SafeSecRETS: A Safety and Security Requirements Tool for Critical IoT Systems

Ernesto Fonseca Veiga
Instituto de Informática
Universidade Federal de Goiás
Goiânia, Goiás, Brazil
ernestoveiga@discente.ufg.br

Taciana Novo Kudo
Instituto de Informática
Universidade Federal de Goiás
Goiânia, Goiás, Brazil
taciana@ufg.br

Renato de Freitas Bulcão-Neto
Instituto de Informática
Universidade Federal de Goiás
Goiânia, Goiás, Brazil
rbulcao@ufg.br

ABSTRACT

[Context] Assuring safety and security from the earliest stages of development is essential, particularly for critical systems. However, addressing these requirements in complex and heterogeneous systems such as those based on the Internet of Things (IoT) is challenging. These systems often operate in dynamic and diverse environments, where vulnerabilities can lead to severe consequences if not effectively mitigated. [Objective] This paper presents *SafeSecRETS*, a tool supporting safety and security Requirements Engineering (RE) for critical IoT systems. [Method] The tool features a collaborative pipeline with a strategic canvas-based IoT project planning and an extended Systems-Theoretic Process Analysis (STPA) method for safety and security. [Results] *SafeSecRETS* assists requirements engineers, domain experts, and other stakeholders in collaboratively defining project scope and eliciting, analyzing, and specifying system requirements. Its interconnected visual components guide users through this process, fostering engagement among information, people, and decision-making. A case study on an automated insulin delivery system illustrates the tool's applicability and presents its structured and integrated approach to RE safer and more secure critical IoT systems.

***SafeSecRETS* demo video:** https://zenodo.org/records/17000378

KEYWORDS

Planning, Analysis, Safety, Security, IoT, Requirements, Traceability

1 Introduction

The growing adoption of Internet of Things (IoT)- based critical systems has transformed several domains, including healthcare, transportation, industry, and critical infrastructure, by enabling intelligent and connected services that directly impact people's safety and well-being. In such environments, ensuring safety and security from the early stages of system development is essential to prevent threats to human life, physical harm, economic losses, or other unacceptable consequences [8, 9].

Safety refers to a system's ability to operate without causing harm to people, the environment, or system assets, primarily addressing accidental failures with no malicious intent [9]. Security focuses on protecting the system from both accidental and intentional threats, including malicious attacks that may compromise functionality or data [9]. Addressing safety and security requirements early in the system development life cycle is, therefore, crucial to ensuring the reliability and resilience of critical IoT systems [14].

To address safety concerns in today's increasingly complex systems, the System-Theoretic Process Analysis (STPA) method was proposed as a proactive approach to identifying accident causes and supporting early mitigation strategies [8]. Although STPA was originally developed to focus on safety, its system-theoretic foundation also revealed parallels with security challenges. As a result, several studies have proposed extensions and adaptations of STPA to jointly address safety and security requirements [5–7, 10, 13, 17].

However, the effective application of STPA-based approaches in critical IoT systems goes beyond the hazard and threat analysis. It requires a clear understanding of the system's objectives, boundaries, and operational context [15]. Regarding that, project planning and scope definition are foundational steps to align the requirements engineering (RE) process with the system's critical functions and constraints. It helps identify key components, interactions, and stakeholder concerns – essential elements for modeling the control structure and deriving meaningful safety and security constraints [14]. In complex and interconnected IoT environments, effective planning leads to more accurate and complete analyses, minimizing the risk of overlooking hazardous scenarios or latent security threats during system development.

Given these challenges, in prior work [14, 15], we introduced the *SafeSecIoT Canvas*, a strategic and visual artifact designed to support the planning of critical IoT projects while bridging the gap between scope definition and STPA-based safety and security analysis. By organizing *fundamental questions* into interconnected *building blocks*, the canvas facilitates the early capture of essential project information, supporting the identification of key elements for the subsequent STPA-based analysis. This structured approach enables a more holistic and integrated analysis of potential vulnerabilities from the outset of system development.

Building on this foundation, this work presents *SafeSecRETS*, a collaborative web-based software tool that integrates agile project planning with STPA-based safety and security analysis. Grounded in a canvas model, the tool is designed to support requirements engineers, domain experts, and security specialists in collaboratively conducting the elicitation, analysis, and specification of safety and security requirements. By structuring project information and enabling traceability between planning elements and STPA analysis artifacts, *SafeSecRETS* promotes a more systematic and comprehensive approach to identifying safety and security requirements. As a proof of concept, the tool was applied to the planning and analysis of a critical IoT system for automated insulin delivery (AID).

This paper is structured as follows: Section 2 presents the theoretical background and artifacts that grounded the development of the proposed tool; Section 3 analyzes related work; Section 4 details the *SafeSecRETS*, including requirements, architecture and functionality; Section 5 presents a tool usage demo; and Section 6 brings final remarks and future work.

2 Background

In prior research, we introduced *MM4Canvas* [15], a metamodel that provides a structured approach for developing canvas-based models. It defines key fundamental questions to support agile project planning and enables the creation of building blocks aligned with the canvas model's purpose. These blocks can be either: (i) general-purpose, addressing broad planning aspects (e.g., project or business-oriented); or (ii) domain-specific, tailored to specialized needs (e.g., IoT, safety, security).

The *SafeSecIoT Canvas* model is an instance of the *MM4Canvas* metamodel, tailored for agile project planning in critical IoT systems. To construct this model, we first instantiate a conceptual structure based on the components and relationships defined in the *Project Model Canvas* (PMC) [4]. We then reuse general-purpose components from PMC to address essential project planning elements. In addition, we extend the model with domain-specific components designed to capture concepts and concerns specific to critical IoT systems, such as safety and security, and to support subsequent STPA-based analysis [14].

The IoT domain-specific building blocks defined in the *SafeSecIoT Canvas* [2] are: i) components – hardware and software elements such as sensors, actuators, and algorithms for identification, control, and orchestration; ii) connectivity – the means of communication between components; iii) actions – relevant system-level interactions; and iv) data – information generated and processed by components. About safety and security concerns [8], the model includes additional blocks: i) assets – valuable entities that require protection, including people, resources, environments, or services; ii) losses – unacceptable outcomes resulting from accidents or attacks; and iii) risks – potential causes of losses, whether due to intentional threats or unintentional failures. These building blocks provide structured inputs for the STPA-based safety and security analysis, following the approach proposed by Veiga et al. [14], presented in Figure 1.

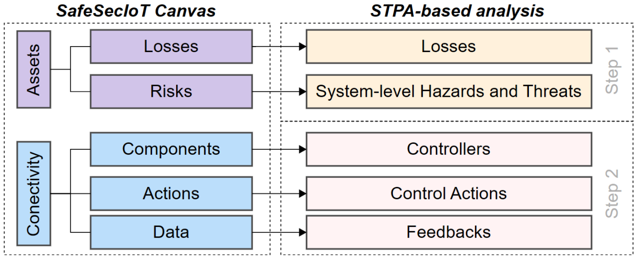


Figure 1: Linking planning with *SafeSecIoT Canvas* and STPA-based analysis [14].

To support integrated safety and security requirements analysis and specification, we propose an extension of the STPA method tailored to critical IoT systems, named *STPA-SafeSecIoT* [13]. This approach enhances the original STPA method by enabling the co-analysis of safety and security from the early design stages. Its objective is to prevent system losses not only due to known or accidental hazards, but also those stemming from intentional threats or unknown vulnerabilities, such as attacks from malicious actors. *STPA-SafeSecIoT* extends traditional STPA activities to explicitly incorporate security-related elements, fostering a unified view of risks and their impact. Figure 2 illustrates the outputs of each step in the method and the traceability links that connect planning components to safety and security analysis artifacts.

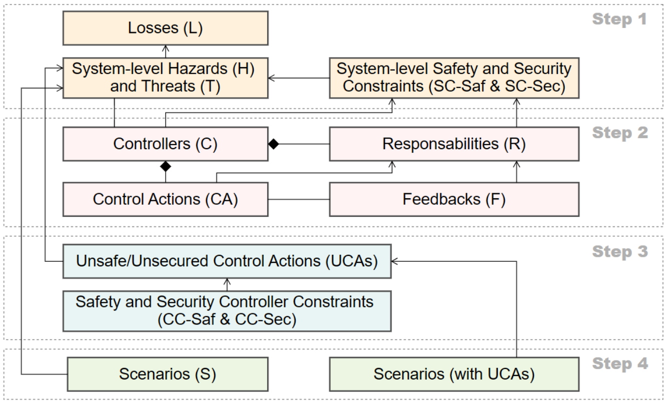

Losses (L)
Step 1
System-level Hazards (H) and Threats (T)
System-level Safety and Security Constraints (SC-Saf & SC-Sec)
Controllers (C)
Responsabilities (R)
Step 2
Control Actions (CA)
Feedbacks (F)
Unsafe/Unsecured Control Actions (UCAs)
Step 3
Safety and Security Controller Constraints (CC-Saf & CC-Sec)
Scenarios (S)
Scenarios (with UCAs)
Step 4

Figure 2: *STPA-SafeSecIoT* steps and traceability [13].

Next, we provide a general description of the activities performed in each step of the analysis process illustrated in Figure 2.

Step 1. Defining the Purpose of the Analysis. Supported by *SafeSecIoT Canvas*, this step begins by establishing the system boundaries (scope) for the analysis. It then involves: (i) identifying potential losses that may impact stakeholders' valuable assets; (ii) determining system-level hazards and threats, which represent unsafe or unsecured conditions that may lead to those losses; and (iii) defining safety and security constraints—conditions or behaviors that must be ensured to prevent the occurrence of such hazards and threats.

Step 2. Modeling the Control Structure. A hierarchical control structure is modeled to address emerging safety and security issues from component interactions. This structure includes controllers, control actions, the controlled process, and feedback loops, which together define the responsibilities and relationships among system elements in the form of control loops.

Step 3. Identifying Unsafe or Unsecured Control Actions and Specify Requirements. This step focuses on: (i) identifying unsafe or unsecured control actions (UCAs) that could lead to safety or security failures; and (ii) deriving corresponding safety and security constraints (requirements) for each controller to mitigate the risks associated with these UCAs.

Step 4. Identifying Loss Scenarios. This step involves identifying and describing loss scenarios — causal chains of events or conditions that can lead to the identified UCAs and, consequently, to hazards and threats.

3 Related Work

With the rise of critical systems requiring safety-security alignment, research has focused on integrating co-analysis through STPA extensions [5–7, 10, 17]. Friedberg et al. [5] linking an abstract control structure to the physical system design to incorporate traditional security analysis. Ribeiro et al. [10] explore an STPA-based safety and security RE process for autonomous vehicle development. Zhou et al. [17] and Gomola and Utne [7] focus on eliminating or mitigating hazards based on identified loss scenarios. However, none of these STPA extensions have been implemented or supported through dedicated software tools, which limits their practical application in system development and industry.

Although many tools have been developed for STPA, there is still a gap in supporting safety and security extensions. WebSTAMP [12] systematizes Steps 1 and 2 of STPA and STPA-Sec and enables collaborative analysis. XSTAMPP [1] offers the most comprehensive functionality, including verification, rich user experience, and portability, but lacks support for Step 2 of STPA-Sec, collaboration, and reusability. Moreover, none of these tools includes a structured project planning stage to guide the analysis, which can lead to inaccuracies when system knowledge is limited [6]. These limitations reveal a gap in supporting fully integrated and systematic safety-security co-analysis. In contrast, *SafeSecRETS* combines both aspects in a structured, visually guided process, enhances collaboration through project sharing, and supports AI-assisted analysis, traceability, and reusability, within a rich user experience.

Building on artifacts proposed in our previous works (presented in Section 2) and aiming to support other STPA-extended approaches, this paper introduces a collaborative tool for project planning at the outset of STPA analysis. The tool facilitates scope definition and the identification of essential project information. Using a canvas-oriented approach, it structures key data into building blocks that serve as inputs for the subsequent STPA-based analysis.

4 *SafeSecRETS* Overview

SafeSecRETS (an acronym for **Saf**ety and **Sec**urity **R**equirements **E**ngineering for Critical **IoT S**ystems) is a web tool for project planning and RE of critical IoT systems. Developed using Bubble and Supabase[1], it provides a structured workflow that supports project planning and elicitation, analysis, and specification of STPA-based requirements. The tool features a set of interconnected building blocks, presented progressively according to the stage to be carried out, enabling multi-user collaboration and real-time UI updates.

4.1 Requirements, Architecture and Development

4.1.1 *Functional and Non-Functional Requirements.*

The functional requirements of the *SafeSecRETS* tool include, but are not limited to: i) user registration and authentication; ii) project management capabilities, including the association of multiple users to enable collaborative work; iii) canvas-based building blocks to support project planning; iv) interactive interfaces for structured information gathering; v) visualization of the *SafeSecIoT Canvas* to provide a clear overview of the project scope; and vi) an integrated pipeline for conducting STPA-based safety and security analysis and for specifying related requirements. The key non-functional requirements are: i) usability: the system must offer an intuitive and interactive interface, aligned with UI/UX best practices to ensure a seamless user experience; ii) scalability: the event-driven architecture (EDA) must support a growing number of concurrent users without degrading performance; iii) security: robust access control and authentication mechanisms must ensure that only authorized users can access or modify project data or receive system events; iv) performance: real-time updates in collaborative sessions must be supported via asynchronous WebSocket communication, minimizing latency and reducing database load; v) persistence and reliability: the underlying database must guarantee atomicity, consistency, isolation and durability compliance. Furthermore, the project planning process supported by the tool adheres to the guidelines of the international standard ISO/IEC/IEEE 15288 [14].

4.1.2 *System Architecture.*

Figure 3 presents the system architecture of the *SafeSecRETS*, which combines the principles of modularity and separation of responsibilities of layered architecture with the reactivity and decoupling of EDA. The architecture layers and their elements are explained below.

- **Presentation Layer**: It manages the tool's graphical interface, enabling users to interact with *SafeSecIoT Canvas* building blocks for project planning and the STPA-based pipeline. Built with Bubble, a no-code platform for dynamic and interactive interfaces, it defines the workflow logic for processing user events and communicating with the backend. Its key components are i) Interactive *SafeSecIoT Canvas*: implements project planning building blocks; ii) Pipeline *STPA-SafeSecIoT*: presents steps for safety and security requirements analysis and specification; iii) Bubble Workflows: handle navigation, updates, and backend interactions; and iv) Bubble API Connector: integrates external APIs like Supabase for authentication and CRUD operations. User interactions with *SafeSecRETS* trigger Bubble Workflows, executing actions such as API calls to Supabase.
- **Communication Layer**: Built in Java-Script as a Bubble plugin, it serves as a bridge between the backend and frontend, enabling Bubble to receive real-time database updates. It connects to Supabase Realtime via WebSockets, eliminating constant database queries and enhancing performance. Its core components are i) WebSocket Client: maintains a persistent connection with Supabase Realtime; and ii) Event Listener: subscribes to specific events (insert, update, delete). When a change occurs in monitored tables, the plugin triggers an event in Bubble, dynamically updating the UI with the new information.
- **Persistence Layer**: It handles project data storage and real-time communication, using Supabase, which integrates a PostgreSQL database with authentication services and REST APIs. Its key components are i) PostgreSQL: relational database; ii) Supabase Realtime: monitors database changes and transmits events via WebSockets; and iii) RESTful API: enables authentication and authorization via Supabase Auth, ensuring secure data access and modification. The Bubble application interacts with the database through Supabase's API, while Supabase Realtime notifies the plugin via WebSockets upon data updates.
- **Artificial Intelligence (AI) Assistant Layer**: Acts as a provider of a Large Language Model (LLM) responsible for the automatic generation of safety and security constraints from specialized information provided by the system's users. The used Mixtral-8x22B-Instruct model is widely applied to interactive prompt-based assistance.

[1] Bubble: https://bubble.io/ and Supabase: https://supabase.com/

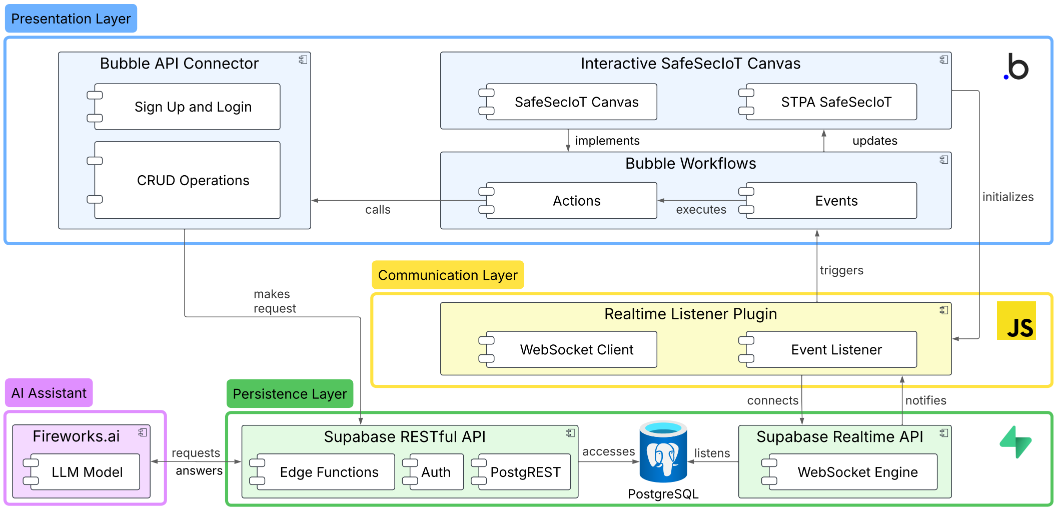


Figure 3: *SafeSecRETS* architecture.

4.1.3 Development and Technologies. *SafeSecRETS* was developed using the following technologies: i) Bubble, a no-code platform for creating web applications with a visual editor for UI design, logic, and database management, along with API integration and custom JavaScript support; ii) Supabase, a backend-as-a-service platform offering a PostgreSQL database with auto-generated APIs, authentication, storage, and real-time features; iii) Fireworks AI, a platform that offers services for the development and use of Generative AI models, focused on large-scale, low-cost inference of LLM; and iv) JavaScript and TypeScript, used to implement auxiliary scripts and plugins (as Realtime Plugin), enabling structured communication between the Bubble frontend and backend services, and the interactions involving LLM-based analysis through Fireworks AI.

4.2 *SafeSecIoT Canvas*: Project Planning and Elicitation

SafeSecRETS implements the *SafeSecIoT Canvas* model [15, 16], which comprises 20 building blocks for project planning, scope definition, and requirements elicitation, as illustrated in Figure 4 (i). These blocks are organized into five categories based on fundamental questions, offering a logical structure for planning. Blocks within a category are interconnected and may also relate to others, supporting the identification of inconsistencies and understanding of change impacts across the project [16].

The first group, Project Rationale, addresses *why* the project is being undertaken. The second, Project Scope, defines *what* must be delivered and is subdivided into three parts: product, system requirements, IoT-specific elements, and safety & security considerations. The third group, Stakeholders and Team, identifies *who* is involved and their roles. The fourth, Planning Parameters, defines *how* the project will be executed, including delivery phases and constraints. The fifth, Management Parameters, addresses *when* deliverables are due, *how much* the project will cost, and potential risks and uncertainties. Each block includes guidelines and shows its relationships with other blocks, with direct navigation links to improve user experience.

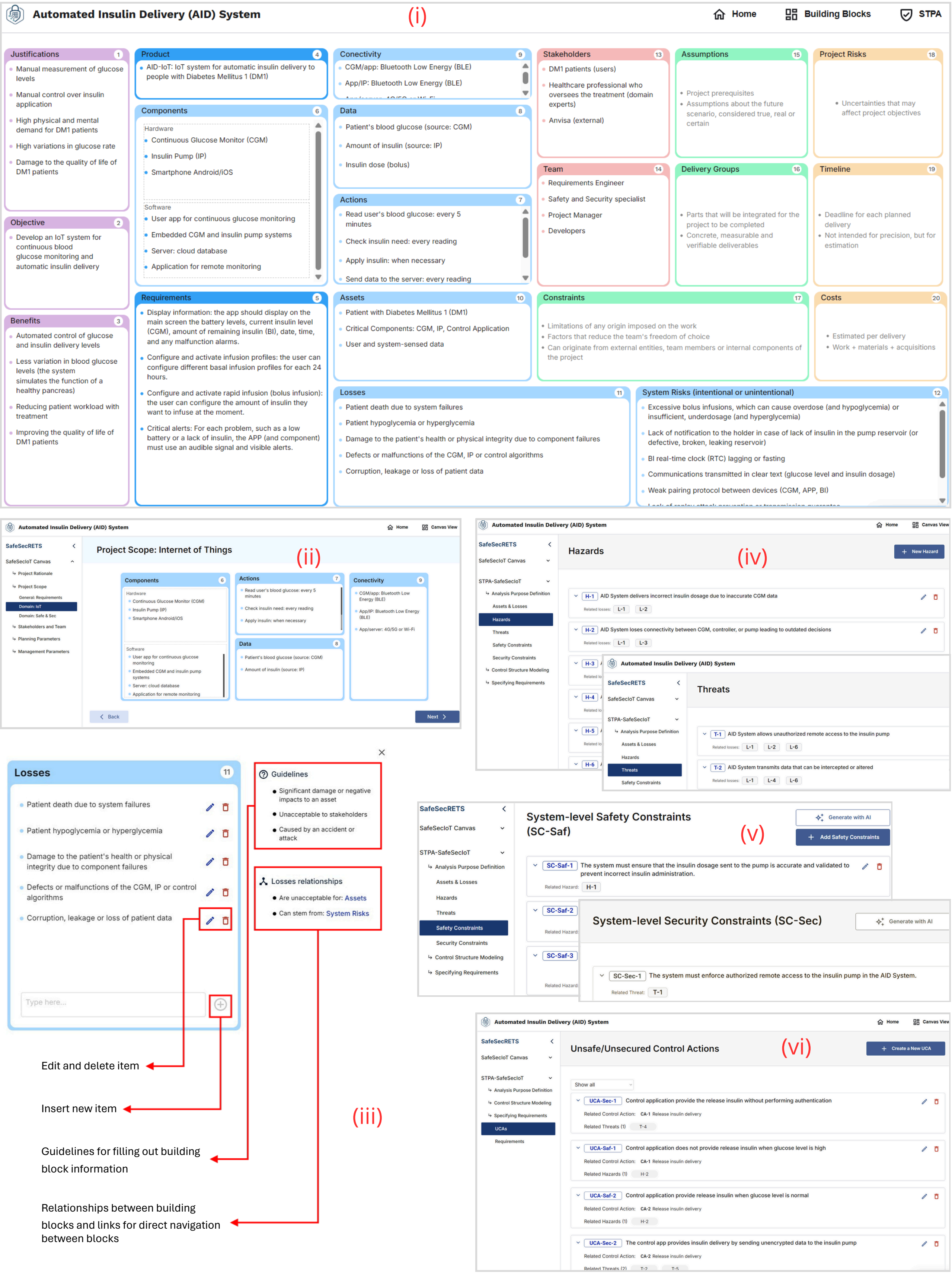
By structuring project planning around interconnected visual components, the canvas model supports stakeholders in identifying, organizing, and reasoning about general and domain-specific concerns. It includes traditional project aspects and specialized dimensions such as IoT elements and safety and security, which are essential in critical contexts. The canvas is an entry point to elicit and structure key information that feeds into more detailed safety and security STPA-based analysis. As such, the *SafeSecIoT Canvas* bridges the gap between strategic planning and technical requirements engineering, enabling *SafeSecRETS* to offer a unified workflow grounded in methodological rigor and visual guidance [16].

4.3 *STPA-SafeSecIoT*: Analysis and Specification

To support the analysis and specification of safety and security requirements based on the strategic information defined in the canvas, *SafeSecRETS* implements the *STPA-SafeSecIoT* method in a visual and structured manner. This extension adapts the original STPA approach to address safety and security concerns in IoT systems by integrating traditional hazard analysis with threat modeling.

The tool guides users through the steps of *STPA-SafeSecIoT* systematically and interactively: 1) identifying potential losses, and system-level hazards, and threats; 2) modeling the control structure, including controllers, their responsibilities, control actions, and feedback; and 3) deriving unsafe or unsecured control actions (UCAs) and corresponding constraints (safety and security requirements). Each step is supported by visual components representing the method's core entities and their interrelationships.

By linking the canvas planning blocks to the corresponding *STPA-SafeSecIoT* entities, *SafeSecRETS* ensures traceability between strategic decisions and technical analysis. For example, the IoT and safety/security concerns defined during planning are reused to establish the definition of hazards, threats, controllers, actions, and feedback. Visual cues and navigation features assist users in the stages of the process implemented by the tool, fulfilling the analysis objectives until obtaining detailed safety and security requirements, promoting consistency and completeness throughout the analysis.



Automated Insulin Delivery (AID) System

Justifications 1
- Manual measurement of glucose levels
- Manual control over insulin application
- High physical and mental demand for DM1 patients
- High variations in glucose rate
- Damage to the quality of life of DM1 patients

Objective 2
- Develop an IoT system for continuous blood glucose monitoring and automatic insulin delivery

Benefits 3
- Automated control of glucose and insulin delivery levels
- Less variation in blood glucose levels (the system simulates the function of a healthy pancreas)
- Reducing patient workload with treatment
- Improving the quality of life of DM1 patients

Product 4
- AID-IoT: IoT system for automatic insulin delivery to people with Diabetes Mellitus 1 (DM1)

Components 6
Hardware
- Continuous Glucose Monitor (CGM)
- Insulin Pump (IP)
- Smartphone Android/iOS

Software
- User app for continuous glucose monitoring
- Embedded CGM and insulin pump systems
- Server: cloud database
- Application for remote monitoring

Requirements 5
- Display information: the app should display on the main screen the battery levels, current insulin level (CGM), amount of remaining insulin (BI), date, time, and any malfunction alarms.
- Configure and activate infusion profiles: the user can configure different basal infusion profiles for each 24 hours.
- Configure and activate rapid infusion (bolus infusion): the user can configure the amount of insulin they want to infuse at the moment.
- Critical alerts: For each problem, such as a low battery or a lack of insulin, the APP (and component) must use an audible signal and visible alerts.

Conectivity 9
- CGM/app: Bluetooth Low Energy (BLE)
- App/IP: Bluetooth Low Energy (BLE)
- App/server: 4G/5G or Wi-Fi

Data 8
- Patient's blood glucose (source: CGM)
- Amount of insulin (source: IP)
- Insulin dose (bolus)

Actions 7
- Read user's blood glucose: every 5 minutes
- Check insulin need: every reading
- Apply insulin: when necessary
- Send data to the server: every reading

Assets 10
- Patient with Diabetes Mellitus 1 (DM1)
- Critical Components: CGM, IP, Control Application
- User and system-sensed data

Stakeholders 13
- DM1 patients (users)
- Healthcare professional who oversees the treatment (domain experts)
- Anvisa (external)

Team 14
- Requirements Engineer
- Safety and Security specialist
- Project Manager
- Developers

Assumptions 15
- Project prerequisites
- Assumptions about the future scenario, considered true, real or certain

Delivery Groups 16
- Parts that will be integrated for the project to be completed
- Concrete, measurable and verifiable deliverables

Constraints 17
- Limitations of any origin imposed on the work
- Factors that reduce the team's freedom of choice
- Can originate from external entities, team members or internal components of the project

Project Risks 18
- Uncertainties that may affect project objectives

Timeline 19
- Deadline for each planned delivery
- Not intended for precision, but for estimation

Costs 20
- Estimated per delivery
- Work + materials + acquisitions

Losses 11
- Patient death due to system failures
- Patient hypoglycemia or hyperglycemia
- Damage to the patient's health or physical integrity due to component failures
- Defects or malfunctions of the CGM, IP or control algorithms
- Corruption, leakage or loss of patient data

System Risks (intentional or unintentional) 12
- Excessive bolus infusions, which can cause overdose (and hypoglycemia) or insufficient, underdosage (and hyperglycemia)
- Lack of notification to the holder in case of lack of insulin in the pump reservoir (or defective, broken, leaking reservoir)
- BI real-time clock (RTC) lagging or fasting
- Communications transmitted in clear text (glucose level and insulin dosage)
- Weak pairing protocol between devices (CGM, APP, BI)
- Lack of replay attack prevention or transmission guarantee

Automated Insulin Delivery (AID) System

SafeSecRETS
SafeSecIoT Canvas
- Project Rationale
- Project Scope
  - General: Requirements
  - Domain: IoT
  - Domain: Safe & Sec
- Stakeholders and Team
- Planning Parameters
- Management Parameters

Project Scope: Internet of Things

Components 6
Hardware
- Continuous Glucose Monitor (CGM)
- Insulin Pump (IP)
- Smartphone Android/iOS

Software
- User app for continuous glucose monitoring
- Embedded CGM and insulin pump systems
- Server: cloud database
- Application for remote monitoring

Actions 7
- Read user's blood glucose: every 5 minutes
- Check insulin need: every reading
- Apply insulin: when necessary

Data 8
- Patient's blood glucose (source: CGM)
- Amount of insulin (source: IP)

Conectivity 9
- CGM/app: Bluetooth Low Energy (BLE)
- App/IP: Bluetooth Low Energy (BLE)
- App/server: 4G/5G or Wi-Fi

Losses 11
- Patient death due to system failures
- Patient hypoglycemia or hyperglycemia
- Damage to the patient's health or physical integrity due to component failures
- Defects or malfunctions of the CGM, IP or control algorithms
- Corruption, leakage or loss of patient data

Guidelines
- Significant damage or negative impacts to an asset
- Unacceptable to stakeholders
- Caused by an accident or attack

Losses relationships
- Are unacceptable for: Assets
- Can stem from: System Risks

Automated Insulin Delivery (AID) System

SafeSecRETS
SafeSecIoT Canvas
STPA-SafeSecIoT
- Analysis Purpose Definition
  - Assets & Losses
  - Hazards
  - Threats
  - Safety Constraints
  - Security Constraints
- Control Structure Modeling
- Specifying Requirements

Hazards

- H-1 AID System delivers incorrect insulin dosage due to inaccurate CGM data
  Related losses: L-1 L-2
- H-2 AID System loses connectivity between CGM, controller, or pump leading to outdated decisions
  Related losses: L-1 L-3
- H-3
- H-4
- H-5
- H-6

Threats

- T-1 AID System allows unauthorized remote access to the insulin pump
  Related losses: L-1 L-2 L-6
- T-2 AID System transmits data that can be intercepted or altered
  Related losses: L-1 L-4 L-6

System-level Safety Constraints (SC-Saf)

- SC-Saf-1 The system must ensure that the insulin dosage sent to the pump is accurate and validated to prevent incorrect insulin administration.
  Related Hazard: H-1
- SC-Saf-2
  Related Hazard:
- SC-Saf-3
  Related Hazard:

System-level Security Constraints (SC-Sec)

- SC-Sec-1 The system must enforce authorized remote access to the insulin pump in the AID System.
  Related Threat: T-1

Automated Insulin Delivery (AID) System

SafeSecRETS
SafeSecIoT Canvas
STPA-SafeSecIoT
- Analysis Purpose Definition
- Control Structure Modeling
- Specifying Requirements
  - UCAs
  - Requirements

Unsafe/Unsecured Control Actions

- UCA-Sec-1 Control application provide the release insulin without performing authentication
  Related Control Action: CA-1 Release insulin delivery
  Related Threats (1) T-4
- UCA-Saf-1 Control application does not provide release insulin when glucose level is high
  Related Control Action: CA-1 Release insulin delivery
  Related Hazards (1) H-2
- UCA-Saf-2 Control application provide release insulin when glucose level is normal
  Related Control Action: CA-2 Release insulin delivery
  Related Hazards (1) H-2
- UCA-Sec-2 The control app provides insulin delivery by sending unencrypted data to the insulin pump
  Related Control Action: CA-2 Release insulin delivery
  Related Threats (2) T-2 T-5

Figure 4: *SafeSecRETS* tool user interface: (i) the complete *SafeSecIoT Canvas* view for a project; (ii) detailed view of a group within the Project Scope; (iii) detailed view of a building block; and (iv, v, vi) view of STPA-based analysis.

5 Tool Demonstration

To demonstrate the capabilities of *SafeSecRETS*, we used the tool to plan a critical IoT system for AID system [11], as shown in Figure 4. These systems connect a control application to a continuous glucose monitor (CGM) and an insulin pump (IP) [3], automatically adjusting insulin doses to regulate blood glucose levels in patients with type 1 diabetes. Given their modular hardware and software architecture, AIDs qualify as critical IoT systems that present significant safety and security challenges. Precision and reliability are crucial for patient health, while failures or cyber-attacks may compromise data integrity or lead to serious harm.

Upon logging in *SafeSecRETS*, the user is directed to the Home screen, where they can create or access projects and share them with other users via email. Selecting a project opens the full canvas view (Figure 4, [i]), which organizes building blocks by fundamental questions and suggests a structured sequence for completion. In the figure we present the completed canvas to demonstrate the example of use in the AID system project.

Effectively planning and analyzing such systems requires close collaboration between a safety and security requirements engineer and a domain expert with knowledge of diabetes management and AID technology. The definition of the system scope, identification of critical functions, and specification of safety and security requirements depend on a shared understanding of technical, clinical, and operational aspects. *SafeSecRETS* facilitates this joint effort by providing a structured and visual environment where both specialists can iteratively contribute to the project scope definition.

Clicking on a specific block redirects the user to its corresponding group (Figure 4, [ii]), allowing them to complete or modify that block as well as any related elements. When editing a building block (Figure 4, [iii]), users are provided with contextual guidelines and references to related blocks (navigation shortcuts) to support clarity and consistency in the input process. Through interconnected blocks, visual components, and real-time updates on canvas, *SafeSecRETS* supports shared decision-making and helps align strategic goals with technical constraints and risk considerations.

In this project planning phase, informed by the domain expert's knowledge, the tool enables the extraction of critical information about the IoT system's architecture, such as its components, interconnections, performed actions, and data flows (Figure 4, [ii]). This step supports a shared understanding of the system's behavior and structure. In parallel, the tool guides the identification of system assets, the potential losses associated, and the risks that could lead to such losses. These elements establish the strategic foundation required to initiate the STPA-based analysis.

The second part of the tool, *STPA-SafeSecIoT* analysis, starts by validating *Losses* and their *Assets* association, and defining *Hazards* and *Threats* based on the previously identified Losses (Figure 4, [iv]). Based on this input, the tool AI assistant can help generate high-level *System-level Safety or Security Constraints* (Figure 4, [v]).

The AI assistant receives a prompt based on a structured template that defines the reference input and the expected output format. Since both the input and output are simple and highly controlled, tests showed no signs of hallucination. Multiple tests, adjustments, and refinements were conducted to arrive at the final version of the prompt presented next.

```
content: "Transform a hazard described in the format:
<Hazard> = <System> <Unsafe Condition>
into a safety constraint in the format:
<System-level Safety Constraint> = <System> <Condition to Enforce>
Your response must begin with 'The system must...'
and be written in technical, formal English. No explanations."
```

For modeling the control structure, the hardware and software components identified in the canvas are reused to instantiate the system controllers and assign their responsibilities. Similarly, *Control Actions* and *Feedback* are derived from the actions and data defined in the IoT project scope. The control structure generated by *SafeSecRETS* for the AID system is presented in Figure 5.

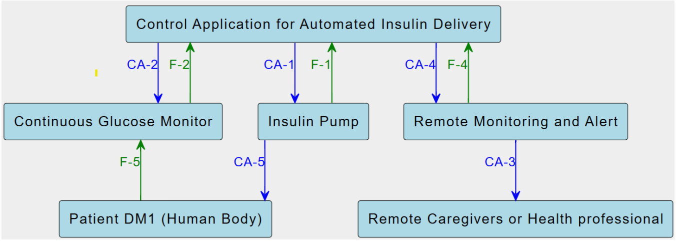

Control Application for Automated Insulin Delivery
CA-2
F-2
CA-1
F-1
CA-4
F-4
Continuous Glucose Monitor
Insulin Pump
Remote Monitoring and Alert
F-5
CA-5
CA-3
Patient DM1 (Human Body)
Remote Caregivers or Health professional

Figure 5: Control structure generated for AID system.

With the control structure established, the tool supports the analysis of *Unsafe and Unsecured Control Actions* (UCAs), based on how specific control actions could lead to hazards or threats under certain system conditions (Figure 4, [vi]). Finally, the AI assistant employs structured prompts to derive the relevant *Safety and Security Requirements* for each identified UCA. This process ensures that requirements are grounded in formal analysis and traceable to strategic and operational elements defined at the planning stage.

6 Final Remarks

This paper presented *SafeSecRETS*, a safety and security requirements tool for critical IoT systems. Its visual framework fosters consistency, supports early risk identification, and connects collaborative strategic planning to technical analysis. *SafeSecRETS* incorporates an extension of STPA tailored for safety and security co-analysis. Integrated features like LLM-assisted constraint generation enhance user guidance and reduce manual effort.

An automatic insulin delivery system case study highlighted the tool's capabilities in a critical project scenario, integrating project planning and requirements elicitation, analysis, and specification into a unified workflow. Future work includes expanding analytical features, improving usability for diverse user profiles, and conducting empirical studies to assess this tool's adoption and effectiveness in industry projects.

ARTIFACT AVAILABILITY

The authors declare that the research artifacts supporting the findings of this study are accessible at: https://doi.org/10.5281/zenodo.17000378

ACKNOWLEDGMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. Our thanks to Karlla Lima for her technical contribution to the tool's development.

REFERENCES

[1] Asim Abdulkhaleq and Stefan Wagner. 2015. XSTAMPP: an eXtensible STAMP platform as tool support for safety engineering. *Online Publications of University Stuttgart* (2015), 1–4. https://doi.org/10.18419/opus-3533

[2] Luigi Atzori, Antonio Iera, and Giacomo Morabito. 2010. The Internet of Things: A survey. *Computer Networks* 54, 15 (2010), 2787–2805. https://doi.org/10.1016/j.comnet.2010.05.010

[3] Sunil Deshpande, Jordan E Pinsker, Stamatina Zavitsanou, Dawei Shi, Randy Tompot, Mei Mei Church, Camille Andre, Francis J Doyle III, and Eyal Dassau. 2019. Design and Clinical Evaluation of the Interoperable Artificial Pancreas System (iAPS) Smartphone App: Interoperable Components with Modular Design for Progressive Artificial Pancreas Research and Development. *Diabetes Technology & Therapeutics* 21, 1 (2019), 35–43. https://doi.org/10.1089/dia.2018.0278

[4] José Finocchio-Júnior. 2013. *Project Model Canvas: Gerenciamento de Projetos sem Burocracia*. Elsevier Brasil. http://pmcanvas.com.br/livro/

[5] Ivo Friedberg, Kieran McLaughlin, Paul Smith, David Laverty, and Sakir Sezer. 2017. STPA-SafeSec: Safety and security analysis for cyber-physical systems. *Journal of Information Security and Applications* 34 (2017), 183–196. https://doi.org/10.1016/j.jisa.2016.05.008

[6] Jon Arne Glomsrud and J Xie. 2019. A Structured STPA Safety and Security Co-analysis Framework for Autonomous Ships. In *29th European Safety and Reliability conference*. 38–45. https://doi.org/10.3850/978-981-11-2724-3_0105-cd

[7] Alojz Gomola and Ingrid Bouwer Utne. 2024. A novel STPA approach to software safety and security in autonomous maritime systems. *Heliyon* 10, 10 (30 May 2024), 1–34. https://doi.org/10.1016/j.heliyon.2024.e31483

[8] Nancy G Leveson and John Thomas. 2018. *STPA Handbook*. MIT. http://psas.scripts.mit.edu/home/get_file.php?name=STPA_Handbook.pdf

[9] Xiaorong Lyu, Yulong Ding, and Shuang-Hua Yang. 2019. Safety and security risk assessment in cyber-physical systems. *IET Cyber-Physical Systems: Theory & Applications* 4, 3 (2019), 221–232. https://doi.org/10.1049/iet-cps.2018.5068

[10] Quelita Ribeiro and Jaelson Freire Brelaz de Castro. 2022. Safety & Security Alignment in Requirements Engineering Process for Autonomous Vehicles. In *Proceedings of the Workshop on Requirements Engineering (WER '22)*. 1–10. https://doi.org/10.29327/1298262.25-25

[11] Jennifer L Sherr, Lutz Heinemann, G Alexander Fleming, Richard M Bergenstal, Daniela Bruttomesso, Hélène Hanaire, Reinhard W Holl, John R Petrie, Anne L Peters, and Mark Evans. 2023. Automated insulin delivery: benefits, challenges, and recommendations. A Consensus Report of the Joint Diabetes Technology Working Group of the European Association for the Study of Diabetes and the American Diabetes Association. *Diabetologia* 66, 1 (Jan. 2023), 3–22. https://doi.org/10.1007/s00125-022-05744-z

[12] Fellipe G.R. Souza, Daniel P. Pereira, Rodrigo M. Pagliares, Simin Nadjm-Tehrani, and Celso M. Hirata. 2019. WebSTAMP: a Web Application for STPA & STPA-Sec. *MATEC Web of Conferences, ICSC-ESWC 2018* 273 (2019), 1–12. https://doi.org/10.1051/matecconf/201927302010

[13] Ernesto Fonseca Veiga and Renato Freitas Bulcão Neto. 2023. Toward a Method for Safety and Security Requirements Alignment in Critical IoT Systems. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering* (Campo Grande, Brazil) *(SBES '23)*. ACM, 452–457. https://doi.org/10.1145/3613372.3613373

[14] Ernesto Fonseca Veiga, Taciana Novo Kudo, and Renato Freitas Bulcão Neto. 2024. Linking Agile Planning and Safety and Security Analysis in Critical IoT Systems: An Approach based on ISO/IEC/IEEE 15288. In *Proceedings of the XXIII Brazilian Symposium on Software Quality (SBQS '24)*. ACM, 81–91. https://doi.org/10.1145/3701625.3701648

[15] Ernesto Fonseca Veiga, Taciana Novo Kudo, and Renato Freitas Bulcão-Neto. 2024. A Canvas Metamodel to Bridging Agile Project Planning and Requirements Engineering. In *Proceedings of the Workshop on Requirements Engineering (WER '24)*. 1–14. https://doi.org/10.29327/1407529.27-18

[16] Ernesto Fonseca Veiga, Karlla Loane Santos Lima, Taciana Novo Kudo, and Renato Freitas Bulcão-Neto. 2025. SafeSecRETS: A Project Planning Tool for Critical IoT Systems. In *Proceedings of the Workshop on Requirements Engineering (WER '25)*. 1–8.

[17] Xiang-Yu Zhou, Zheng-Jiang Liu, Feng-Wu Wang, and Zhao-Lin Wu. 2021. A system-theoretic approach to safety and security co-analysis of autonomous ships. *Ocean Engineering* 222 (2021). https://doi.org/10.1016/j.oceaneng.2021.108569