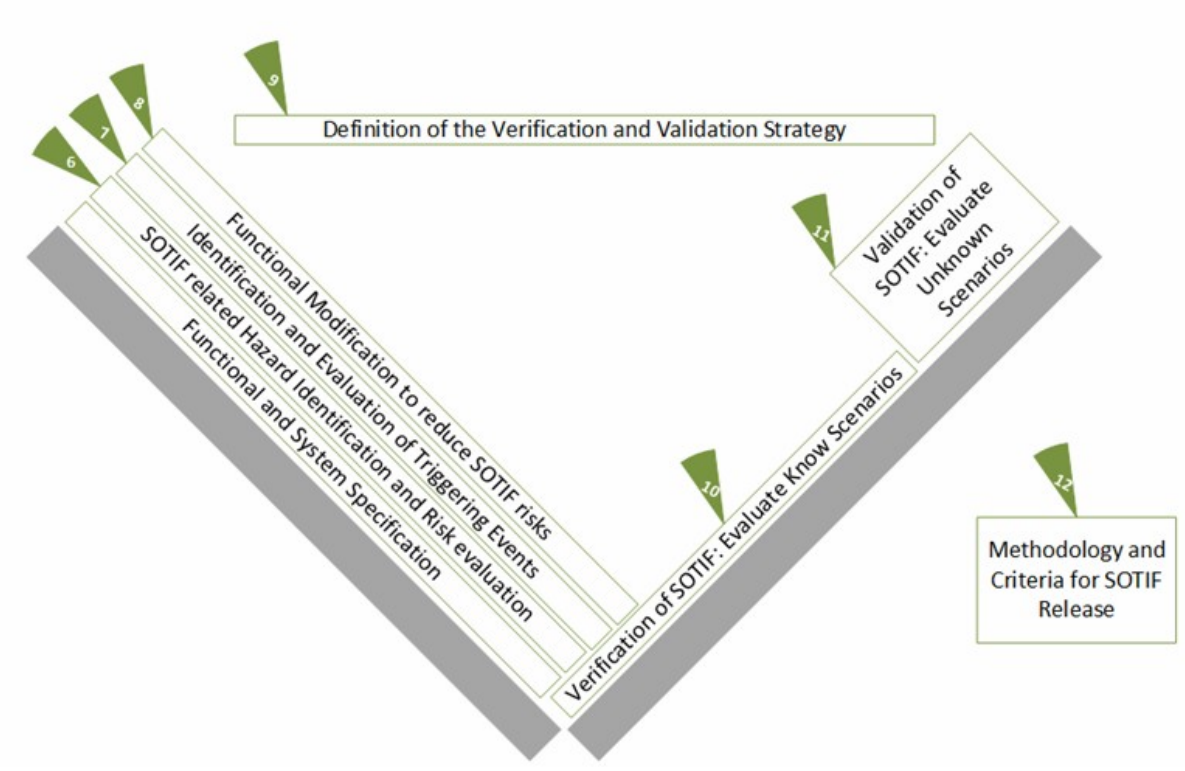

Рис. 2.3. Особливості ISO 21448 – Safety of the Intended Functionality

2.1.4. Стандарт кібербезпеки

Стандарт CE або ISO 21434 можна використовувати для розгляду аспекту кібербезпеки для E/E систем у транспортних засобах [9]. Основна

мета стандарту полягає в тому, щоб гарантувати, що виробники та OEM-виробники можуть розробляти системи, які включають заходи проти навмисних атак. Таким чином, вимоги щодо управління ризиками, розробки продукту, експлуатаційного використання, технічного обслуговування та виведення з експлуатації визначені в самому стандарті.

Сфера застосування стандарту CE обмежена системами E/E для серійних дорожніх транспортних засобів, а також включає будь-які відповідні компоненти та інтерфейси [9].

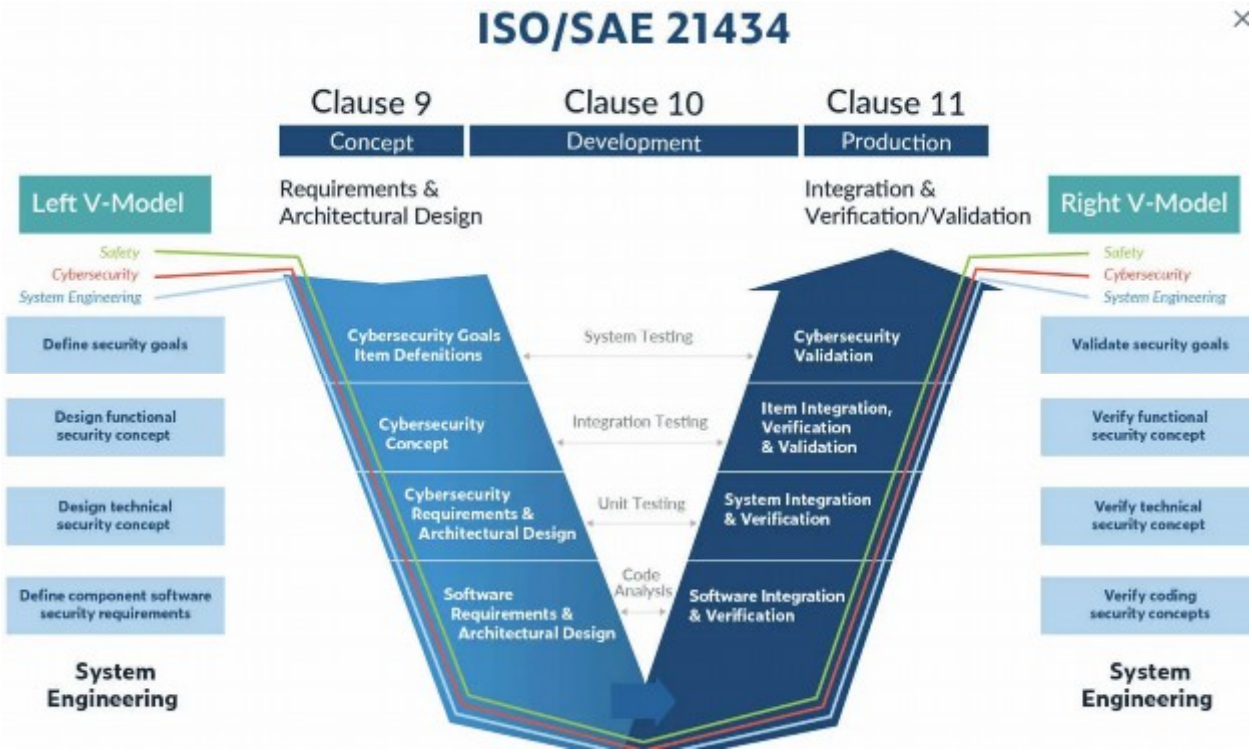

Рис. 2.4. Особливості ISO 21434: Cybersecurity

CE містить загалом 15 пунктів. Однак наступні пункти є актуальними щодо процесу розробки системи [9]:

• **Концепція:** фаза концепції починається з визначення елемента, а потім продовжується аналізом загроз та оцінкою ризиків (TARA) і подальшим визначенням цілей безпеки. Крім того, після визначення цілей безпеки необхідно розробити концепцію кібербезпеки. Концепція

кібербезпеки в основному зосереджена на визначенні вимог ¬щодо кібербезпеки та розподілі вищезазначених вимог кібербезпеки до елемента.

• **Розробка продукту:** Етап розробки продукту в основному зосереджений на розподілі вимог до кібербезпеки для відповідних компонентів. Крім того, мають бути визначені специфікації кібербезпеки. Нарешті, також повинні бути виконані дії з інтеграції та перевірки.

• **Перевірка кібербезпеки**. Етап перевірки кібербезпеки зосереджується на перевірці елемента на рівні транспортного засобу щодо кібербезпеки.

• **Виробництво**: на етапі виробництва охоплюється виготовлення та складання предмета або компонента. Тому необхідно створити план контролю виробництва.

• **Експлуатація та технічне обслуговування**: Етап експлуатації та технічного обслуговування зосереджується на підтримці кібербезпеки і, таким чином, спрямований на визначення можливих інцидентів у сфері кібербезпеки та надання заходів для вирішення цих інцидентів.

• **Припинення підтримки кібербезпеки та виведення з експлуатації**: нарешті, завершення підтримки кібербезпеки та виведення з експлуатації зосереджується на завершенні підтримки щодо кібербезпеки та подальшому виведенні з експлуатації елементів і компонентів.

2.2. Концепція об’єднання стандартів

Концепція об'єднання стандартів ISO 21434, ISO 21448 і ISO 26262 полягає в забезпеченні комплексного підходу до безпеки автомобілів, охоплюючи різні аспекти їхньої розробки та експлуатації. Кожен зі стандартів має свою сферу застосування, але їх поєднання спрямоване на те, щоб створити єдину систему управління ризиками, яка охоплює як функціональну безпеку, так і кібербезпеку, а також запобігання загрозам, які можуть виникнути через некоректну роботу систем автомобіля.

Ось у чому полягає суть кожного стандарту та їх зв'язок:

- ISO 26262 — це стандарт з функціональної безпеки автомобілів. Він зосереджений на забезпеченні того, щоб електронні та електричні системи автомобіля працювали без збоїв і не призводили до аварій чи небезпечних ситуацій. Стандарт охоплює весь життєвий цикл систем: від концепції до розробки, виробництва, експлуатації та утилізації. Його мета — зменшення ризиків, пов'язаних із системними відмовами через проектні помилки чи технічні несправності.

- ISO 21448 — це стандарт, який охоплює безпеку очікуваного функціонування (SOTIF, Safety Of The Intended Functionality). Він зосереджений на запобіганні ситуаціям, коли система працює згідно з її специфікаціями, але все одно може викликати небезпечну ситуацію. Це стосується, зокрема, автономних систем, де ризики можуть виникати через обмеження датчиків або помилкові інтерпретації зовнішніх умов.

- ISO 21434 — це стандарт для кібербезпеки автомобільних систем. Його мета — захистити автомобілі від кібератак, які можуть вплинути на функціонування електронних систем і, як наслідок, поставити під загрозу безпеку пасажирів. Цей стандарт передбачає процеси виявлення загроз, управління вразливостями та захисту від потенційних атак протягом усього життєвого циклу автомобільної системи.

Поєднання цих стандартів дозволяє створити всеосяжну систему управління ризиками, яка охоплює всі критичні аспекти безпеки автомобілів:

- ISO 26262 забезпечує функціональну безпеку систем.

- ISO 21448 доповнює його, зосереджуючись на непередбачуваних загрозах у правильному функціонуванні систем.

- ISO 21434 вводить кібербезпеку, захищаючи від зовнішніх загроз і атак.

Разом ці стандарти допомагають автовиробникам забезпечити комплексну безпеку автомобілів, що особливо важливо для сучасних

автомобілів з інтенсивною електронною автоматизацією та зростаючою кількістю підключених і автономних функцій.

Оскільки всі три стандарти безпеки ISO тепер визначені та описані, можна також обговорити актуальність їх об’єднання в єдиній стратегії розвитку. Оскільки всі три стандарти спеціально зосереджені на різних аспектах безпеки чи захисту, відмінності між сферами застосування можна відобразити на рисунку 2.5 показано область застосування кожного стандарту.

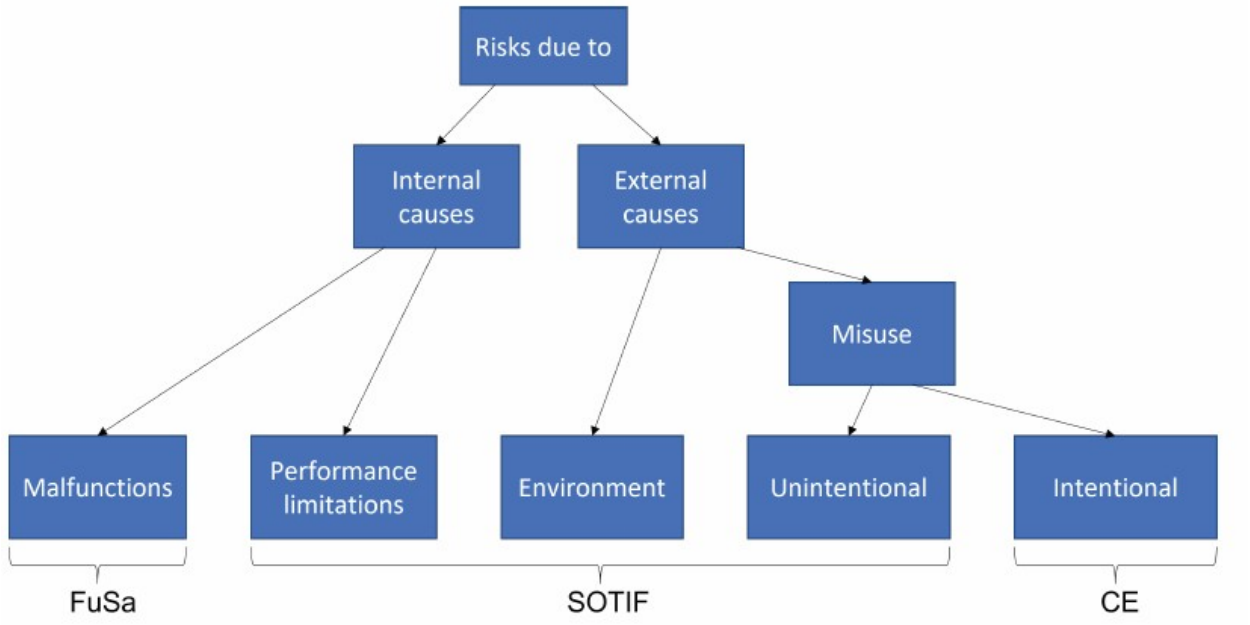

Рис. 2.5. Різниця між сферами застосування стандартів

Як вже було зазначено сфера застосування FuSa зосереджена на можливості несправної поведінки систем E/E. Таким чином, ризики, які можуть виникнути, є внутрішніми, оскільки вони стосуються лише самої системи. Однак сфера застосування SOTIF не обмежується єдиними ризиками, які виникають через внутрішні причини самої системи. Замість цього він також зосереджується на зовнішніх причинах, таких як ненавмисне неправильне використання.

Нарешті, для CE сфера застосування зосереджена на ризиках, які виникають через навмисне неправомірне використання фізичної чи юридичної особи.

Об’єднавши три стандарти ISO в єдиний процес розробки, розробку нових систем безпеки, які відповідають вищезазначеним стандартам, можна спростити.

2.3. Представлення процесу розробки системи

Оскільки дослідження в основному зосереджені на розробці автомобільних систем безпеки, важливо зрозуміти, як ці системи розроблені. Як правило, розробка систем, пов’язаних з автомобілем, виконується відповідно до підходу V-циклу [27].

V-модель (V-цикл) є одним з класичних методів управління процесом розробки програмних систем. Її особливістю є ітераційний та послідовний підхід, де етапи розробки системи поділені на дві основні гілки: ліву частину, яка стосується етапів проєктування (декомпозиції), та праву частину, яка охоплює процеси валідації та тестування (верифікації). Назва "V-модель" походить від того, що етапи розміщуються у формі літери "V". Нижче наведено основні особливості V-моделі для розробки програмних систем:

1. Послідовність етапів

- *Ліва частина V-моделі: розробка та проєктування*

- Етапи на лівій стороні стосуються аналізу вимог, проєктування архітектури та специфікацій. Вони включають:

- Аналіз вимог (системних, функціональних та нефункціональних).

- Системне проєктування: визначення загальної архітектури системи.

- Детальне проєктування: опис функцій на рівні компонентів та модулів.

- *Права частина V-моделі: верифікація та тестування*

- Кожен етап проєктування на лівій стороні має відповідний етап тестування на правій стороні:

- Тестування компонентів: верифікація правильності функцій на рівні компонентів.

- Інтеграційне тестування: перевірка взаємодії між компонентами системи.

- Системне тестування: тестування всієї системи на відповідність вимогам.

- Тестування на відповідність вимогам користувача: перевірка відповідності очікуванням та потребам користувачів.

2. Раннє виявлення дефектів

Оскільки кожен етап проєктування на лівій стороні має відповідний етап верифікації на правій, це дозволяє виявляти помилки ще на ранніх стадіях розробки. Це сприяє зниженню витрат на виправлення дефектів на пізніх етапах, адже чітке визначення вимог та проєктування гарантує меншу кількість помилок при реалізації.

3. Ясна взаємозалежність етапів

Кожен етап на лівій стороні V-моделі безпосередньо відповідає певному етапу на правій стороні. Наприклад, системні вимоги на початку розробки перевіряються на кінцевому етапі за допомогою тестування системи. Це забезпечує логічний зв'язок між аналізом вимог і тестуванням.

4. Сильний фокус на верифікацію та валідацію

Однією з основних особливостей V-моделі є її орієнтація на верифікацію (перевірка правильності розробки згідно специфікацій) та валідацію (перевірка, чи відповідає система вимогам користувача). Це допомагає забезпечити високу якість розробленої системи, гарантуючи, що вона відповідає як технічним, так і користувацьким вимогам.

5. Підтримка чіткої документації

Кожен етап розробки та тестування має свої конкретні вимоги до документації. V-модель передбачає створення чітких специфікацій, які пізніше використовуються для верифікації та тестування на кожному рівні.

6. Відсутність гнучкості до змін

Однією з основних слабких сторін V-моделі є її низька гнучкість. Оскільки V-модель є водоспадною методологією, внесення змін на пізніх етапах розробки є складним і часто дорого коштує. Будь-які зміни в вимогах призводять до необхідності переробки як проєктування, так і тестування, що може суттєво вплинути на час і бюджет проєкту.

7. Придатність для критично важливих систем

V-модель ідеально підходить для розробки критично важливих систем, де безпека і надійність мають першочергове значення (наприклад, в авіації, автомобільній промисловості, медицині тощо). Вона дозволяє точно відслідковувати виконання всіх вимог і тестування на кожному етапі.

8. Сильний фокус на контроль якості

Через постійний процес тестування та верифікації на всіх етапах, V-модель забезпечує високий рівень контролю якості, що знижує ризики невідповідності кінцевого продукту вимогам і підвищує надійність системи.

Стандартний V-цикл починається зі специфікації вимог. Після визначення вимог систему можна проектувати. Як тільки це буде зроблено, компоненти системи також повинні бути спроектовані. Потім цей крок супроводжується впровадженням усіх компонентів. Заключні кроки V-циклу зосереджені спочатку на тестуванні окремих блоків, а зрештою — на тестуванні всієї системи. Базовий V-цикл також може бути розроблений у новий процес розробки системи, який дозволяє включати процеси безпеки. Приклад цього можна побачити на рисунку 2.6, на якому показано (функціональні) процеси безпеки для лівої сторони V-циклу.

Як показано на рисунку 2.6 різні стандарти, такі як FuSa в цьому випадку, адаптуються до V-циклу. Крім того, стандарти надають рекомендації щодо проектування системи, розробки, проектування

апаратного та програмного забезпечення та інтеграції компонентів [27]. Однак більшість стандартів застосовуються лише до певних компонентів і функцій, які передбачають, що водій усе ще контролює транспортний засіб [28]. Для високоавтоматизованих систем автомобіля необхідно створити окремий безпечний конвеєр розробки, а також оцінити загальну продуктивність перед виробництвом. Тому було запропоновано, щоб автомобільна безпека була впроваджена в робочий процес розробки, починаючи від створення концепції аж до валідації та тестування системи [28].

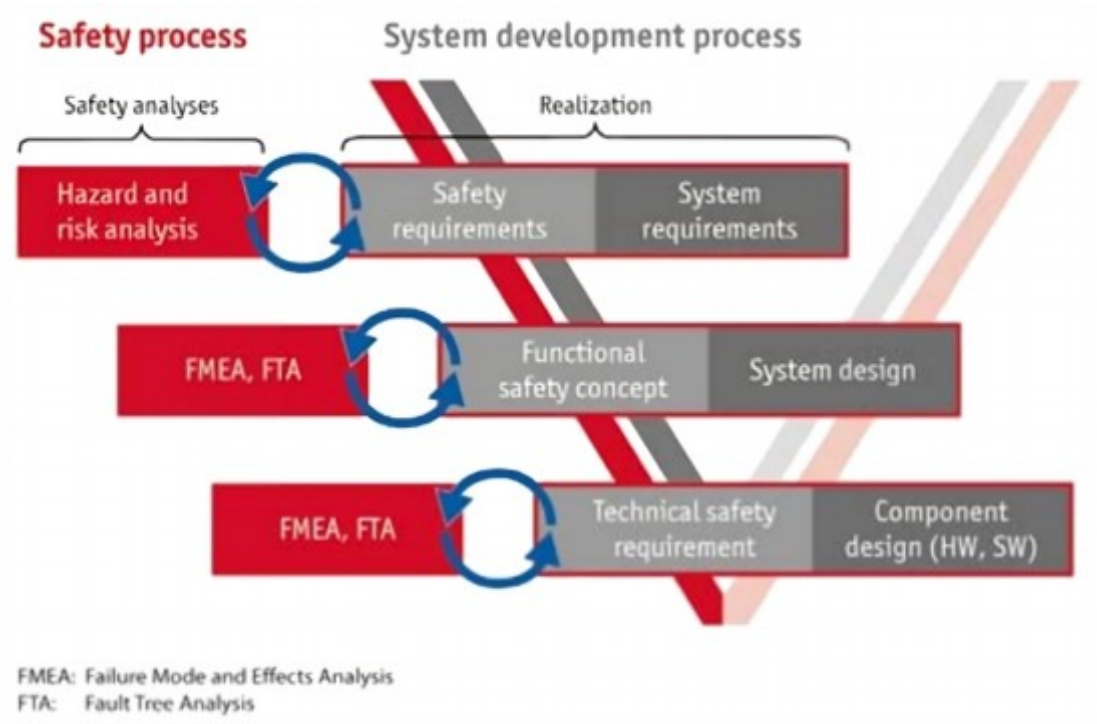

Рис. 2.6. Приклад V-моделі, яка включає процеси безпеки

Інша потенційна проблема щодо розвитку автомобільних систем виникає із запровадженням стандарту CE. Цей стандарт спрямований на вирішення теми кібербезпеки під час розробки автомобільних систем. Однак, оскільки CE спрямована на інтеграцію діяльності для кожної фази життєвого циклу, належна інтеграція розглядається як необхідна [30]. Інтеграція діяльності CE Разом із діяльністю FuSa щодо безпеки, це те, що дослідники

також досліджували. Наприклад, було підкреслено необхідність системного підходу до CE та FuSa, оскільки розширення аналізу FuSa та методів та інструментів, пов’язаних з розробкою, було визнано недостатнім [31]. Натомість автори запропонували інтегрований підхід безпеки та безпеки та підкреслили наявні можливості. Подібним чином дослідники також мали на меті проаналізувати як стандарти FuSa, так і CE, а також потенційну взаємодію між ними [32]. За словами авторів, потенційна синергія між процесами розробки може призвести до підвищення загальної якості та скорочення часу виходу на ринок, що додатково підкреслює важливість цього дослідження.

Висновки до розділу

Отже, попереднє дослідження було спрямоване на детальне вивчення стандарту SAE J3016 та стандартів ISO, таких як FuSa, SOTIF і CE. Результати дослідження підкреслили необхідність системного підходу, що інтегрує процеси, пов'язані з CE, із процесами функціональної безпеки FuSa. Також було зроблено висновок, що впровадження стандартів ISO в процес розробки систем може сприяти скороченню часу виходу продукції на ринок і підвищенню загальної якості. Дослідження додатково акцентувало увагу на важливості інтеграції стандартів ISO у всі етапи розробки систем, що забезпечує їх комплексну реалізацію.

РОЗДІЛ 3. МОДЕЛІ ТА МЕТОДОЛОГІЇ РОЗРОБКИ ПРОГРАМНИХ СИСТЕМ НА ОСНОВІ КОНЦЕПЦІЙ БЕЗПЕКИ ТА ЗАХИСТУ

3.1. Принцип використання стандартів для розробки системи

Перший варіант, який розглядається це об’єднання FuSa та CE. Тому результати поділяються на три категорії, а саме:

- включення FuSa та CE у процес розробки;
- поєднання підходів HARA і TARA;
- методи, які поєднують безпеку та захист.

Включення FuSa та CE у процес розробки

Процес розробки, який включав діяльність як FuSa, так і CE, був «підходом спільної розробки» [32]. У цьому випадку підхід спільної розробки гарантує, що атрибути FuSa та CE аналізуються одночасно, а не окремо. Перевагою підходу спільної розробки є той факт, що взаємодія між безпекою та захистом розглядається на кожному етапі життєвого циклу [32]. Це потенційно може підвищити загальну повноту усунення всіх ризиків і, таким чином, зменшити ймовірність необхідності внесення коригувань у подальшому. Підхід спільної розробки показано на рисунку 3.1.

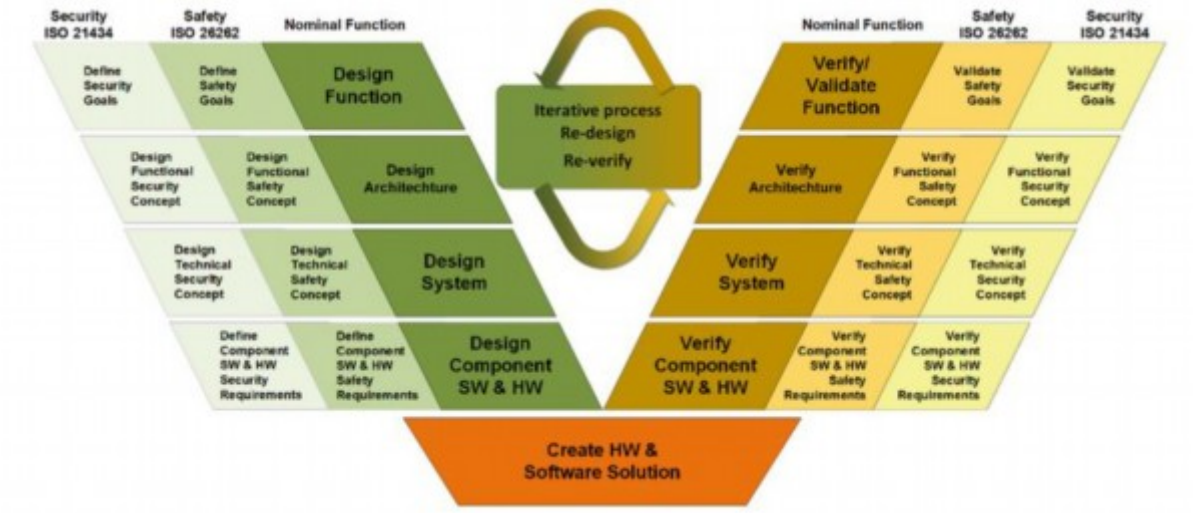

Рис. 3.1. Підхід спільної розробки

Підхід спільної розробки показує діяльність як FuSa, так і CE. Однак для лівої сторони V-циклу перекриття між стандартами було обмежено з точки зору методів аналізу [32]. Замість цього було запропоновано, щоб дії слід виконувати паралельно, і таким чином також враховувати взаємодію.

Як показано на рисунку 3.1 показано, що підхід спільної розробки починається з визначення цілей безпеки та захисту. Однак перед визначенням цілей безпеки та захисту спочатку було виконано HARA та TARA. Під час виконання HARA та TARA взаємодія між FuSa та CE ще не розглядалася, оскільки початковий обсяг не враховував аналіз впливу між двома проблемами [32]. Натомість оцінки обох стандартів проводилися окремо, але одночасно один з одним. Однак можна поєднати обидва підходи або безпосередньо, або за допомогою використання спеціальних методів, які поєднують безпеку та захист.

Взаємодія між FuSa та CE стає очевидною, коли визначено вхідні дані для TARA та HARA [32]. Що стосується TARA, ці дані можуть включати ідентифіковані активи, серед інших прикладів. Активи – це об’єкти, які мають властивості, пов’язані з кібербезпекою, які, у свою чергу, можуть призвести до сценарію загрози, якщо вони скомпрометовані [9]. Безпека може стати активом, якщо вона викликає занепокоєння для організації , і, отже, створюється залежність між безпекою та захистом [32]. Залежність між безпекою та захистом знаходить свій шлях до цілей безпеки та цілей безпеки. Крім того, це також переноситься на інші етапи проектування, оскільки вимоги, пов’язані з безпекою, тепер залежать від виконання вимог безпеки [32].

Для правої сторони підходу спільної розробки є більше можливостей для співпраці, оскільки і FuSa, і CE не потребують спеціальних методів тестування [32]. Натомість обидва методи вимагають використання методів тестування, які можуть забезпечити загальну якість продукту. Крім того, переваги запропонованого підходу спільної розробки можна розділити на три основні напрямки, а саме [32]:

- Тестове середовище: транспортний засіб, повний макет автомобіля, і так далі.
- Цілі тестування: ефективність механізмів, правильність виконання специфікації, надійність тощо.
- Техніки тестування: тестування на основі вимог, послідовне тестування, фази тестування тощо.

З точки зору середовищ тестування, спільне проектування дозволить повторно використовувати вказані середовища для різних цілей, тобто для перевірки та перевірки безпеки та захисту. Щодо цілей тестування, які спрямовані на виявлення систематичних помилок, велике перекриття полегшує спільну верифікацію. Це перекриття можна збільшити, якщо підвищити компетентність тестувальників. Подібним чином, для методів тестування, компетентність тестувальників і засвоєні уроки можуть допомогти збільшити збіги.

Поєднання підходів HARA і TARA

Перший підхід щодо поєднання HARA і TARA називається «THARA» [33]. HARA спрямована на ідентифікацію та класифікацію будь-якої небезпечної події в результаті несправної поведінки відповідного елемента [33]. Подібним чином автори зазначили, що основною метою TARA є виявлення загроз та оцінка ризиків, які з ними пов’язані. Незважаючи на те, що цілі двох підходів були різними, існувала можливість певною мірою поєднати обидва підходи, і тому був розроблений підхід THARA. По суті, підхід THARA поділяється на два різні сценарії, а саме:

- Сценарій 1: Від безпеки до захисту;
- Сценарій 2: Від захисту до безпеки.

У сценарії 1 коефіцієнт керованості оцінюється на основі інформації про атаку безпеки, яка визначається в підході TARA [33]. Навпаки, сценарій 2 вимагає оцінки фактора серйозності TARA на основі визначеного

пов’язаного рівня ASIL активу, який визначено в HARA. Блок-схеми обох сценаріїв відображено на рисунку 3.2.

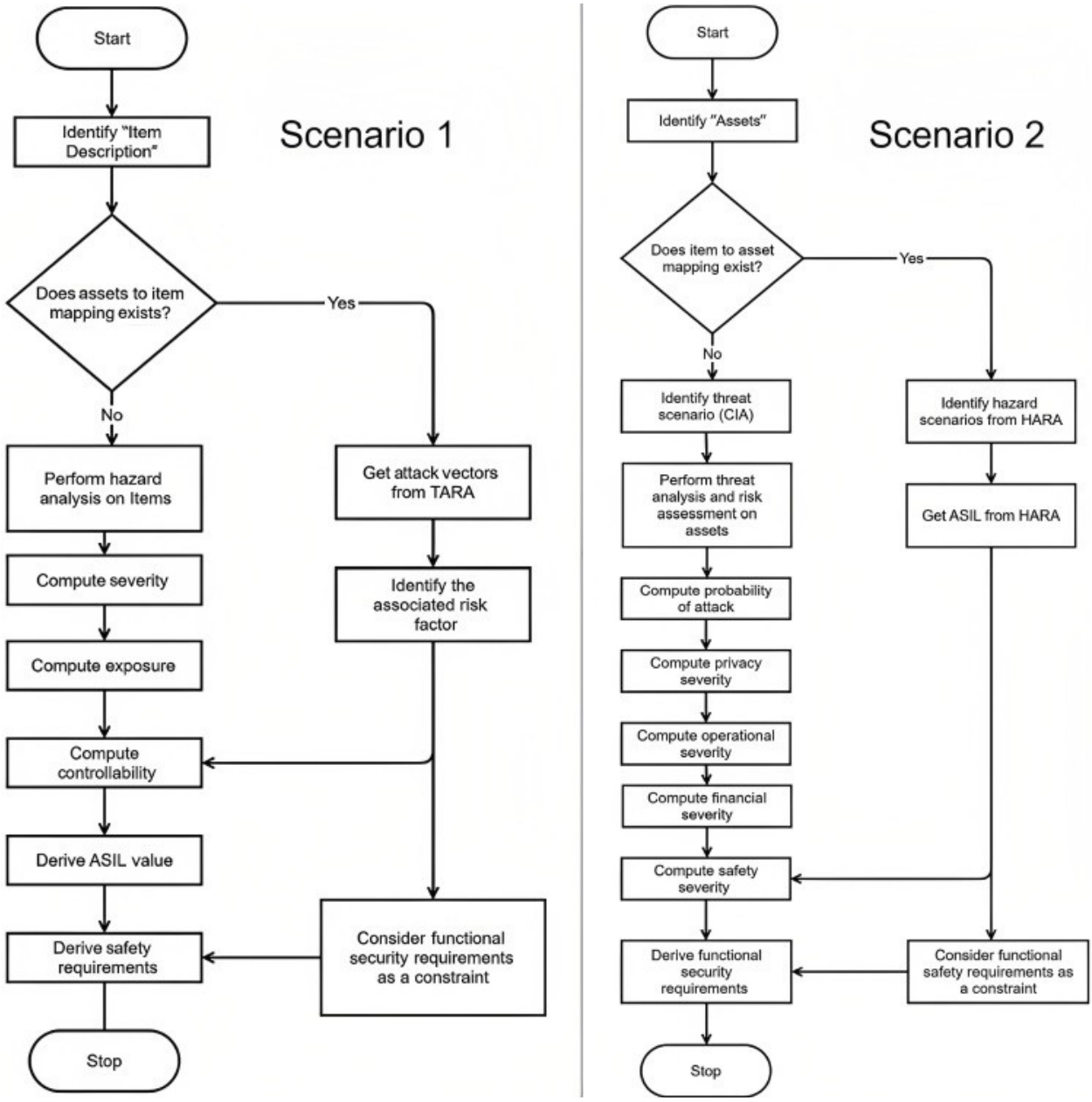

Рис. 3.2. Блок-схеми сценарію 1: від безпеки до захисту та сценарію 2: від захисту до безпеки

Як показано на рисунку 3.2, сценарій 1 обертається навколо оцінки впливу інцидентів безпеки на потенційні загрози безпеці. Подібним чином

сценарій 2 базується на оцінці впливу потенційних загроз безпеці на ризик кібербезпеки.

Однак існують деякі обмеження підходу THARA, а саме:

• Цей підхід вимагає, щоб звіти HARA та TARA були доступні протягом концептуальної фази обох життєвих циклів стандартів.

• Щоб виконати аналіз ризиків підходу THARA, необхідно мати відповідність між елементами HARA та активами TARA.

• У підході THARA коефіцієнт керованості, отриманий від HARA, спирається на відомі вектори загроз. Для забезпечення його ефективності потрібне повне відображення предметів і активів, яке має бути легкодоступним.

• Підхід THARA визначає коефіцієнт серйозності безпеки загрози на основі ASIL відповідного елемента безпеки. Однак не було запропоновано жодного конкретного методу для вирішення цього завдання. Отже, використання формальної моделі стає необхідним.

Методи, які поєднують безпеку та захист

Замість прямого поєднання HARA та TARA існує також можливість вибору відповідного методу для інтеграції безпеки та захисту. І HARA, і TARA залишають відкритою можливість вибору відповідного методу для системи/елемента, що розробляється. Наприклад, типовими методами, що використовуються для HARA, є дослідження небезпеки та працездатності (HAZOP) і аналіз режиму та наслідків відмови (FMEA). Крім того, прикладом типового методу TARA є оцінка критичних операційних загроз, активів і вразливостей (OCTAVE). Однак, як було зазначено раніше, існують методи, які поєднують аспекти безпеки та безпеки в окремих областях [34]. Візуальне представлення класифікації методів HARA (Hazard Analysis and Risk Assessment) і TARA (Threat Analysis and Risk Assessment), а також комбінованих методів можна побачити на рисунку 3.3.

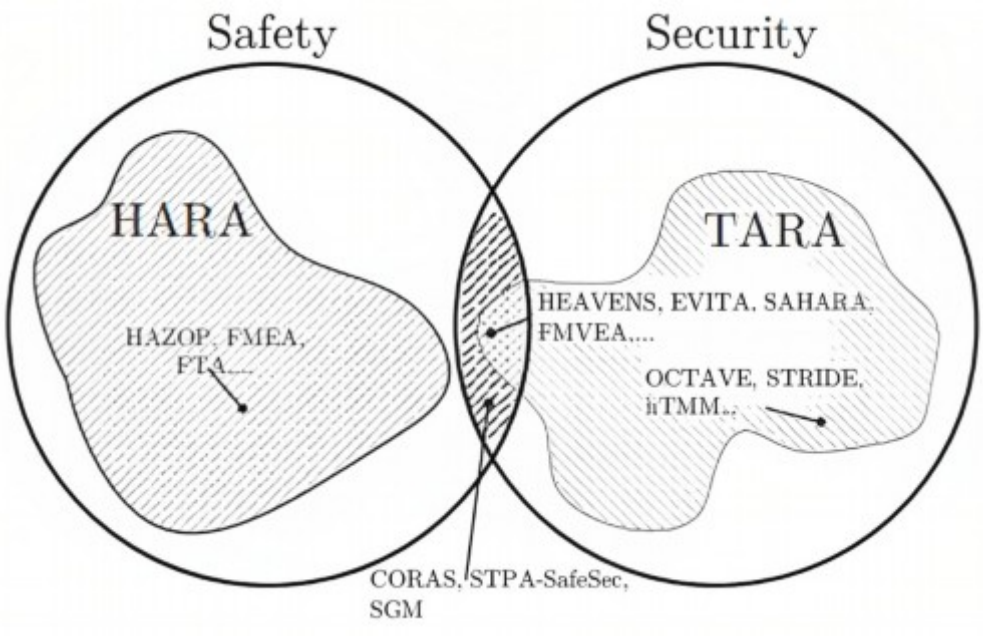

Рис. 3.3. Візуальне представлення різних методів для HARA і TARA

Як показано на рисунку 3.3 такі методи, як HAZOP і OCTAVE, призначені виключно для безпеки та безпеки, класифікуються відповідно. Крім того, також згадуються комбіновані методи, такі як аналіз небезпек і оцінка ризиків з урахуванням безпеки (SAHARA), усунення вразливостей для підвищення безпеки та безпеки програмного забезпечення (HEAVENS) і захищені програми електронної безпеки від проникнення в транспортні засоби (EVITA). Ці методи по суті є частиною методів моделювання загроз TARA, але також стосуються певних частин безпеки [34]. Практичним прикладом використання такого комбінованого методу є оцінка загроз кібербезпеці та загроз безпеці за допомогою методу SAHARA для системи Electrical Steering Column Lock (ESCL) [31].

На рисунку 3.3 показано, що існують додаткові методи, які об’єднують як аспекти безпеки, так і безпеки, такі як CORAS і Systems Theoretic Process Approach-Safety and Security (STPA-SafeSec). Ці методи спрямовані на виконання спочатку аналізу небезпеки, а потім – аналізу загроз безпеці [34]. Однак у вищезгаданих методів є недолік, оскільки вони складні. Крім того, і CORAS, і STPA-SafeSec вимагають певних кроків, перш ніж їх можна буде інтегрувати в поточний аналіз безпеки [34]. Зрештою, такі методи, як

CORAS і STPA-SafeSec, не є частиною ані HARA, ані TARA, що показано на рисунку 3.3 , і тому відкидаються для цього конкретного дослідження.

Стосовно методів TARA, які також включають аспекти безпеки, різноманітність зазначених методів було розглянуто в попередніх дослідженнях [35]. Виходячи з результатів, найбільш відповідними методами були SAHARA, EVITA, HEAVENS та аналіз бінарного ризику (BRA). Однак BRA не є ані оцінкою ризику загрози, ані технікою виявлення загроз. Однак це необхідно для TARA на ранніх стадіях розвитку. Зрештою результати перевірки були оброблені і подані в таблиці 3.1, в якій висвітлено переваги та недоліки кожного методу.

Таблиця 3.1.

Таблиця з переліком переваг і недоліків комбінованих методів

Method:	Advantage(-s):	Disadvantage(-s):
SAHARA	• Easy classification of threats • ASIL aligned classification which makes it ideal for combined safety and security engineering • Originated from HARA and STRIDE thus focuses on safety but adds security evaluation	• Strong relation with safety engineering • Possibility to overlook multi-stage attacks • Lacks quantification for car fleet attacks
EVITA	• Suitable for concept evaluation • Proper separation of the security threat consequences	• Classification of (non) safety-related threats differs • Severity factor is not classified according to FuSa guidelines • Requires too many details for proper classification
HEAVENS	• Additional support for threat scenario estimation by means of STRIDE approach • Not as much classification effort is required compared to EVITA	• Significant discussion potential for individual factors of a single threat • Security Level assessment and analysis for each threat requires vast amounts of work input
BRA	• Easy classification in general	• Not applicable for TARA • Methodology is not entirely dedicated to risk management • Lack of structured threat scenario estimation

На підставі висновків, зазначених у таблиці 3.1 підхід BRA можна розглядати як найменш сприятливий варіант. Здебільшого це пов’язано з тим, що він не застосовувався до TARA, про яку також згадувалося раніше. Що стосується інших методів, кожен із трьох методів, що залишилися, має свої сильні та слабкі сторони, як показано в таблиця 3.1.

3.2. Використання підходу об’єднання стандартів функціональної безпеки та захисту

Наступним варіантом, який необхідно розглянути, є можливість об’єднання FuSa та SOTIF. У цьому випадку результати поділяються на дві категорії, а саме:

- Включення FuSa та SOTIF у процес розробки;
- Поєднання підходів HARA FuSa та SOTIF.

Включення FuSa та SOTIF у процес розробки

Загальний життєвий цикл безпеки на основі об'єднання стандартів FuSa (ISO 26262) і SOTIF (ISO 21448) являє собою комплексний процес, який охоплює всі етапи розробки та експлуатації автомобільних систем. Це забезпечує не лише функціональну безпеку систем, а й запобігання ризикам, що можуть виникнути через обмеження або помилки в інтерпретації навколишнього середовища.

Етапи життєвого циклу безпеки на основі об'єднання FuSa і SOTIF:

1. Концептуальна фаза:

- Оцінка функціональних вимог системи для визначення критичних елементів безпеки згідно з FuSa.

- Аналіз потенційних загроз і небезпек, пов'язаних із очікуваною функціональністю (SOTIF).

- Розробка Safety Goals (цілей безпеки) для FuSa і SOTIF objectives (цілей SOTIF), що враховують і функціональні ризики, і ризики від помилкового функціонування.

2. Аналіз небезпек і оцінка ризиків:

- FuSa: Ідентифікація відмов систем, аналіз впливу цих відмов на безпеку, визначення рівня ASIL (Automotive Safety Integrity Level) для кожної небезпеки.

- SOTIF: Виявлення потенційних ситуацій, коли система функціонує правильно за специфікацією, але може викликати небезпеку, наприклад, через неправильну інтерпретацію оточення або обмеження сенсорів.

3. Стадія розробки:

- FuSa: Проєктування системних компонентів з урахуванням заходів для запобігання відмовам і зменшення їх впливу (апаратні та програмні рішення, архітектурні стратегії).

- SOTIF: Забезпечення очікуваної функціональності через вдосконалення алгоритмів, датчиків, механізмів тестування та валідації, які дозволяють знизити ризики, пов'язані з правильним, але небезпечним функціонуванням.

4. Тестування і верифікація:

- FuSa: Тестування систем на відповідність вимогам безпеки, валідація протоколів виявлення відмов, використання методологій моделювання для оцінки відмов у реальних умовах.

- SOTIF: Тестування систем у нестандартних умовах (corner cases) для перевірки коректності роботи в складних сценаріях, що можуть виходити за межі специфікації.

5. Виробництво та впровадження:

- FuSa: Включення процесів контролю якості на етапах виробництва для забезпечення функціональної безпеки.

- SOTIF: Забезпечення коректності калібрування систем і відповідності розроблених рішень реальним умовам експлуатації.

6. Експлуатація та підтримка:

- FuSa: Постійний моніторинг системи в реальних умовах, оновлення компонентів за необхідності для підтримки високого рівня безпеки.

- SOTIF: Підтримка та вдосконалення систем на основі отриманих даних під час експлуатації, зокрема додаткові оновлення програмного забезпечення для покращення роботи в специфічних ситуаціях.

7. Аналіз інцидентів і покращення:

- В обох стандартах передбачено аналіз будь-яких інцидентів або небезпечних ситуацій, які сталися під час експлуатації. Це дозволяє вносити зміни в систему та знижувати ризики в майбутньому, забезпечуючи безперервний цикл покращення.

Взаємодія FuSa та SOTIF:

- FuSa (ISO 26262) забезпечує безпеку на основі виявлення та управління відмовами систем, зокрема, електронних і електричних компонентів.

- SOTIF (ISO 21448) розширює цей підхід, додаючи аналіз непередбачуваних небезпек, які виникають навіть при правильній роботі системи.

Поєднання цих стандартів дозволяє не тільки керувати відмовами, а й запобігати ситуаціям, коли система функціонує згідно зі специфікацією, але все ще може спричинити небезпеку через обмеження в сенсорах, програмному забезпеченні або умовах експлуатації.

Стосовно комбінації FuSa та SOTIF було зазначено, що немає чіткої різниці між двома стандартами [36]. Тому замість того, щоб уточнювати визначення, щоб досягти зазначеної відмінності, було запропоновано підхід, який об’єднує аспекти в єдиний життєвий цикл. Це призвело до розробки життєвого циклу розробки, який включав аспекти як FuSa, так і SOTIF [36].

Цей новий життєвий цикл розробки під назвою «Загальний життєвий цикл безпеки» зрештою поєднав робочі продукти обох стандартів ISO. Робочі продукти загального життєвого циклу безпеки, а також вихідні робочі продукти FuSa та SOTIF визначені в таблиці 3.2.

Таблиця 3.2.

Перелік робочих продуктів узагальненого життєвого циклу безпеки

FuSa:	**SOTIF:**	**Generalized Safety Life-cycle:**
	Work products:	
Item Definition	Functional and System Specification	Item Definition
HARA	Hazard Identification Hazard Analysis Risk Evaluation Specification of Validation Target	HARA
	Identification of Triggering Events Acceptability of Triggering Events	Triggering Event Analysis
Functional Safety Concept	Functional modifications to reduce SOTIF risks	Functional Safety Concept
	Definition of V & V Strategy	SOTIF V & V Strategy
System Verification Report	SOTIF Verification Report	System Verification Report
Validation Plan		Validation Plan
Technical Safety Concept		Technical Safety Concept
System Design Specification		System Design Specification
Safety Analysis Report		Safety Analysis Report
Integration Testing Plan and Report		Integration Testing Plan and Report
Validation Plan and Report	SOTIF Validation Report	Validation Plan and Report
FuSa Assessment Report		Safety Analysis Report
Release for Production Report	SOTIF Release Report	Safety Analysis Report

Замість того, щоб виконувати дії FuSa та SOTIF окремо, узагальнений життєвий цикл безпеки [36] можна використовувати, оскільки він інтегрує життєві цикли як FuSa, так і SOTIF, і це робить SOTIF більш зрозумілим загалом.

Однак у узагальненого життєвого циклу безпеки було два недоліки, оскільки він зосереджувався виключно на поведінці на рівні системи, а по-друге, він застосовувався лише до ADAS [37]. Натомість було запропоновано

новий робочий процес для FuSa та SOTIF, який показано на рисунку 3.4, який охоплював усі архітектурні рівні [37].

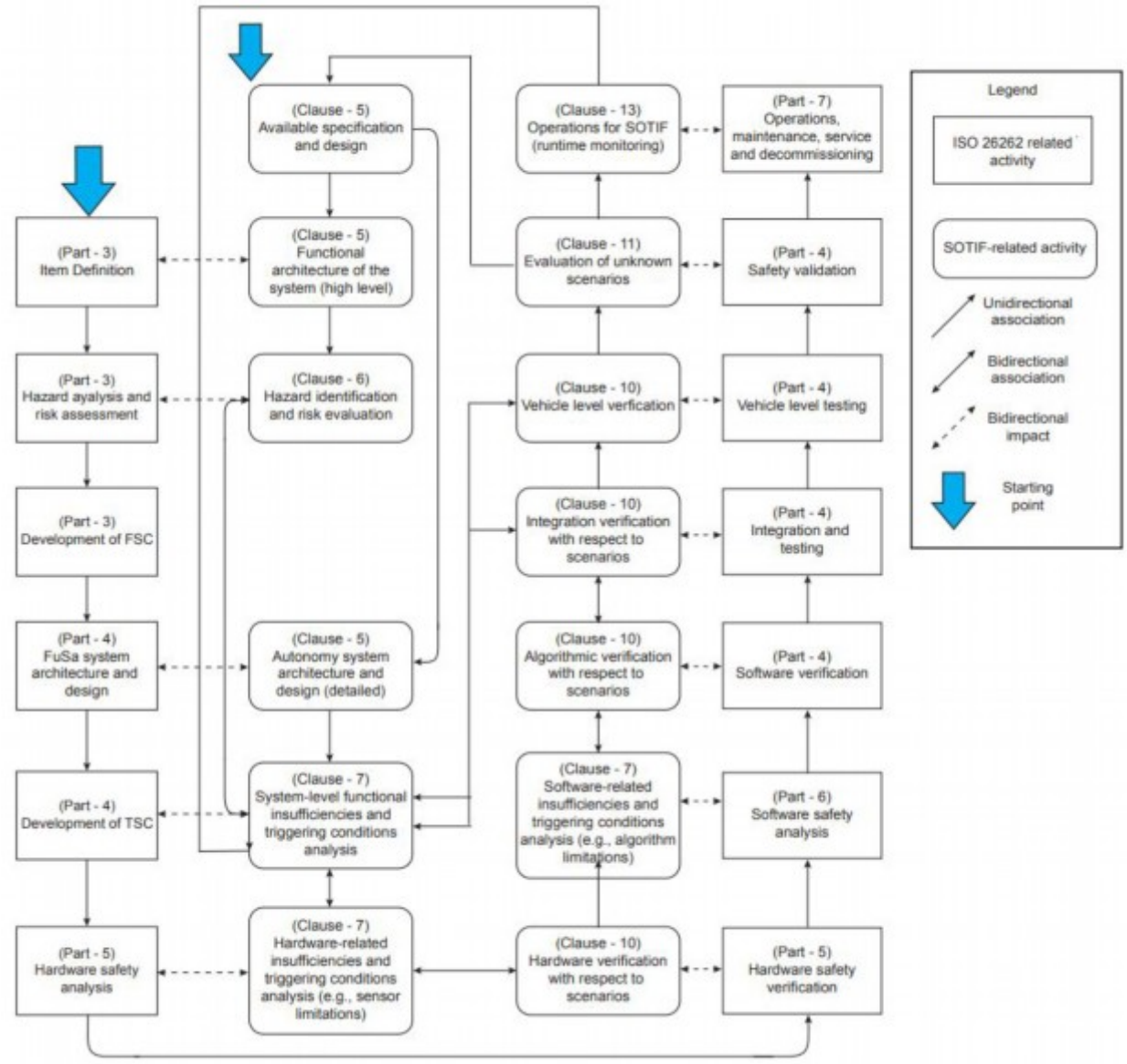

Рис. 3.4. Робочий процес, який узгоджує FuSa та SOTIF

Цей робочий процес охоплює різноманітні автономні системи та охоплює три різні типи архітектур. Ці архітектури спеціально спрямовані на взаємодію між FuSa та системою автономності. Хоча це виходить за межі цього проекту, варто зазначити, що автори також підкреслили, як вибір архітектури може вплинути на ступінь впливу між FuSa та SOTIF [37].

«Робочий процес, який узгоджує FuSa та SOTIF» [37], представлений на рисунку 3.4 також показано зв’язки між діяльністю кожного стандарту. У

випадку SOTIF асоціації також можуть бути двонаправленими, що означає, що на основі оновлень, можливо, доведеться переглянути дію. Крім того, робочий процес підкреслює вищезгаданий вплив, який FuSa та SOTIF мають один на одного [37]. Вплив FuSa на SOTIF і навпаки чітко видно на рисунку 3.4, оскільки майже на всі дії впливають дії іншого стандарту.

Як показано на рисунку 3.4 робочий процес починається зі специфікації та дизайну SOTIF і визначення елемента FuSa. Доступна інформація для специфікації та дизайну може бути обмеженою на початку, оскільки SOTIF дуже ітераційний, тобто нова інформація збирається під час кожної ітерації SOTIF [37]. На основі наявної специфікації та дизайну створюється високорівнева архітектура. Архітектура високого рівня, у свою чергу, пов’язана з визначенням елемента від FuSa. Немає необхідності збігатися між двома видами діяльності відповідних стандартів, але важливо мати на увазі, що обидві дії можуть впливати одна на одну [37]. Після визначення обох видів діяльності можна проводити аналіз небезпеки. Після завершення аналізу небезпек робочий процес продовжує виконання дій, доки не завершиться після перевірки та підтвердження.

Однак як узагальнений життєвий цикл безпеки [36], так і робочий процес, який узгоджує FuSa та SOTIF [37] розглядає HARA окремо для кожного підходу. Навпаки, підходи HARA можуть прийняти ту саму методологію, як зазначено в стандарті SOTIF [8]. Причина відмови від інтеграції в обидвох підходах HARA здебільшого пов’язані з тим, що підхід до визначення рівнів ризику відрізняється [36]. Наприклад, для FuSa рівні ризику слід визначати відповідно до ASIL. Однак це неможливо для SOTIF, оскільки він не дозволяє визначити ASIL [8]. Крім того, для SOTIF рівень прийнятного ризику та зусилля, необхідні для його досягнення, не прописані та повинні визначатися окремо для кожного випадку використання [36].

Всупереч вищезазначеним висновкам, «цикл розробки, який поєднує FuSa та SOTIF» [38] дозволяє належним чином інтегрувати рейтинг ризику

обох підходів HARA. Цикл розробки, який поєднує FuSa та SOTIF [38], показано на рисунку 3.5.

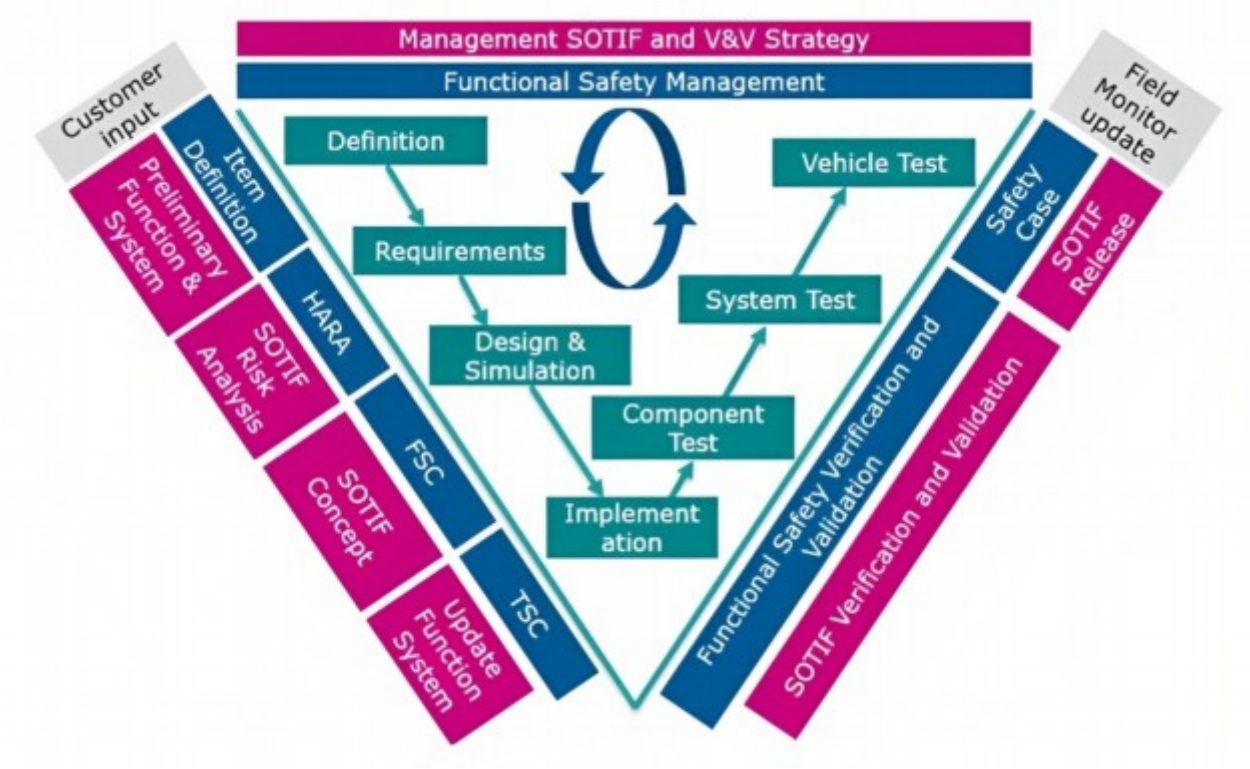

Рис. 3.5. Цикл розробки, який поєднує FuSa та SOTIF

Як показано на рисунку 3.5 процес розробки починається з визначення елемента (FuSa) і попередньої специфікації функції та системи (SOTIF). Попередня функція та специфікація системи є основою для визначення елемента [38]. Після завершення першого кроку, тобто визначення, процес розробки продовжується аналізом ризиків HARA та SOTIF. Наступним кроком у процесі розробки є розробка FSC і концепції SOTIF відповідно. FSC покладається на результат концепції SOTIF, оскільки вона визначає функціональні модифікації, які повинні бути виконані [38]. На основі концепції SOTIF необхідно оновити функцію та опис системи, а також створити TSC FuSa. TSC визначає заходи безпеки, які повинні бути вжиті, щоб уникнути або зменшити потенційну небезпеку, спричинену несправностями [38].

Праворуч від циклу розробки, зображеного на рисунку 3.5 увага зосереджена на верифікації та валідації, тому заходи SOTIF і FuSa повинні бути перевірені для кожного рівня інтеграції [38].

Поєднання підходів HARA FuSa та SOTIF

Як зазначено вище, сфера застосування FuSa зосереджена на випадкових апаратних і систематичних апаратних/програмних збоях. Подібним чином сфера застосування SOTIF зосереджена на виникненні тригерних подій як послідовність ненавмисного неправильного використання, умови навколишнього середовища або обмеження продуктивності. І тригерні події, і несправності, описані FuSa, можуть зрештою призвести до виникнення небезпеки. Щоб подати це в системну перспективу, було проведено порівняння стандартів FuSa та SOTIF [39]. Виходячи з цього порівняння, на рисунку 3.6 показано потік подій, які в кінцевому підсумку можуть призвести до потенційної небезпеки.

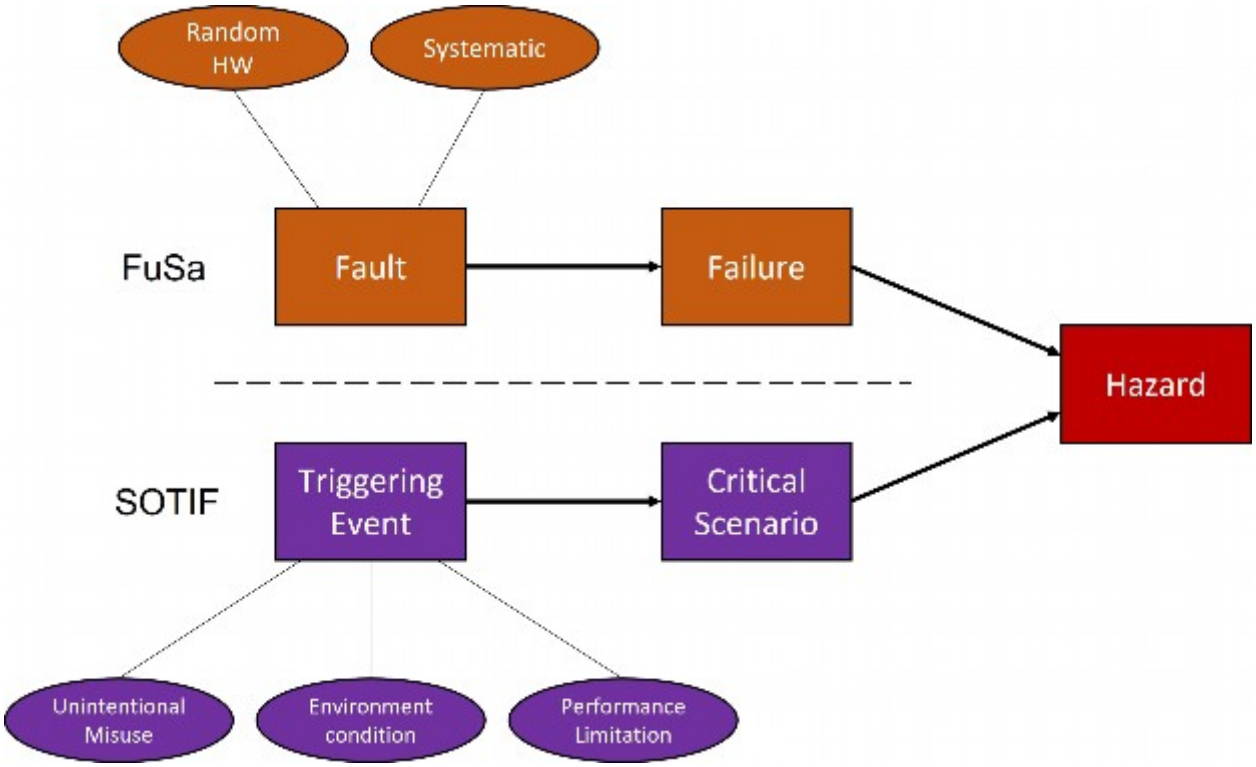

Рис. 3.6. Порівняння між масштабами та потоком подій, що призводять до небезпеки

Як зображено на рисунку 3.6 є чітке розмежування щодо обсягу. Однак, коли справа доходить до аналізу небезпеки та інших пов'язаних аспектів, між ними є подібність. Наприклад, методи визначення небезпеки для FuSa також застосовуються до SOTIF [8]. Але, як було зазначено раніше, існує різниця щодо оцінки ризику, оскільки SOTIF не дозволяє визначити ASIL. По суті, рівень ASIL визначається шляхом оцінки значень серйозності (S), експозиції (E) і контрольованості (C). Однак для SOTIF значення E не існує, оскільки воно не вважається релевантним [38].

Фундаментальні принципи аналізу небезпеки та оцінки ризику для SOTIF і FuSa можна включити як розширення (рис. 3.6). Це розширення передбачає поділ небезпеки на три окремі категорії: несправність, небезпечна подія та шкода [38]. Результати цього аналізу подано на рис. 3.7.

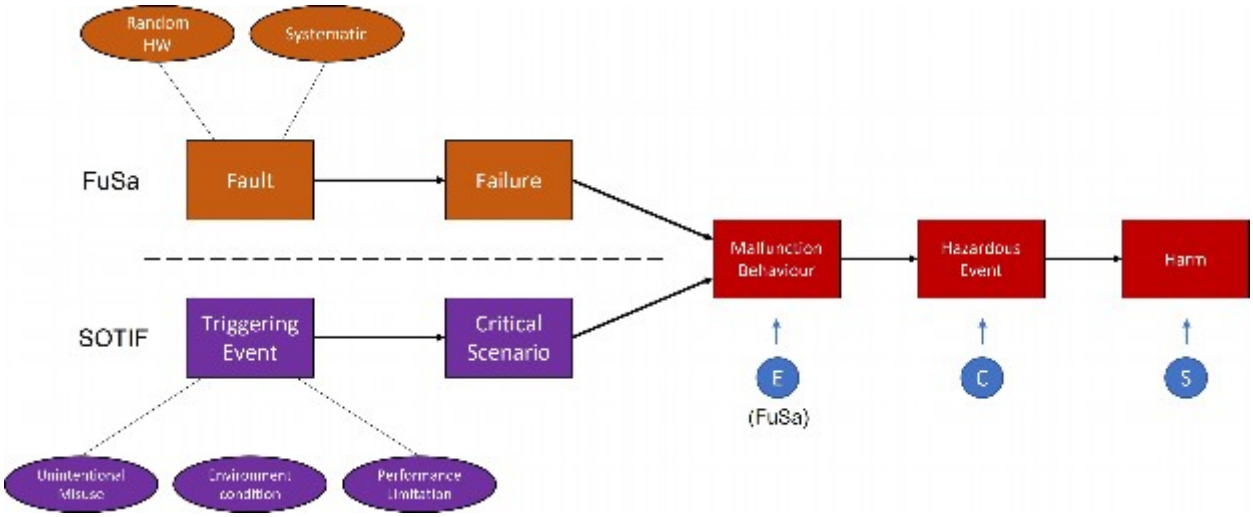

Рис. 3.7. Розширення аналізу небезпеки та оцінки ризику

Після виникнення події, що запускає, або несправності, критичний сценарій або збій, які можуть виникнути, зараз виступає ініціатором несправності. У свою чергу, несправність може спричинити небезпечну подію, яка потенційно може призвести до шкоди. Виходячи з цього, E визначається відповідно до ймовірності виникнення зазначеної несправності. Як було зазначено раніше, ця оцінка призначена виключно для визначення

ASIL FuSa. Крім того, оцінка C, яка визначає керованість, виконується, коли несправність викликає небезпечну подію. Нарешті, S оцінюється, коли небезпека завдає будь-якої шкоди.

Після визначення значень E, C і S необхідно також оцінити рівень ризику. Для FuSa це означає, що кожна небезпека має отримати рівень ASIL, тоді як для SOTIF ризик оцінюється інакше. Існує ризик SOTIF, якщо керованість і серйозність перевищують C0 і S0 відповідно [38]. Таким чином, існує ризик SOTIF, якщо обидва:

• S > 0

• C > 0

Як було показано, тепер можна об'єднати оцінку ризику FuSa та SOTIF, як також згадувалося авторами, які створили цикл розробки, який поєднує FuSa та SOTIF [38]. У цьому випадку була створена окрема матриця оцінки ризику, яка показує результати як визначення ASIL, так і рейтингу ризику SOTIF.

3.3. Поєднання трьох стандартів для розробки програмних систем

Тепер, коли обговорювалися комбінації FuSa & CE, FuSa & SOTIF, у цьому розділі докладніше розглядається комбінація всіх трьох стандартів. Знову ж таки, результати поділяються на весь процес розробки та аналіз ризиків, який у цьому випадку:

- Включення FuSa, SOTIF і CE в процес розробки;
- Поєднання підходів FuSa HARA, SOTIF HARA та TARA.

Включення FuSa, SOTIF і CE в процес розробки

Об'єднання всіх трьох стандартів в єдиний процес розробки також описано в літературі. Наприклад, було запропоновано «мультиконцептний процес розробки», який показано на рисунку 3.8, і він підкреслює взаємодію між проблемами безпеки та захисту.

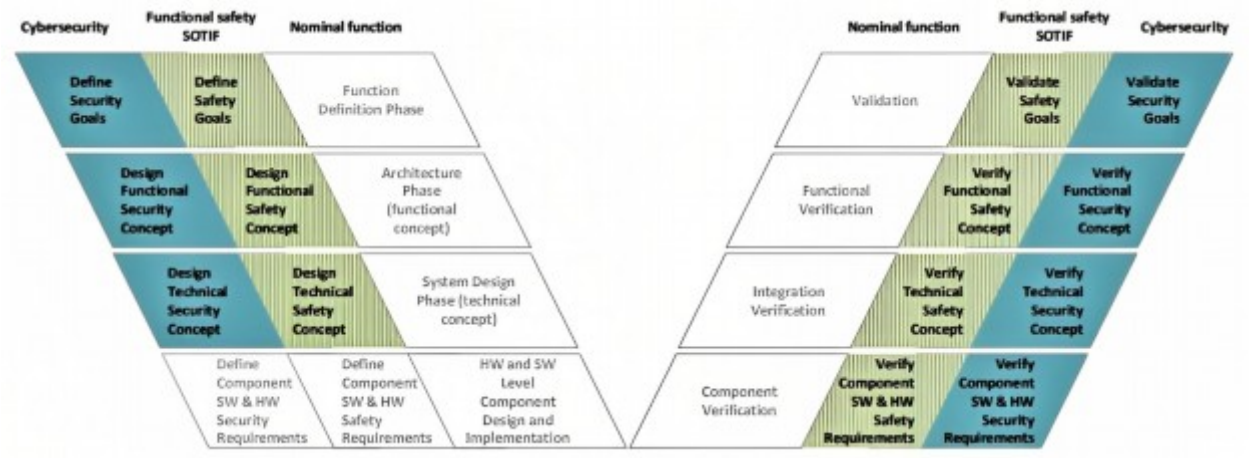

Рис. 3.8. Процес розробки на основі декількох концептів (стандартів)

Як показано на рисунку 3.8 процес розробки містить номінальну функцію, яка розширена діяльністю безпеки та захисту. Також можна адаптувати структуру стандартів FuSa, SOTIF і CE до процесу розробки. Однак, як видно на рисунку, SOTIF і FuSa об'єднані в єдиний процес безпеки. FuSa та SOTIF можуть бути об'єднані в одну проблему безпеки, оскільки обидва стандарти переплітаються один з одним. Натомість CE зосереджується на проблемі безпеки, тому його обслуговують окремо. Однак слід зазначити, що цей поділ також створюється для того, щоб можна було вирішити невідповідності, коли дві проблеми збігаються.

Метою процесу розробки на основі декількох стандартів є забезпечення синхронізації між проблемами. Дозволивши синхронізацію, можна вирішити вищезазначені невідповідності. Наприклад, загрози кібербезпеці, які можуть поставити під загрозу безпеку, повинні бути виявлені та усунені [40]. Таким чином, вісім точок синхронізації визначаються для процесу розробки багатьох концептів [40]. Вісім точок синхронізації рівномірно розподілені по обидві сторони V-циклу, тобто чотири точки для лівої сторони та чотири для правої.

Для лівої сторони частини «спільного проектування» точками синхронізації є функціональне визначення, аналіз ризиків, архітектура та фаза проектування системи [40]. Шляхом узгодження специфікації та дизайну (SOTIF) і визначення елемента (як для FuSa, так і для CE) можна

розробити залежності в операційному контексті. Подібним чином, для синхронізації аналізу ризиків також розглядаються взаємодія, наслідки та інші припущення. Крім того, для фази архітектури синхронізація дозволить усунути невідповідності в архітектурних вимогах, стратегії деградації тощо [40]. Нарешті, під час спільного проектування фаза проектування системи повинна бути синхронізована, оскільки вона додатково вимагає аналізу міркувань, прийнятих на етапі архітектури.

У частині «спільної перевірки», або правій частині, точки синхронізації складаються з перевірки компонентів, перевірки інтеграції, функціональної перевірки та перевірки системи [40]. Автори припускають , що тестування не зосереджене на конкретних проблемах, а скоріше спрямоване на забезпечення загальної якості продукту. Таким чином, повне узгодження стандартів не є основною вимогою, доки будь-які невідповідності, виявлені під час тестування, ефективно передаються на відповідну фазу спільного проектування [40].

Подібно до результатів процесу розробки на основі декількох стандартів, інші дослідження також дійшли висновку про наявність залежності між процесами безпеки та захисту [41]. Крім того, перевага - об’єднання FuSa, CE та SOTIF в єдиний процес розробки полягає в тому, що можна знизити загальні ризики [41]. Зрештою, це призвело до розробки «комбінованої V-моделі» [41], яку представлено на рисунку 3.9.

Комбінована V-модель [41] — це ітераційний процес, у якому процеси з лівої сторони моделі або підтверджуються, або перевіряються процесами з правої сторони. Крім того, як показано в комбінованій V-моделі [41], існує взаємодія між діяльністю (функціональної) безпеки та захисту. Це важливо, оскільки CE впливає, наприклад, на FuSa [41]. Тому рекомендується належний підхід до розробки, який об’єднує аспекти FuSa, CE та SOTIF. Це, зрештою, буде корисним, оскільки дає змогу включати всі можливі небезпечні події [41].

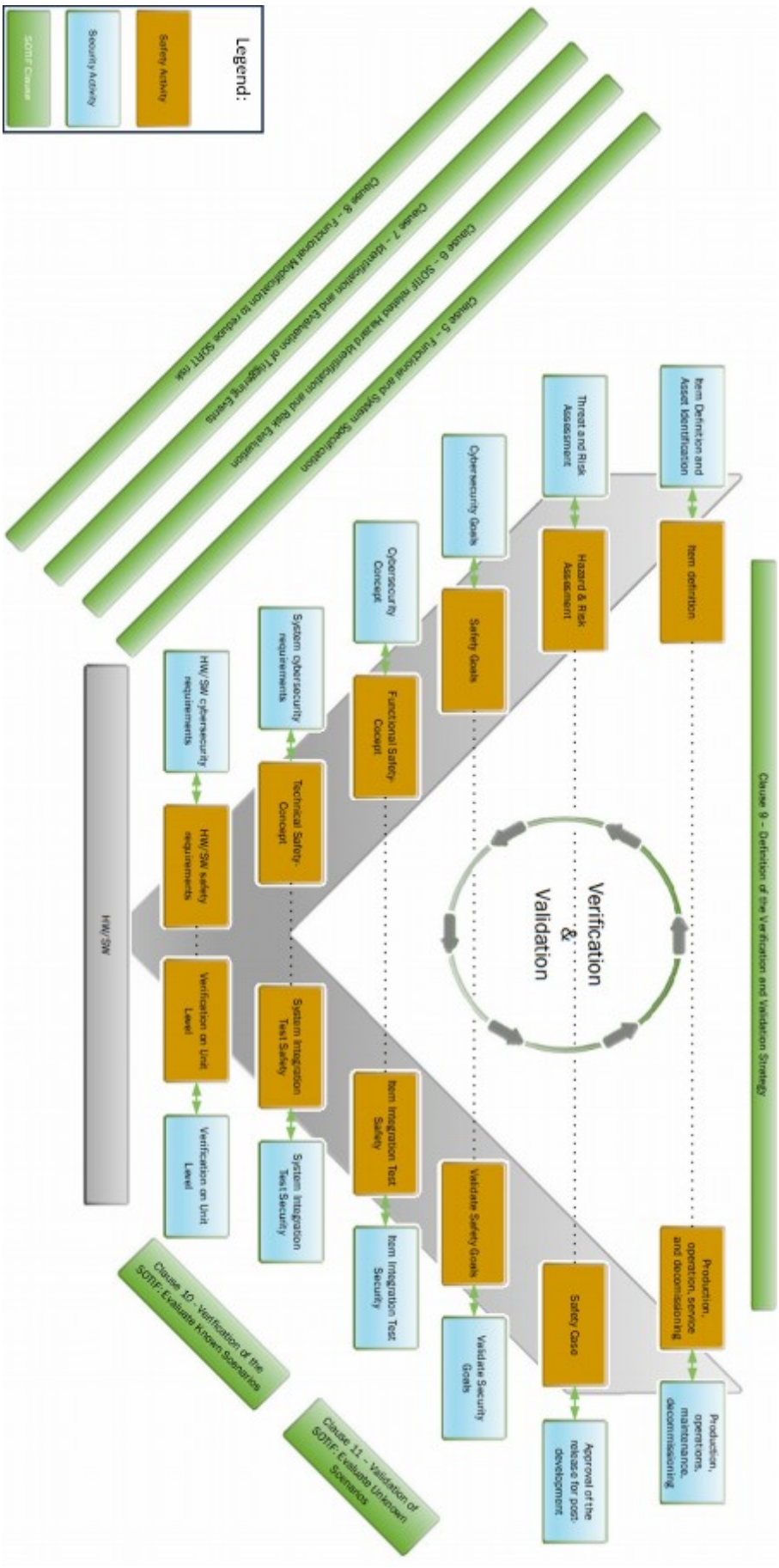

Рис. 3.9. Комбінована V-модель

Комбінована V-модель починається з визначення елемента як FuSa, так і CE, а також ідентифікації специфічних активів CE. Після завершення TARA та HARA також повинні бути виконані. Більш детально ця діяльність розглядається пізніше. Наступною діяльністю є визначення цілей та концепцій для FuSa та CE відповідно. Останні дії з лівої сторони комбінованої V-моделі вимагають визначення вимог до системи та апаратного/програмного забезпечення щодо CE та FuSa. Однак, як показано на рис. 3.9, у комбінованій V-моделі також присутні спеціальні положення SOTIF.

Наприклад, пункти SOTIF, які присутні в лівій частині об'єднаної V-моделі:

• пункт 5 – Специфікація функцій і системи;

• пункт 6 – Ідентифікація небезпеки та оцінка ризику, пов'язана з SOTIF;

• пункт 7 – Ідентифікація та оцінка ініціюючих подій;

• пункт 8 – Функціональна модифікація для зменшення ризику SOTIF.

Послідовність кроків SOTIF для лівої сторони комбінованої V-моделі починається з п. 5 і завершується п. 8. Однак, як показано в комбінованій V-моделі, взаємодія між, наприклад, пунктами SOTIF і відповідна діяльність FuSa, не розглядається.

Для правої сторони комбінованої V-моделі найважливішими аспектами є перевірка та підтвердження. Перевірка та підтвердження виконуються для FuSa, CE та SOTIF. Що стосується FuSa та CE, першим кроком є перевірка на рівні пристрою. Це супроводжується тестом інтеграції системи та, зрештою, тестом інтеграції предметів. Після цього необхідно також перевірити цілі безпеки та безпеки. Далі слідує перевірка безпеки (FuSa) і схвалення випуску (CE). Нарешті, діяльність з виробництва, експлуатації, обслуговування та виведення з експлуатації також виконується для обох стандартів. Стосовно SOTIF, права частина комбінованої V-моделі зосереджена на верифікації,

тобто оцінці відомих сценаріїв, і в кінцевому підсумку на перевірці, тобто оцінці невідомих сценаріїв.

Поєднання підходів FuSa HARA, SOTIF HARA та TARA

Відповідно до літератури, оцінка ризику є важливим аспектом процесу розробки, і співпраця між CE, FuSa та SOTIF повинна бути встановлена на кожному рівні процесу. Як зазначено в стандарті CE, процес TARA також включає оцінку впливу кожного ідентифікованого активу окремо [9]. Однак рейтинг впливу оцінює не тільки фінансовий, операційний і конфіденційний вплив, але й визначає, чи є вплив на безпеку.

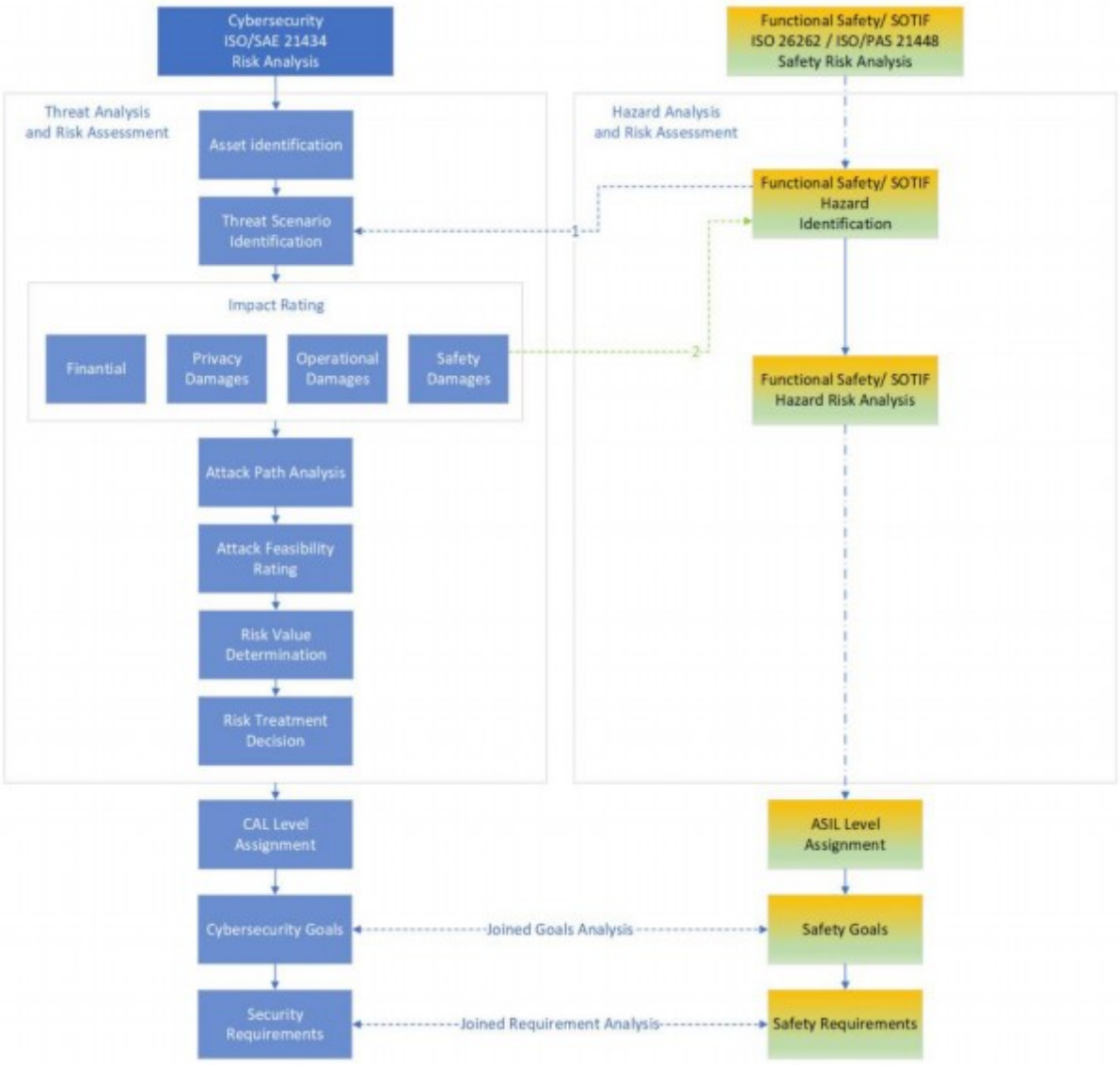

Рис. 3.10. Комбінована оцінка стандартів HARA та TARA

Таким чином, рейтинг впливу TARA на безпеку можна поєднати з процесом HARA для FuSa та SOTIF, як показано на рисунку 3.10 [41]. Зв'язок між рейтингом збитків безпеці та визначенням небезпеки для FuSa та SOTIF виділено зеленою стрілкою. Якщо шкода безпеці перевищує визначений поріг, її необхідно враховувати під час процесу HARA. Точний рейтинг збитків, пов'язаних із безпекою, базується на S, визначеному в стандарті FuSa, як зазначено в стандарті CE [9]. Однак, як показано в Малюнок 5.9 існує також зв'язок від ідентифікації небезпеки до ідентифікації сценарію загрози. Це з'єднання необхідне для забезпечення безпеки критично важливих для безпеки/автономних систем [41]. Таким чином, ідентифіковані небезпеки також повинні бути включені в ідентифікацію сценарію загрози.

Після визначення TARA та HARA для FuSa та SOTIF необхідно визначити рівень ASIL та рівень забезпечення кібербезпеки (CAL). На основі результатів обох завдань необхідно також визначити цілі кібербезпеки, а також цілі безпеки. Подібним чином, коли визначено цілі безпеки та безпеки, необхідно включити нові вимоги. Зрештою, цілі та вимоги безпеки можна об'єднати з цілями та вимогами безпеки [41].

Огляд літератури показав, що існує різноманітна література щодо включення стандартів ISO у процес розробки системи. Однак обсяги дослідницьких робіт значно відрізнялися. Деякі документи зосереджені лише на об'єднанні FuSa та CE, або FuSa та SOTIF, тоді як інші об'єднують усі три стандарти (FuSa, SOTIF та CE). Тому література була розділена на три основні категорії, а саме:

- Включення FuSa та CE у процес розробки;
- Включення FuSa та SOTIF у процес розробки;
- Включення FuSa, SOTIF і CE в процес розробки.

Під час огляду літератури було зроблено висновок, що всі три стандарти ISO впливають один на одного. Тому під час розробки слід також враховувати взаємодію фази. Подібним чином підходи HARA і TARA також

повинні бути узгоджені. Стосовно HARA і TARA було зроблено висновок, що існують різні варіанти як прямого поєднання HARA і TARA, так і шляхом вибору методу, який об'єднує аспекти доменів безпеки та кіберзахисту. Зрештою, огляд літератури виявив відсутність єдиного підходу до включення стандартів ISO. Отже, необхідно визначити найбільш прийнятний підхід для інтеграції стандартів ISO.

Основні зусилля цього дослідження були зосереджені на інтеграції стандартів FuSa, SOTIF і CE у процес розробки. Крім того, інструментарій Motar також був інтегрований у процес розробки. Отриманий комбінований процес розробки системи на основі безпеки та безпеки взяв до уваги різні етапи всіх трьох вищезазначених стандартів ISO.

Як згадувалося, дослідження дали пропозицію щодо комбінованого процесу розробки системи на основі безпеки та захисту. Спочатку був розроблений V-цикл, який інтегрував різні аспекти стандартів FuSa, SOTIF і CE. Комбінований V-цикл розробки потім був удосконалений у комбінований процес розробки системи на основі безпеки та безпеки, який, по суті, є робочим процесом.

Комбінований процес розробки системи на основі безпеки та захисту відрізняється від аналогів, які були знайдені в літературі, оскільки аналоги або поєднують FuSa та SOTIF в єдину проблему безпеки [40], або не розглядають взаємодію між ними [41]. Причина відмови від поєднання FuSa та SOTIF полягає в тому, що хоча FuSa та SOTIF значною мірою перетинаються один з одним і обидва стосуються аспекту безпеки, робочий процес виявився різним [37]. Тому було прийнято рішення підтримувати належне розмежування між FuSa та SOTIF і, таким чином, включити аспекти робочого процесу, який вирівнює FuSa та SOTIF [37] у комбінований процес розробки системи на основі безпеки та захисту.

Крім того, огляд літератури також показав, що FuSa, SOTIF і CE впливають один на одного, а це означає, що слід розглянути потенційну взаємодію між стандартами. Тому було прийнято рішення створити

об’єднану оцінку ризику, яка поєднує HARA від FuSa та SOTIF з TARA від CE. На відміну від підходу THARA, комбінована оцінка ризику не вимагає відображення активів HARA елемента TARA і навпаки [33]. Подібним чином комбінована оцінка ризику не вимагає розрахунку коефіцієнта серйозності безпеки загрози відповідно до ASIL, в якому немає запропонованого методу [33]. Інші методи, запропоновані в літературі, були спрямовані на поєднання сфери безпеки та безпеки, а не на пряме поєднання підходів HARA та TARA. Зрештою, комбінована оцінка ризику виявилася більш простим підходом, водночас безпосередньо пов’язуючи підхід HARA та TARA, тому її було обрано для цього дослідження.

Як було також зазначено, комбінований процес розробки систем безпеки та безпеки застосовується до систем SAE на рівнях 1-5. Таким чином, процес можна використовувати, наприклад, для розробки не тільки ADAS, але й ADS. Однак, як зазначено у вступі, стандарт ISO 5083, який розглядає етапи проектування та методи валідації для транспортних засобів рівня 3 і 4 SAE [14], може вплинути на комбінований процес розробки системи на основі безпеки та безпеки. З ISO 5083 також фокусується на кібербезпеці, він може надати цінну інформацію. Оскільки в цьому дослідженні не враховувався стандарт ISO 5083, оскільки він все ще розробляється, висновки ISO 5083 слід враховувати в майбутніх дослідженнях, особливо коли мова йде про системи рівня SAE 3-4.

Це дослідження також пояснило прогалини в стандартах FuSa та SOTIF щодо ризиків, які не завдають шкоди [38]. Оскільки результат події, яка не завдає шкоди, все одно може призвести до збитків, хоча й незначних, їх можна вирішити в майбутньому. Подібним чином це дослідження виявило той факт, що рівні системної інженерії між стандартами відрізняються. У той час як FuSa та SOTIF використовують такі терміни, як елемент, система та компонент, CE використовує елемент, компонент та підкомпонент. Оскільки це може призвести до обговорення, рекомендується узгодити терміни рівнів системної інженерії в усіх стандартах.

Нарешті, як також згадувалося, поки що Motar toolbox не має жодної форми відповідності інструменту, наприклад, FuSa. Подібним чином, для CE керування інструментами також відіграє важливу роль. Таким чином, рекомендується додатково досліджувати теми відповідності інструментів і керування інструментами, щоб Motar toolbox можна було затвердити як інструмент для розробки автомобільних систем безпеки та систем SAE.

Висновки до розділу

У цьому розділі представлено моделі та методології розробки програмних систем, засновані на концепціях безпеки та захисту. Розділ структуровано таким чином, щоб спочатку пояснити застосований підхід, а далі подати детальний аналіз результатів огляду літератури. Огляд літератури поділено на три категорії: 1) висновки, пов’язані з інтеграцією стандартів FuSa і CE, 2) висновки, що стосуються включення стандартів FuSa і SOTIF, та 3) результати, що охоплюють одночасне використання стандартів FuSa, SOTIF і CE. Кожна з цих категорій розглядається в окремих підрозділах. Варто зазначити, що комбінація стандартів SOTIF і CE не виявила відповідних літературних джерел під час початкового пошуку, тому ця частина дослідження не розглядається.

ВИСНОВКИ

У магістерській роботі було проведено дослідження моделей, методологій та стандартів розробки програмних систем, заснованих на концепціях безпеки та захисту. На початкових етапах дослідження розглянуто методи, що використовувалися в процесі розробки, з особливою увагою до результатів попередніх досліджень, які зосереджувалися на стандартах ISO (зокрема FuSa, SOTIF і CE), важливості їх поєднання та інтеграції в процес розробки програмних систем.

Ключовим питанням дослідження було визначення способів інтеграції стандартів автомобільної безпеки в процес розробки систем для автомобільних застосунків. На початковій стадії дослідження зосереджено увагу на отриманні інформації щодо використання інструменту Motar toolbox клієнтами та процесу розробки. Крім того, було оцінено загальний рівень обізнаності стосовно вищезгаданих стандартів ISO. За результатами дослідження було встановлено, що в процесі розробки часто використовували V-cycle (циклічну модель розробки), а інструмент Motar toolbox зазвичай застосовувався під час реалізації систем після етапу специфікації вимог. На основі отриманих даних та результатів огляду літератури було створено загальний V-цикл, який пізніше було доповнено відповідною інформацією з літературних джерел.

Подальше дослідження охоплювало огляд літератури з метою отримання інформації щодо інтеграції стандартів FuSa, SOTIF та CE у розроблений V-цикл. Літературні джерела було структуровано в три категорії, що охоплюють можливі варіанти поєднання цих трьох стандартів. Використовуючи систематичний підхід, було отримано відповідну літературу та зроблено висновки, які стали основою для пропозиції комбінованого V-циклу розробки. Цей V-цикл базувався на загальному циклі, розширеному результатами літературного огляду.

Отриманий комбінований V-цикл розробки був удосконалений до комбінованого процесу розробки системи, заснованого на концепціях безпеки та захисту, який включав робочий процес із переліком усіх дій, що мають бути виконані на кожному етапі розробки. Крім того, створений робочий процес враховував можливу взаємодію між стандартами FuSa, SOTIF і CE. Остаточний варіант робочого процесу було оформлено у вигляді посібника, в якому розглядалися всі аспекти комбінованого процесу розробки системи на основі безпеки та захисту.

Оцінювання комбінованого процесу розробки системи проводилося у два етапи. Перший етап стосувався обґрунтування комбінованого V-циклу за допомогою літературних джерел. Другий етап полягав у практичному застосуванні розробленого циклу. Метою цього прикладу було перевірити, чи відповідає комбінований V-цикл та подальший процес розробки робочого процесу поставленим цілям. Результати оцінки показали, що робочий процес комбінованого процесу розробки систем на основі безпеки та захисту є ефективним інструментом для оптимізації розробки програмних систем.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. K. Barry, “Most Drivers Like New Advanced Safety Technology, CR Survey Says.” https://www.consumerreports.org/car-safety/car-safety-technology-adas-survey-a2785002723/
2. C. Caferri, “General Safety Regulation comes into force.” https://www.acea.auto/news/general-safety-regulation-comes-into-force/
3. SAE, “J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles.” https://www.sae.org/standards/content/j3016_202104
4. B. Kelechava, “Defining Automated Driving Systems in SAE J 3016-2021.” https://blog.ansi.org/defining-automated-driving-systems-sae-j-3016-2021/#gref
5. Y. Dajsuren and Van den Brand, M., Automotive Systems and Software Engineering. Springer Nature Switzerland AG, 2019.
6. M. Staron, Automotive Software Architectures. Springer Nature Switzerland AG, 2017.
7. International Organization for Standardization, “ISO 26262: Road vehicles — Functional safety.” https://www.iso.org/standard/68383.html
8. International Organization for Standardization, “ISO 21448: Road vehicles — Safety of the intended functionality.” https://www.iso.org/standard/77490.html
9. International Organization for Standardization, “ISO 21434: Road vehicles — Cybersecurity engineering.” https://www.iso.org/standard/70918.html
10. ICT Group, “Over ICT Group.” https://www.ict.eu/nl/over-ons/over-onze-group/over-ict-group.
11. ICT Group, “MOTAR.” https://www.ict.eu/nl/producten/motar-model-based-software-development-voor-automotive
12. ICT Group, “Save Time Converting Your Prototype into a Finished Product,”.

13. Safetronic, “ISO TC22-SC32-WG13 ISO TS 5083.” https://safetronic.fraunhofer.de/wp-content/uploads/2021/11/Nov17-UpdatesStandarization-Fuerst.pdf,
14. International Organization for Standardization, “ISO/AWI TS 5083: Road vehicles — Safety for automated driving systems — Design, verification and validation.” https://www.iso.org/standard/81920.html
15. B. Martin, B. Hanington, and B. Hanington, Universal Methods of Design: 100 Ways to Research Complex Problems, Develop Innovative Ideas, and Design Effective Solutions. Rockport Publishers, 2012.
16. Merkus, J., “Een introductie tot exploratief onderzoek (exploratory research).” https://www.scribbr.nl/onderzoeksmethoden/exploratief-onderz
17. George, T. and Merkus, J., “Explanatory Research.” https://www.scribbr.com/methodology/explanatory-research/
18. Voxco, “Conceptual Research.” https://www.voxco.com/blog/conceptual-research/#:~:text=Conceptual%20research%2C%20as%20the%20name,information%20on%20a%20given%20topic.
19. Voxco, “What is Evaluation Research?.” https://www.voxco.com/blog/what-is-evaluation-research/
20. American Society for Quality, “WHAT IS THE NINE WINDOWS TECHNIQUE?.” https://asq.org/quality-resources/nine-windows
21. M. Sharan B. and T. Elizabeth J., Qualitative Research : A Guide to Design and Implementation, vol. Fourth edition of The Jossey-Bass Higher and Adult Education Series. Jossey-Bass, 2016.
22. Skjott Linneberg, M. and Korsgaard, S., “Coding qualitative data: a synthesis guiding the novice,” Qualitative Research Journal, vol. 19, no. 3, pp. 259–270, 2019.
23. D. V. Thiel, Research Methods for Engineers. Cambridge University Press, 2014.
24. Cao, S., “Tesla’s Claim That Its Cars Are Self-Driving May Cross the Line From Permitted ‘Puffery’ to False Advertising.”

https://observer.com/2022/09/tesla-self-driving-software-face-false-advertising-elon-musk/

25. SAE, “SAE Levels of Driving Automation Refined for Clarity and International Audience.” https://www.sae.org/blog/sae-j3016-update
26. Toma, S., “Mercedes-Benz Boasts Automated Driving Redundancy Strategy, Elon Should See It.” https://observer.com/2022/09/tesla-self-driving-software-face-false-advertising-elon-musk/
27. R. Rana, M. Staron, C. Berger, J. Hansson, M. Nilsson, and F. Törner, “Increasing Efficiency of ISO 26262 Verification and Validation by Combining Fault Injection and Mutation Testing with Model Based Development,” July 2013.
28. Schneier, B. (2015). Applied Cryptography: Protocols, Algorithms, and Source Code in C. Wiley.
29. Bishop, M. (2002). Computer Security: Art and Science. Addison-Wesley.
30. Anderson, R. (2020). Security Engineering: A Guide to Building Dependable Distributed Systems. Wiley.
31. Pfleeger, C. P., & Pfleeger, S. L. (2012). Security in Computing. Prentice Hall.
32. Stallings, W., & Brown, L. (2018). Computer Security: Principles and Practice. Pearson.
33. Viega, J., & McGraw, G. (2001). Building Secure Software: How to Avoid Security Problems the Right Way. Addison-Wesley.
34. Amoroso, E. (2012). Cyber Attacks: Protecting National Infrastructure. Elsevier.
35. Landwehr, C. E. (2001). Formal Models for Computer Security. ACM Computing Surveys.
36. Sommerville, I. (2019). Software Engineering. Pearson.
37. Saltzer, J. H., & Schroeder, M. D. (1975). The Protection of Information in Computer Systems. IEEE Proceedings.
38. Shostack, A. (2014). Threat Modeling: Designing for Security. Wiley.

39. Gollmann, D. (2011). Computer Security. Wiley.
40. Howard, M., & LeBlanc, D. (2002). Writing Secure Code. Microsoft Press.
41. Smith, R. E. (2002). Authentication: From Passwords to Public Keys. Addison-Wesley.
42. Jones, C., & Rastogi, S. (2017). Securing DevOps: Security in the Cloud. Manning Publications.
43. Garfinkel, T., & Rosenblum, M. (2003). A Virtual Machine Introspection Based Architecture for Intrusion Detection. ACM SIGOPS.
44. Ross, R. (2014). Security and Privacy Controls for Federal Information Systems and Organizations. NIST Special Publication.
45. Nunes Leal Franqueira, V., et al. (2011). Attack Patterns for Security Requirements Elicitation. International Conference on Software Engineering and Knowledge Engineering.
46. Fernandez, E. B. (2013). Security Patterns in Practice: Designing Secure Architectures Using Software Patterns. Wiley.
47. Sutton, M. (2007). Fuzzing: Brute Force Vulnerability Discovery. Addison-Wesley.
48. Peltier, T. R. (2013). Information Security Risk Analysis. CRC Press.
49. Swiderski, F., & Snyder, W. (2004). Threat Modeling. Microsoft Press.
50. Schneier, B. (2000). Secrets and Lies: Digital Security in a Networked World. Wiley.
51. Snekkenes, E. (1992). Roles and Security in Computer Systems. ACM Transactions on Information Systems Security.
52. De Capitani di Vimercati, S., & Samarati, P. (2003). Access Control: Policies, Models, and Mechanisms. Springer.