Thesis presented to the Instituto Tecnológico de Aeronáutica, in partial fulfillment of the requirements for the degree of Doctor of Science in the Graduate Program of Aeronautical and Mechanical Engineering, Field of Aeronautical Design, Aerospace Systems and Structures.

Guilherme Micheli Bedini Moreira

USE OF LARGE LANGUAGE MODELS TO SUPPORT AEROSPACE DEFENSE SYSTEMS ENGINEERING

Thesis approved in its final version by signatories below:

gov.br
Documento assinado digitalmente
WILLER GOMES DOS SANTOS
Data: 27/02/2025 13:32:43-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Willer Gomes dos Santos
Advisor

gov.br
Documento assinado digitalmente
CHRISTOPHER SHNEIDER CERQUEIRA
Data: 27/02/2025 14:36:57-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Christopher Shneider Cerqueira
Co-advisor

Campo Montenegro
São José dos Campos, SP – Brazil
2025

Cataloging-in Publication Data
Documentation and Information Division

Moreira, Guilherme Micheli Bedini
Use of large language models to support aerospace defense systems engineering / Guilherme Micheli Bedini Moreira.
São José dos Campos, 2025.
207 f.

Thesis of Doctor of Science – Graduate Program of Aeronautical and Mechanical Engineering, Field of Aeronautical Design, Aerospace Systems and Structures – Instituto Tecnológico de Aeronáutica, 2025. Advisor: Prof. Dr. Willer Gomes dos Santos; co-advisor: Prof. Dr. Christopher Shneider Cerqueira.

1. Sistemas aeroespaciais. 2. Elicitação de requisitos. 3. Inteligência artificial. 4. Engenharia de sistemas. I. Instituto Tecnológico de Aeronáutica. II. Title.

BIBLIOGRAPHIC REFERENCE

MOREIRA, Guilherme Micheli Bedini. **Use of large language models to support aerospace defense systems engineering**. 2025. 207 f. Thesis (Doctor of Aeronautical and Mechanical Engineering) – Instituto Tecnológico de Aeronáutica, São José dos Campos, 2025.

CESSION OF RIGHTS

AUTHOR'S NAME: Guilherme Micheli Bedini Moreira
PUBLICATION TITLE: Use of Large Language Models to Support Aerospace Defense Systems Engineering
PUBLICATION KIND/YEAR: Thesis / 2025

_____

Guilherme Micheli Bedini Moreira
Rua H25C, 112 – Campus do CTA
CEP: 12228-560, São José dos Campos – SP

USE OF LARGE LANGUAGE MODELS TO SUPPORT AEROSPACE DEFENSE SYSTEMS ENGINEERING

Guilherme Micheli Bedini Moreira

Composição da Banca Examinadora:

Profa Dra	Emília Villani	Chairperson	–	ITA
Prof Dr	Willer Gomes dos Santos	Advisor	–	ITA
Prof Dr	Christopher Shneider Cerqueira	Co-advisor	–	ITA
Profa Dra	Juliana de Melo Bezerra	Internal Member	–	ITA
Dra	Ana Maria Ambrosio	External Member	–	INPE
Dr	Geilson Loureiro	External Member	–	INPE

ITA

I dedicate this work to my parents, my wife, and my children – past, present, and future.

Acknowledgements

Permitirei-me expressar minha graditão em minha língua materna. É difícil agradecer a tanta gente que passou pelo meu caminho e me ajudou a chegar até este ponto. Não obstante, gostaria de tornar público o meu enorme sentimento de gratidão às pessoas que citarei adiante. Não poderia deixar de começar por meus genitores, que deram tudo de si para me proporcionar uma educação adequada. Pai, Mãe, obrigado por vosso amor e exemplo.

Entre os irmãos, preciso agradecer ao de sangue e àqueles que escolhi ao longo da vida: William Moreira, Mohand Tomé, Antônio Mariz e Levi Nepomuceno. Obrigado por sempre estarem comigo. Eu sempre estarei com vocês!

Faz-se mister reconhecer a enorme contribuição que Levi Araújo e Luiz Gustavo Muniz proporcionaram à manutenção da minha saúde mental. Valeu, Space Jam! Dos bancos acadêmicos, preciso agradecer meu grande parceiro nesta caminhada, Daniel Rondon Pleffken.

Preciso também demonstrar gratidão pelo enorme voto de confiança que me deu o Brig César Demétrio, ex-Diretor do IAE. Obrigado por acreditar em mim!

É necessário também reconhecer o quanto este trabalho foi influenciado pelas ricas experiências vivenciadas junto às competentes equipes do IFI (em especial a CPA) e do Projeto IFFM4BR, do IAE. Muito obrigado, amigos.

Na área do acompanhamento acadêmico-militar, agradeço imensamente a todos os membros da equipe do PPGAO: obrigado por todo o suporte!

Gostaria também de agradecer ao enorme voto de confiança e oportunidade de aprendizado dados pelo Cel George Luiz Guedes De Oliveira, permitindo que eu realizasse a função de Vice Pró-Reitor de Administração do meu querido Instituto Tecnológico de Aeronáutica. E por falar em ITA, gostaria de externar também minha enorme gratidão à senhora Vice-Reitora, Profª Emilia Villani, pela grande oportunidade dada a mim de contribuir para a implantação do meu querido ITA, no meu querido Ceará. E ainda pela incrível experiência de coordenar o Curso de Especialização em Operações de Sistemas Espaciais. Guardarei tudo isso no coração.

Atualmente, é quase impossível realizar uma pesquisa na área de Inteligência artificial sem dados. Com meu doutorado, não foi diferente. Por isso, é fundamental também

reconhecer a enorme contribuição para esta pesquisa científica realizada pelo Diretor do IFI, Cel Marcelo Zawadzki, e seu Assessor-Técnico, o colega de turma na AFA TCel Gustavo Basílio. Da mesma forma, foi também importantíssima a contribuição de outro colega de turma, desta vez do ITA, o Engenheiro Danilo Miranda, em nome de quem agradeço toda a Mac Jee, pujante empresa brasileira de defesa, que cedeu documentação técnica de produtos reais para servirem de estudo de caso nesta pesquisa. E por falar em pesquisa, agradeço a todos os profissionais especialistas em Engenharia de Sistemas e Certificação de sistemas aeroespaciais que participaram dos *surveys* e análises conduzidos neste trabalho.

Pelo lado acadêmico, gostaria de agradecer a TODOS os professores/instrutores que já passaram pela minha vida. Foram muitos, e todos muito importantes. Mas gostaria de agradecer em especial ao Prof Marcelo Lopes de Oliveira e Souza (ITA-76). Entre todos os ofícios que uma criança pode sonhar em exercer, “cientista” foi uma das respostas mais frequentes durante minha infância e adolescência. Obrigado por despertar meu espírito científico e investigativo, que andava adormecido. Obviamente, nessa seara também não posso deixar de reconhecer a enorme relevância de meus orientadores para esta jornada. Obrigado Prof Christopher Cerqueira, por toda parceria, autonomia e confiança depositada. Obrigado Prof Willer Santos, pelos conselhos, empenho, e todo o rigor ao método, fundamental à ciência e à Academia.

E deixo para o final o agradecimento mais importante de todos: minha amada esposa, Luciana Moreira, rainha da minha morada, que luta ao meu lado todos os dias e que me mostra o real significado do amor. Você deu-me o maior presente de todos: Julia e Leo. Nossa família é minha maior fonte inspiração. Eu amo vocês.

" Live as if you were to die tomorrow. Learn as if you were to live forever".

Mahatma Gandhi

Resumo

Esta pesquisa investiga a integração de Modelos de Linguagem de Grande Escala (LLMs) na engenharia de sistemas aeroespaciais de defesa para automatizar dois processos críticos: a elicitação de requisitos por meio da System Theoretic Process Analysis (STPA) e a atribuição de Means of Compliance (MoCs) a requisitos de sistemas aeroespaciais de defesa. A motivação reside em abordar a natureza trabalhosa e propensa a erros dos métodos tradicionais, que dependem fortemente da expertise humana. O estudo avalia especificamente a viabilidade e o desempenho de LLMs, como GPT-3.5 e GPT-4, quando guiados por técnicas avançadas de Prompt Engineering e metodologias de fine-tuning. Essas abordagens buscam manter ou superar a precisão e a qualidade tipicamente alcançadas por especialistas da área. O problema investigado é a ineficiência e a variabilidade dos processos manuais de engenharia de requisitos e conformidade, que são críticos em sistemas aeroespaciais de defesa devido às rigorosas demandas de segurança e confiabilidade. Utilizando como estudo de caso um Veículo Aéreo de Combate Não Tripulado (UCAV) hipotético, a pesquisa se situa no contexto da Força Aérea Brasileira (FAB), onde esses desafios são particularmente relevantes. A metodologia envolve a automação da Fase 1 da STPA por meio de prompts personalizados para gerar requisitos de sistema e o treinamento de um modelo ajustado para atribuir MoCs com precisão. O desempenho foi comparado a dados de sistemas reais e à performance de especialistas da área. Os resultados destacam que LLMs guiados por *Prompt Engineering* podem gerar requisitos que atendem ou superam oito dos nove atributos de qualidade avaliados, incluindo verificabilidade, completude, clareza e modificabilidade. O modelo "tunado" 'gpt-3.5-turbo' alcançou uma precisão de 80,18% na atribuição de MoCs. Por fim, com técnicas apropriadas, foi possível gerar relatórios de *safety assessment* como PHAs (*Preliminary Hazard Analysis*) a partir da documentação técnica de produtos reais. As implicações desta pesquisa são profundas. Ao simplificar a elicitação de requisitos, a atribuição de MoCs, e a geração de relatórios de engenharia, os LLMs reduzem o tempo, o esforço e os custos associados aos processos de engenharia, mantendo elevados padrões de rigor e confiabilidade. Este trabalho avança a compreensão acadêmica sobre as aplicações de LLMs em sistemas críticos de segurança, introduz um framework escalável e replicável para integrar LLMs em fluxos de trabalho de engenharia e oferece ferramentas práticas para a indústria de defesa aeroespacial.

Abstract

This research investigates the integration of Large Language Models (LLMs) into aerospace defense systems engineering to automate two critical processes: eliciting requirements through System Theoretic Process Analysis (STPA) and assigning Means of Compliance (MoCs) to aerospace defense systems' requirements. The motivation lies in addressing the labor-intensive and error-prone nature of traditional methods, which heavily rely on human expertise. The study specifically evaluates the feasibility and performance of LLMs, such as GPT-3.5 and GPT-4, when guided by advanced Prompt Engineering techniques and fine-tuning methodologies. These approaches aim to maintain or surpass the accuracy and quality typically achieved by experts in the field. The problem under investigation is the inefficiency and variability of manual requirements engineering and compliance processes, which are critical in defense aerospace systems due to stringent safety and reliability demands. Using a hypothetical Unmanned Combat Air Vehicle (UCAV) as a case study, the study situates the research in the context of the Brazilian Air Force (FAB), where these challenges are particularly acute. The methodology involves automating Phase 1 of STPA through tailored prompts to generate system requirements and training a fine-tuned model to assign MoCs accurately. Performance was benchmarked against real-world system data and domain experts' outputs. The findings highlight that LLMs guided by Prompt Engineering can generate requirements that meet or exceed eight of nine evaluated quality attributes, including testability, completeness, clarity, and modifiability. The fine-tuned 'gpt-3.5-turbo' model achieved an 80.18% accuracy in MoC assignments. Finally, with appropriate techniques, it was possible to generate safety assessment reports such as PHAs (Preliminary Hazard Analysis) from the technical documentation of real products. The implications of this research are profound. By streamlining requirements elicitation, MoC assignment, and the generation of engineering reports, LLMs reduce the time, effort, and cost associated with engineering processes while maintaining high standards of rigor and reliability. This work advances academic understanding of LLM applications in safety-critical systems, introduces a scalable and replicable framework for integrating LLMs into engineering workflows, and offers practical tools to the aerospace defense industry.

List of Figures

List of Table

List of Abbreviations and Acronyms

AI	Artificial Intelligence
ASR	Alternative Systems Review
AVOP	*Avaliação Operacional*
BDD	Block Definition Diagram
CDR	Critical Design Review
CNN	Convolutional Neural Network
ConOps	Concept of Operations
DCA	Directive of the Aeronautics Command
DVFS	Dynamic Voltage and Frequency Scaling
EASA	European Union Aviation Safety Agency
EDA	Electronic Design Automation
EFB	Electronic Flight Bag
FAA	Federal Aviation Administration
FAB	*Força Aérea Brasileira*
FAT	Technical Analysis Form
FCA	Functional Configuration Audit
FHA	Functional Hazard Analysis
FMEA	Failure Modes and Effects Analysis
FMU	Fuze Management Unit
FTA	Fault Tree Analysis
FZU	Fuze Control Unit
GAC	*Grupo de Acompanhamento e Controle*
GCS	Ground Control Station
GRPO	Group Relative Policy Optimization

HCS	Hierarchical Control Structure
HDL	Hardware Description Language
HFE	Human Factors Engineering
IAE	*Instituto de Aeronáutica e Espaço*
IBD	Internal Block Diagram
IEA	International Energy Agency
IFI	*Instituto de Fomento e Coordenação Industrial*
INCOSE	International Council on Systems Engineering
ISO	International Organization for Standardization
ITA	*Instituto Tecnológico de Aeronáutica*
LLM	Large Language Model
LoRA	Low-Rank Adaptation
LSTM	Long Short-Term Memory
MBSE	Model-Based Systems Engineering
MCR	Mission Concept Review
MD	Ministry of Defense
ML	Machine Learning
MMLU	Massive Multitask Language Understanding
MoC	Means of Compliance
MoE	Mixture of Experts
NATO	North Atlantic Treaty Organization
NLP	Natural Language Processing
NOP	*Necessidade Operacional*
ODSA	*Órgão de Direção Setorial e de Assistência Direta do COMAER*
OECD	Organisation for Economic Co-operation and Development

OPL	Object-Process Language
OPM	Object-Process Methodology
PCA	Physical Configuration Audit
PDR	Preliminary Design Review
PHA	Preliminary Hazard Analysis
PPA	Power, Performance, and Area
PRISMA	Preferred Reporting Items for Systematic reviews and Meta-Analyses
PRR	Production Readiness Review
RAC	Risk Assessment Code
RAG	Retrieval-Augmented Generation
ReLU	Rectified Linear Unit
RL	Reinforcement Learning
RNN	Recurrent Neural Network
ROP	*Requisitos Operacionais*
RTLI	*Requisitos Técnicos Logísticos e Industriais*
SBPO	*Simpósio Brasileiro de Pesquisa Operacional*
SE	Systems Engineering
SFR	System Functional Review
SHA	System Hazard Analysis
SIGE	*Simpósio de Aplicações Operacionais em Áreas de Defesa*
SLR	Systematic Literature Review
SRR	System Requirements Review
STAMP	System-Theoretic Accident Model and Processes
STPA	System-Theoretic Process Analysis
SVR	System Verification Review

SysML	Systems Modeling Language
ToT	Tree-of-Thought
TPU	Tensor Processing Unit
TRR	Test Readiness Review
UAV	Unmanned Aerial Vehicle
UC	Use Case
UCA	Unsafe Control Action
UCAV	Unmanned Combat Air Vehicle
UML	Unified Modeling Language
V&V	Verification and Validation
XAI	Explainable AI

List of Symbols

$\in$	Belongs to
$\mathbb{R}$	Set of Real numbers
$[...]^T$	Matrix transposition operator
Σ	Summation
$\surd$	Square root

Contents

1 Introduction

The aerospace industry has experienced unprecedented growth and transformation over the past few decades (Diamandis; Kotler, 2020; Doganis, 2013; Schmitt, 2000). Modern aerospace products have become increasingly complex, highly integrated, and heavily reliant on software (Mcdonald *et al*., 2022; Moir; Seabridge, 2012). This evolution has heightened the need for meticulous design and rigorous verification processes to ensure these systems perform safely and effectively (Li *et al*., 2021). The increasing complexity is driven by advancements in technology, heightened safety and regulatory requirements, and the demand for more sophisticated functionalities (Brunton *et al*., 2021).

As aerospace systems evolve, integrating multiple subsystems and the extensive use of software has become commonplace, making it imperative that designs are correct from the outset (Moir; Seabridge, 2012). Errors in the early stages of design can propagate through the development process, leading to significant issues in later stages, including increased costs, delays, and potential safety hazards (Aloisio, 2019).

In this context, the importance of high-quality requirements cannot be overstated. Requirements are the foundation upon which the entire lifecycle of a product is built (Saaksvuori; Immonen, 2008). Incomplete requirements are the main reason why projects fail (Alexander; Stevens, 2002). This happens because requirements guide the design, development, testing, and certification processes, ensuring that the final product meets all necessary specifications and regulatory standards. This is particularly critical in defense aerospace systems, where the stakes are exceptionally high. Defense aerospace systems must operate under stringent conditions and meet rigorous performance, reliability, and safety standards (Boyer *et al*., 2015). Any failure in these systems can have catastrophic consequences, making the quality of requirements a vital concern.

Traditionally, the process of generating, verifying, and validating requirements has been manual and labor-intensive, involving extensive collaboration among multidisciplinary teams (Periaux *et al*., 2015). This process, while thorough, is also prone to human error and inconsistencies. As the complexity and scale of aerospace systems continue to grow, the traditional methods of requirement generation and verification are becoming increasingly untenable (Moir; Seabridge, 2012). There is a pressing need for techniques that can automate and enhance the reliability of these repetitive tasks, ensuring consistency, accuracy, and

efficiency in the verification process.

Other stages of the systems verification process are equally laborious and challenging. For instance, defining the Means of Compliance (MoCs) – the methods used to ensure the fulfillment of aerospace requirements – is a task that demands significant engagement from systems engineering teams. Additionally, due to the nature of aerospace products, the safety analyses required for certification also necessitate a substantial engineering effort, especially when considering how complex, highly integrated, and software-dependent these systems have become (RTCA, 2011; SAE International, 2010).

The advent of Artificial Intelligence (AI) offers a promising solution to this challenge (Dong, 2019; Dong *et al*., 2023; Tondji; Ghazi; Mihaela Botez, 2024). Large Language Models (LLMs) have demonstrated remarkable capabilities in understanding and generating human-like text, making them ideal candidates for automating various aspects of the requirement generation and verification process. This doctoral research aims to harness the power of LLMs to automate the generation of requirements from already established techniques such as System Theoretic Process Analysis (STPA) (Leveson, 2016), assign the corresponding MoCs automatically, and generate safety reports based on real aerospace project documents.

1.1 Contextualization

The automation of requirements engineering and certification processes for aerospace defense systems has evolved significantly over the past decades, driven by the increasing complexity and safety demands of these systems (Depauw *et al*., 2024). Traditionally, requirements for such systems were manually derived through structured analysis methods, such as Fault Tree Analysis (FTA) and Failure Modes and Effects Analysis (FMEA) (Martins; Gorschek, 2017). However, as systems grew more integrated and software-dependent, methodologies like STPA gained prominence due to their ability to address both technical and human factors in system safety. STPA’s focus on the interactions between system components marked a critical shift toward more holistic approaches in safety-critical systems (Leveson, 2016).

In recent years, there has been growing interest in automating aspects of the requirements engineering process. While Model-Based Systems Engineering (MBSE) has been widely adopted to manage the complexity of aerospace systems, few studies have fully

explored the integration of Natural Language Processing (NLP) technologies, particularly Large Language Models (LLMs), in generating and analyzing system requirements (Ray *et al*., 2023). Recent advancements in LLMs, such as the latest versions of ChatGPT, have introduced opportunities to automate the extraction of system requirements and safety analysis from textual data (Taramsari *et al*., 2024). However, there remains a significant gap in applying these technologies specifically to defense aerospace systems, where certification and safety analyses are particularly rigorous and demand high accuracy.

The significance of this research lies in its potential to enhance both the efficiency and accuracy of requirements engineering for aerospace defense systems. The manual effort involved in defining requirements, allocating Means of Compliance (MoCs), and generating safety documentation is immense, especially for Unmanned Combat Air Vehicles (UCAVs) and similar complex systems. Automating these processes can drastically reduce engineering workload and improve traceability and compliance with certification standards. This is particularly critical in defense sectors, where delays or errors in certification can have profound operational and financial impacts.

For instance, when I led a project team responsible for developing avionics at the Aeronautics and Space Institute (IAE), a multidisciplinary team of approximately ten specialists had to devote themselves to assigning the Means of Compliance (MoCs) for around 600 requirements over more than a week. The team included experts in hardware, software, integration, laboratory testing, systems engineering, and certification. We meticulously reviewed each requirement, engaged in discussions, and sometimes referred to relevant standards before determining the necessary MoCs to verify each requirement. This effort required over 500 valuable engineering man-hours.

The central challenge addressed by this research is the automation of critical processes traditionally performed manually by systems engineers, such as the derivation of requirements through STPA and the assignment of MoCs. Previous research has shown the feasibility of automating certain aspects of requirements management (Umar; Lano, 2024), but no comprehensive solution exists that integrates STPA with LLMs for generating safety-critical documents like Preliminary Hazard Analysis (PHA) and Functional Hazard Analysis (FHA). By leveraging datasets derived from compliance matrices of aerospace products, this research advances current methodologies by introducing an automated, LLM-driven approach for the generation and verification of system requirements and safety analyses.

In a practical context, this research has direct applications in the certification of aerospace defense systems, specifically in environments where compliance with rigorous safety standards is mandatory. The use of real certification documents from aerospace products certified by the Industrial Fostering and Coordination Institute (IFI), the Brazilian Military Aerospace Certification Authority, exemplifies the applicability of this research in a real-world setting. By demonstrating how ChatGPT-4o, paired with prompt engineering techniques, can automate the creation of safety documentation, this research showcases a practical solution that addresses both industry needs and certification bodies' requirements, contributing to the modernization of certification processes in defense aerospace.

1.2 Justification

The increasing complexity of aerospace defense systems, particularly in the context of safety-critical applications, has created an urgent need for more efficient, reliable, and automated processes to support certification and compliance activities (Ruiz *et al*., 2011). Traditional methods for generating system requirements, assigning MoCs, and conducting safety analyses are often labor-intensive and subject to human error (Thomas, 2013). Given the stringent safety and regulatory demands within the aerospace industry, there is a clear gap in the literature and practice regarding the automation of these processes using advanced machine learning models, such as LLMs.

This research addresses this gap by proposing an innovative methodology that leverages STPA and LLMs to automate the generation of system requirements, the attribution of MoCs, and the production of safety reports. By automating these critical tasks, the proposed approach has the potential to significantly reduce the time and resources required for aerospace certification processes while increasing accuracy and consistency compared to traditional, manual methods.

From a practical perspective, this study contributes to the aerospace defense industry by offering a tool to streamline certification workflows, thereby improving operational efficiency. Automating MoC assignments, in particular, has the potential to mitigate some bottlenecks often associated with regulatory compliance.

Furthermore, this research is timely, as it coincides with the growing adoption of artificial intelligence (AI) in high-stakes industries such as aerospace. Integrating LLMs into safety and compliance processes aligns with current trends in AI-driven automation and

presents a transformative opportunity for the aerospace sector (Depauw *et al*., 2024). It also addresses the broader push towards adopting AI technologies in domains where safety and compliance are paramount.

Considering the academic perspective of a doctoral thesis, according to (CAPES, 1965), a thesis should possess three fundamental characteristics:

1. **Originality** - the central idea must be unprecedented.
2. **Generality** - the concept demonstrated by the thesis should have scalability for diverse domains.
3. **Utility** - the work should be valuable to the scientific and/or engineering community.

This research fills a significant gap in the literature by contributing to the theoretical understanding of STPA and the practical application of AI in aerospace systems verification. It offers real-world solutions to ongoing challenges in aerospace certification.

As will be explored in the subsequent chapters, this study aims to combine techniques already utilized in engineering and science yet not previously addressed in a unified manner or with the same purpose presented here. Furthermore, the results have the full potential for application in more complex systems and even in domains beyond aerospace or defense. Lastly, this work is anticipated to demonstrate the utility of the tools employed in the development of complex systems and in approaching integration challenges.

1.3 Research Problem

Given the contextualized scenario outlined in Section 1.1, the following concern arises: "*How Large Language Models (LLMs) can automate the application of System Theoretic Process Analysis (STPA) to elicit aerospace defense systems' requirements, automatically assign their Means of Compliance (MoCs), and generate safety reports while maintaining or exceeding the current performance level of experts on the field*?"

To answer this question, the objectives of this research were outlined in Section 1.5, which guided the conduct of this work. The most modern candidate solutions were sought, always using the most up-to-date language models available at each research stage.

1.4 Hypothesis

Large Language Models (LLMs), when guided by Prompt Engineering Techniques for System Theoretic Process Analysis (STPA), can effectively automate the elicitation of aerospace defense systems' requirements, accurately assign Means of Compliance (MoCs) to these requirements through fine-tuning techniques, and generate reasonable safety reports' drafts, achieving or exceeding the performance accuracy of field experts in the aerospace defense industry.

1.5 Research Objectives

The main objective of this work is to investigate how Large Language Models (LLMs), through the application of Prompt Engineering and fine-tuning, can automate the process of eliciting aerospace defense systems' requirements via System Theoretic Process Analysis (STPA), accurately assign their respective Means of Compliance (MoCs), and generate safety reports with the aim of achieving expert-level performance in these tasks.

For this, the following specific objectives were outlined:

1. To develop and evaluate a Prompt Engineering methodology for guiding LLMs in the application of System Theoretic Process Analysis (STPA) to elicit requirements for aerospace defense systems, ensuring the method's relevance and accuracy when compared to established requirements of a real aerospace system in operation by the Brazilian Air Force (FAB).
2. To create a fine-tuned LLM model specifically trained for the automatic assignment of Means of Compliance (MoCs) to aerospace defense system requirements, utilizing a dataset of MoCs extracted from compliance matrices for certified aerospace defense products.
3. To benchmark the performance of the LLM-driven STPA approach against traditional expert-led requirement elicitation, measuring accuracy, relevance, and completeness in capturing critical safety and functional requirements.
4. To assess the fine-tuned LLM's MoC assignment accuracy by comparing it to the performance of domain experts, analyzing whether the model can meet or exceed expert-level accuracy in MoC assignments.

5. To analyze the limitations and potential improvements in using Prompt Engineering and fine-tuning approaches with LLMs in the aerospace domain, particularly in areas where automated methods may fall short or require further calibration to align with expert standards.
6. To develop a framework for generating comprehensive safety assessment reports, such as Preliminary Hazard Analysis (PHA) and Functional Hazard Analysis (FHA), by leveraging LLM-driven Prompt Engineering techniques and fine-tuning methods, ensuring the reports' structure, content, and depth align with aerospace industry standards and expert evaluations.

1.6 Discussions about the adoption of AI tools

One of the central themes explored in this chapter is the potential for AI-driven approaches to introduce new biases, both in the elicitation and validation processes, which may influence the quality and reliability of the research's outputs. LLMs are shaped by the datasets they are trained on and the engineering of their prompts, making their performance inherently susceptible to the biases embedded in these inputs. Understanding and mitigating these biases is essential for ensuring that AI applications do not perpetuate or exacerbate systemic flaws, particularly in sectors where safety is critical.

Closely related to the issue of bias is the need for greater transparency in the inner workings of AI systems. While LLMs like ChatGPT demonstrate remarkable capabilities in natural language processing and reasoning, their decision-making processes often resemble a "black box," making it challenging for users to ascertain the rationale behind specific outputs. This opacity raises significant concerns in safety-critical applications, where accountability and traceability are not just desirable but legally and ethically mandated. This chapter reflects on the methods used to enhance transparency in this research, including prompt engineering and iterative validation, and discusses how these practices can be expanded to foster trust in AI systems.

Finally, accountability emerges as a key topic of discussion. As AI systems assume more responsibilities traditionally managed by human experts, the question of who is accountable for the decisions and outcomes generated by these systems becomes increasingly relevant. This chapter explores the implications of using AI in highly regulated domains, emphasizing the importance of establishing clear accountability frameworks that balance the

benefits of automation with the ethical imperatives of human oversight.

By critically engaging with these themes, this chapter seeks to provide a holistic understanding of the findings and their implications. It also aims to contribute to the growing discourse on how AI can be responsibly deployed in safety-critical systems, offering insights and recommendations that extend beyond the technical scope of this research. Through this discussion, the potential of AI as a transformative tool in aerospace systems engineering will be examined alongside the ethical and practical considerations necessary to ensure its safe adoption.

1.6.1 Ethical Issues in the Use of AI

The rapid growth of AI technologies has sparked intense debates regarding their ethical implications. Tegmark (2017), in *Life 3.0: Being Human in the Age of Artificial Intelligence*, advocates for a balanced yet forward-thinking approach to AI development—one that recognizes both the immense potential of these technologies and the profound ethical responsibilities they entail. This Section endeavors to explore these ethical concerns by integrating perspectives from religion, evolutionary theory, and the principle that nature consistently seeks an optimal—or lowest-energy—state. By reflecting on the question of whether a supernatural Creator shapes the cosmos in a state of perfection, or whether nature evolves under adaptationist pressures, we can glean broader insights into how AI should be ethically engineered to align with optimal and sustainable principles.

1.6.2 Contrasting Perspectives on the Origins of Perfection

One perspective, grounded in the faith of over 80% of the world's population, posits the existence of a perfect divine entity that created the universe. If God is indeed perfect, it follows that the universe, as God's creation, embodies an intrinsic perfection. In this view, each facet of reality—from physical laws to biological processes—reflects a form of divine optimization. The ethical corollary for AI, then, is that human endeavors to create intelligent systems should strive to mirror this supposed divine order. Ethical guidelines, under this framework, might be viewed as a means of preserving and not distorting this divine perfection. In line with Tegmark's suggestion that AI development must focus on long-term societal well-being, those who hold a theistic worldview could argue that AI systems be designed to uphold moral values such as compassion, justice, and harmony—attributes often ascribed to the divine.

On the other hand, a massive proportion of scientists and philosophers either reject or remain agnostic about any supernatural underpinnings of the universe. Instead, they focus on evolutionary principles, emphasizing the role of adaptation and selection processes that shape living organisms. Despite the apparent complexity of life, many argue that nature's underlying processes can be described by fundamental physical laws that favor states of minimal or optimized energy. Within this framework, evolution is viewed not as a linear path toward "perfection" but rather as a dynamic adaptation that results in locally optimal solutions—organisms that fit their environments in ways that maximize survival and reproduction. For AI ethics, this perspective underscores the importance of designing algorithms that dynamically adapt to real-world complexity. As Tegmark and others have pointed out, AI should be robust, flexible, and responsive to feedback, much like evolutionary processes that incrementally refine organisms' adaptations over time.

1.6.3 The Principle of Nature's Optimization and AI

Regardless of one's stance on divinity, a unifying observation emerges: nature, in its fundamental workings, appears to favor energy minimization or optimization. From the principle of least action in physics to the incremental "gradient descent" in machine learning, the concept of seeking lower-energy states is pervasive. This natural inclination toward optimization provides a valuable design paradigm for AI systems. Indeed, the success of contemporary machine learning—particularly methods that rely on iterative error-based learning—mirrors how biological organisms learn and adapt through trial, error, and incremental improvement.

1. **Gradient Descent and Learning by Error**: In machine learning, gradient descent algorithms iteratively adjust parameters to minimize a cost (or error) function. This process is analogous to the way organisms refine behaviors to better suit their environments—learning from mistakes and gradually converging on viable strategies. Ethically, this invites us to ensure that the "cost function" of AI systems is properly aligned with societal values. If the system is not guided by well-defined, ethically sound metrics, it might reach an "optimal" state that is harmful or unjust. As Tegmark and other AI theorists such as Stuart Russell have emphasized, alignment of AI goals with human values is vital.
2. **Systems-Level Optimization vs. Local Minima**: While nature tends to favor states of energy minimization, it often settles into local minima, which may not

always be globally optimal in the grand scheme. Translating this lesson into AI design, we must be cautious about the risk of creating systems that become "stuck" in ethically questionable local optima—behaviors that seem efficient but violate fairness, privacy, or security. Researchers like Nick Bostrom highlight the existential risks of poorly aligned superintelligence: if an AI optimizes a metric unmoored from broader human values, it could produce catastrophic outcomes (Brundage, 2015). Thus, the principle of nature's optimization warns us to balance local efficiency gains with a more holistic consideration of universal ethical principles.

3. **Transparency and Interpretability**: In nature, the adaptive processes are visible over evolutionary timescales through fossils, observable mutations, and ecological shifts. By analogy, ethical AI requires transparency and interpretability. Black-box algorithms that reach their "optimal" solutions without offering insight into their decision-making processes risk undermining trust, exacerbate biases, and limit accountability. Scholarly work by Floridi (2019) and the IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems further emphasizes that traceability in AI systems is foundational for robust ethical oversight (Chatila *et al*., 2017).

1.6.4 Integrating Ethical Reflections: Beyond Religious and Scientific Dichotomies

Moving beyond a dichotomous view—whether the universe is a product of divine creation or of self-organizing physical laws—highlights a shared imperative: to harness AI in a manner conducive to collective flourishing. In Tegmark's framework, the aspirational goal is not merely to build powerful AI but to build AI systems that enhance human (and potentially non-human) well-being (Tegmark, 2017). Under a theistic interpretation, this may be seen as honoring a sacred creation by ensuring that technological progress does not harm or disrupt divine harmony. From a purely scientific standpoint, the same conclusion can be reached by recognizing that stable, sustainable states—those that do not precipitate conflict or environmental depletion—are also more "optimal" for societal survival over the long term.

Furthermore, some thinkers question whether nature is indeed optimally efficient or if it follows a path that only *appears* efficient over specific observational scales. For instance, complexity theorists and those studying evolutionary biology sometimes highlight cases where biological processes involve apparent redundancy or "inefficiency" that later proves advantageous. This cautionary note suggests that while emulating nature's optimization

strategies can be insightful, humans must also remain aware of potential blind spots and emergent complexities. AI systems might similarly display unexpected behaviors or unanticipated interactions with society.

From Tegmark's call for a responsible stewardship of AI to broader academic discussions of algorithmic fairness, transparency, and accountability, the ethical concerns surrounding AI usage are multifaceted. Whether one interprets the universe through the lens of a perfect divine creation or through the lens of evolutionary adaptation, there is consensus that nature's inclination toward optimization can illuminate valuable lessons for AI design. Machine learning, with its reliance on gradient descent and iterative improvement, already exemplifies the principle that "nature is the optimum"—but this optimization must be guided by ethical considerations that transcend narrow cost functions and short-term objectives.

The key challenge, therefore, is to ensure that AI's pursuit of efficiency or "lowest-energy" solutions remains aligned with human dignity, equity, and well-being. Future research should investigate how to embed robust ethical frameworks—rooted in either religious values, secular humanist principles, or hybrid approaches—into AI algorithms and governance structures. Furthermore, ongoing dialogue between ethicists, engineers, policymakers, and the broader public is essential for crafting an AI ecosystem that is as harmonious as it is innovative. By learning from nature's evident search for optimum states while acknowledging both its achievements and pitfalls, we set the stage for ethical AI that can genuinely benefit humanity and, ideally, the global biosphere as well.

1.6.5 Environmental Impacts of AI

The rapid proliferation of artificial intelligence has yielded transformative benefits across diverse sectors, ranging from health care to finance and beyond. However, the surge in AI adoption also underscores pressing environmental concerns—particularly with respect to the energy demands associated with developing and deploying cutting-edge systems. These concerns pivot on the computational intensity of training and running large AI models, a challenge that is exacerbated by an ever-increasing demand for more complex models and near-instantaneous inference. This Section provides an overview of these environmental impacts, drawing on the work of leading scholars who have examined the carbon footprint of AI and exploring contrasting perspectives on how best to mitigate ecological repercussions.

1.6.6 High Energy Consumption of Large-Scale AI

A recurring theme in the literature is the significant energy usage required to train large-scale AI models. Pioneering studies by Strubell, Ganesh and McCallum (2019) have drawn attention to the carbon footprint of training deep neural networks, especially those in natural language processing (NLP). These authors revealed that training a single high-end model can emit as much CO_2 as multiple cars throughout their entire lifespans. Subsequent work by Patterson *et al*. (2021) corroborates this notion, demonstrating that the computational power needed to achieve incremental improvements in model accuracy grows substantially with each new generation of architectures.

While training garners the bulk of the attention, the inference phase—whereby trained models are continuously queried—also incurs an energy cost. As AI becomes integrated into consumer devices, smartphones, and large-scale server architectures, the aggregated energy draw becomes substantial. Optimized hardware accelerators have partially alleviated this issue, yet the sheer number of real-time inferences can still lead to significant electricity consumption.

1.6.7 Carbon Footprint and Climate Impact

Data centers—housing the massive compute infrastructures essential for AI research—are responsible for an appreciable portion of global energy usage. According to the International Energy Agency (IEA, 2023), data centers accounted for about 1% of global electricity demand, a figure projected to rise with increasing AI workloads. One common viewpoint, supported by Crawford (2022), is that AI technology's carbon footprint reflects broader systemic inefficiencies: the expansion of digital services and Big Tech infrastructures strains power grids reliant on fossil fuels, thereby contributing to greenhouse gas emissions.

Some researchers argue for a more holistic analysis of AI's environmental footprint—encompassing the entire life cycle of hardware, from raw material extraction to manufacturing, transportation, operation, and disposal. This perspective, championed by Gebru and others, highlights the ethical trade-offs: while AI can yield social and economic value, it may inadvertently perpetuate environmental harm if such externalities remain unaccounted for. Efforts are underway to standardize these life-cycle assessment methodologies, with the goal of precisely quantifying the full extent of AI's ecological cost.

1.6.8 Contrasting Viewpoints: Threat or Opportunity?

Critics of current AI research and deployment practices focus on the apparent mismatch between the pace of model growth and existing climate targets. Schwartz *et al*. (2020) introduced the notion of "Green AI," critiquing the trend toward training increasingly large models without commensurate gains in efficiency. These detractors maintain that the relentless pursuit of state-of-the-art performance leads researchers to prioritize model size over sustainability. The negative environmental externalities—particularly from institutions with access to "hyperscale" computing resources—could, in this view, undermine international climate goals.

A more optimistic stance recognizes that AI itself may help mitigate the very environmental challenges it accelerates. An Editorial published by Nature Machine Intelligence (2022) has illustrated ways in which machine learning could be harnessed to optimize energy grids, reduce waste in supply chains, and model complex climatic phenomena with unprecedented accuracy. By improving the efficiency of renewable energy systems or guiding better agricultural practices, AI has the potential to yield net-zero or even net-positive environmental outcomes in the long run. This camp argues that while AI's current footprint is non-trivial, its net impact could become positive if responsibly managed and directed toward solving ecological problems.

1.6.9 Mitigating Strategies and Ongoing Debates

One prominent approach to mitigating AI's environmental effects involves hardware-based optimizations. The emergence of specialized AI chips—such as Google's Tensor Processing Units (TPUs) and various purpose-built accelerator architectures—demonstrates a concerted effort to reduce power consumption relative to traditional GPUs. Additionally, dynamic voltage and frequency scaling (DVFS), improved cooling solutions, and advanced manufacturing processes all contribute to more energy-efficient AI systems. Advocates of hardware-focused solutions argue that technical innovations may significantly offset the energy demands of large-scale AI, thereby alleviating the strain on power grids.

Beyond hardware enhancements, Bender *et al*. (2021) have underscored the importance of algorithmic efficiency. Techniques such as model pruning, weight quantization, and knowledge distillation aim to reduce model size and computational overhead without significantly compromising performance. If widely adopted, these strategies could curb AI's

carbon footprint while still maintaining accuracy. Yet, tensions remain between those pushing for efficiency-first AI research and others who remain focused on pushing the boundaries of model performance and capabilities.

Policy frameworks and corporate strategies also shape how AI can evolve sustainably. Governments and regulatory bodies are examining carbon taxes, energy-efficiency standards, and investment in renewable energy sources to incentivize responsible AI growth. Likewise, an increasing number of technology companies are pledging to offset or eliminate carbon emissions from their data centers. However, critics point to the possibility of "greenwashing," where companies tout modest environmental initiatives as major leaps forward, without fundamentally altering carbon-intensive practices.

1.6.10 Toward a Sustainable AI Ecosystem

Achieving a balance between AI innovation and environmental stewardship will likely require a multi-pronged strategy. On one hand, AI is poised to advance scientific discovery, reduce waste, and coordinate more efficient energy usage—potentially aiding humanity's transition to a low-carbon economy. On the other hand, the proliferation of large-scale AI models demands systemic efforts to avoid exacerbating climate change. By merging policy interventions, hardware innovations, and algorithmic efficiencies, the AI community can seek to minimize the environmental costs of rapid technological progress.

In conclusion, the environmental impacts of AI are neither trivial nor insurmountable. Meaningful changes depend on transparent reporting of AI's carbon footprint, adoption of more efficient computational strategies, and aligning AI's rapid growth with global sustainability goals. Only by weaving together these diverse efforts can we hope to construct an AI ecosystem that maximizes societal benefits while preserving the planet's ecological balance.

1.6.11 Social Impacts of AI

AI has become inextricably linked to contemporary debates on the future of work, wealth distribution, and societal well-being. While many commentators celebrate AI's potential to boost productivity and innovation, others raise concerns about rising unemployment and social displacement. Scholars such as Erik Brynjolfsson and McAfee (2014) have explored how digital technologies might augment human labor and simultaneously displace certain jobs, whereas Ford (2015) warns of a future in which

automation, powered by AI, could create widespread economic upheaval. This Section offers a nuanced examination of these divergent viewpoints, culminating in the argument that, much like earlier technological innovations (e.g., spreadsheets, word processors), AI is fundamentally a tool. Its proper integration into the workforce, paired with widespread upskilling and education, can yield significant social benefits rather than an erosion of employment.

1.6.12 Technological Displacement vs. Technological Augmentation

One longstanding fear is that machines will become so adept at performing tasks previously reserved for humans that vast segments of the workforce will become obsolete. Frey and Osborne's (2013) seminal study on the future of employment fueled these anxieties by estimating that a substantial percentage of current occupations could be automated within the coming decades. Similarly, Ford (2015) posits that highly capable AI systems could automate not just blue-collar but also white-collar jobs in fields such as law, journalism, and finance.

Contrasting these concerns, a significant body of literature highlights AI's potential to act as a complement rather than a substitute to human labor. Brynjolfsson and McAfee (2014) suggest that advanced automation can free employees from routine tasks, thereby enabling them to focus on higher-value activities like creative problem-solving, complex decision-making, and interpersonal communication. Acemoglu and Restrepo (2019) similarly emphasize that while automation can displace workers performing repetitive tasks, it also creates new roles that demand different skill sets. This phenomenon has been observed in prior technological revolutions—such as the introduction of personal computing in the 1980s—where job displacement coexisted with job creation in new sectors.

1.6.13 AI as a Productivity Tool

To contextualize current debates, it is instructive to compare AI to previous technological breakthroughs. For instance, the widespread adoption of the spreadsheet software Microsoft Excel in the 1990s and 2000s revolutionized accounting, finance, and administration. Despite initial fears that it would replace accountants and administrative staff, Excel ultimately served as a powerful tool that enhanced job efficiency and accuracy, rather than rendering humans superfluous. David Autor's research (Autor; Levy; Murnane, 2003) on the "task approach" to technological change further illustrates that new tools typically

transform the task composition of jobs rather than eliminate entire occupations outright.

LLMs and AI Agents, exemplified by GPT-series models and other advanced neural architectures, similarly offer productivity gains. They can assist in drafting emails, coding, or data analysis tasks, accelerating workflows and reducing human error. Critics worry that these platforms will lead to large-scale layoffs in knowledge-intensive industries. Yet, from a historical perspective, these AI tools are akin to next-generation productivity suites. Just as proficiency in spreadsheet software is often a baseline requirement for employment in managerial or administrative roles, familiarity with LLMs and AI Agents is likely to become a basic skill for knowledge workers. Their function, much like that of Microsoft Excel or Microsoft 365, is to enhance human capabilities rather than outright replace them.

1.6.14 The Essential Role of Upskilling

While the fears of "job theft" by AI may be overstated, there is a genuine risk that individuals lacking AI-related skills could find themselves at a disadvantage in an evolving labor market. In the same way that an engineer who cannot navigate a standard Office Suite application might face hurdles in a modern workplace, professionals unfamiliar with AI tools may appear unprepared for the data-driven, automated workflows of today. The World Economic Forum (2023) highlights the critical need for continuous learning programs—both formal and on-the-job—to ensure that the labor force remains adaptable and capable of leveraging these new technologies.

Universities and vocational institutions can play a key role in fostering AI literacy by integrating AI-related courses into standard curricula. Simultaneously, corporations have an incentive to offer ongoing professional development opportunities to keep their workforce competitive and innovative. Initiatives such as "AI literacy programs" can demystify algorithms and models, illustrating that they are extensions of human intelligence rather than existential threats. As Brynjolfsson and McAfee (2014) stresses, technological transitions are most successful when they incorporate "people-first" strategies—where employees are not only trained to use new tools but encouraged to explore their creative potentials in tandem with computational assistance.

1.6.15 Balancing Risk and Reward

Beyond productivity, AI harbors the potential for broader social impacts. When responsibly deployed, it can streamline public services, democratize access to education, and

improve healthcare via predictive diagnostics. Indeed, Amy Webb (2019) argues that AI can yield a net positive outcome if guided by ethical frameworks and thoughtful governance.

On the other hand, a failure to address AI-driven inequalities could polarize societies. Without equitable access to AI education or fair labor policies, certain demographics may become marginalized. Additionally, concerns about algorithmic bias and data privacy could erode public trust. These issues highlight the responsibility borne by stakeholders—governments, corporations, academic institutions—to mitigate risks and ensure that AI remains a force for good.

The societal impact of AI is multifaceted, neither wholly utopian nor purely dystopian. Fears that AI will "steal our jobs" mirror historical anxieties about new technologies that often dissipate as societies adapt and discover new avenues of value creation. LLMs, AI Agents, and other advanced systems are best understood as potent productivity tools. Their successful adoption depends on widespread digital literacy, continuous upskilling, and proactive policy measures that ensure fair access and responsible use.

Ultimately, as with every technological invention since humanity learned to harness fire, AI can be wielded for immense benefit or considerable harm. Our collective challenge lies in shaping AI ecosystems that maximize social good—nurturing human ingenuity rather than replacing it—and in ensuring that these new tools remain servants to our broader societal goals, not the other way around.

1.6.16 Biases in AI

AI systems, particularly LLMs, have demonstrated remarkable capabilities across various domains. However, their deployment in critical sectors, such as aerospace defense, necessitates a thorough examination of inherent biases that may compromise decision-making processes. Biases in AI can emerge from multiple sources, including training data, model architecture, and deployment contexts. Understanding these biases is essential to mitigate their impact and ensure the reliability and fairness of AI applications.

1.6.17 Types of Biases in AI

AI systems are susceptible to several forms of bias (Bevara *et al.*, 2023), each influencing outputs in distinct ways:

Gender Bias: LLMs often perpetuate gender stereotypes present in their training data, leading to associations that reinforce traditional gender roles. For instance, models may link professions like nursing predominantly with women and engineering with men, reflecting societal biases embedded in the data (Ajith; Rithani; Syamdev, 2023).

Racial and Ethnic Bias: Training data that underrepresent certain racial or ethnic groups can result in models that produce outputs favoring majority groups. This underrepresentation can lead to disparities in AI-generated content, affecting fairness and inclusivity (Fang *et al*., 2023).

Ageism and Beauty Bias: Subtler biases, such as those related to age and physical appearance, can influence AI outputs. Studies have shown that LLMs may exhibit preferences or prejudices based on age and perceived attractiveness, which can affect the quality and fairness of generated content (Kamruzzaman; Shovon; Kim, 2023).

Political Bias: LLMs trained on data with prevalent political viewpoints may generate content that leans towards specific ideologies. This bias can influence the neutrality of AI-generated information, particularly in sensitive contexts (Feng *et al*., 2023).

Language and Cultural Bias: Models predominantly trained on English-language data may present Anglo-American perspectives as universal truths, neglecting or misrepresenting non-English viewpoints. This bias can lead to a lack of cultural sensitivity and inclusivity in AI outputs (Luo; Puett; Smith, 2023).

Stereotype Bias: In addition to gender and race, these models have the potential to perpetuate a multitude of stereotypes, encompassing those rooted in nationality, religion, or profession. Consequently, this can result in outputs that unjustly categorize or lampoon groups of individuals, occasionally in detrimental or disparaging manners (Cheng; Durmus; Jurafsky, 2023).

Emergent Bias: It arises from the deployment and dependence upon algorithms within novel or unforeseen circumstances. These algorithms may not have been calibrated to accommodate emergent knowledge, including novel pharmaceuticals or medical advancements, revised legislation, innovative commercial paradigms, or evolving societal mores. Consequently, this can lead to the marginalization of specific cohorts via technology without offering clear accountability frameworks to determine the agents responsible for their exclusion. Likewise, challenges may arise when the training data (the exemplars provided to a machine for the purpose of modeling specific outcomes) are incongruent with the real-world

contexts encountered by an algorithm (Mehrabi *et al*., 2022).

Technical bias: It manifests through inherent limitations within a program, encompassing computational capacity, design parameters, or other systemic constraints. Such bias can also arise from design choices; for instance, a search engine displaying a limited number of results per screen inherently prioritizes those initial results over subsequent ones, akin to an airline fare display. Furthermore, software relying on stochastic processes for equitable distribution of outcomes can introduce bias if the random number generation mechanism deviates from true randomness, potentially skewing selections towards items at the extremities of a list (Greene, 2022).

Data Bias: It arises when the training datasets are unrepresentative or contain inherent prejudices. This can lead to models that perpetuate existing societal biases or fail to generalize across diverse scenarios. For instance, if an LLM is trained predominantly on English-language texts from Western sources, it may not perform well with inputs from other cultures or languages, leading to skewed outputs. Such biases can manifest as selection bias, where certain groups are underrepresented, or reporting bias, where the data reflects specific viewpoints more prominently (Mavrogiorgos *et al*., 2024).

Algorithmic bias: This bias occurs due to the design and functioning of the AI algorithms themselves. Certain algorithms may inadvertently favor specific outcomes based on their structure or optimization criteria. For example, optimization functions that prioritize accuracy over fairness can result in models that perform well on average but poorly for minority groups. Additionally, the choice of regularization methods and hyperparameters can influence the model's behavior, potentially introducing unintended biases (Mavrogiorgos *et al*., 2024).

1.6.18 Factors for LLMs' Biases

The biases observed in LLMs can be attributed to some factors:

Training Data: The quality and diversity of the data used to train AI models are critical. Datasets that lack representation of certain groups or perspectives can embed existing societal biases into the model (Ferrara, 2023).

Model Architecture: The design of AI models can influence how biases are learned and manifested. Certain architectures may be more prone to amplifying biases present in the training data (Zhang *et al*., 2024).

Deployment Context: The environment in which AI systems are deployed can introduce or exacerbate biases. For example, user interactions and feedback loops can reinforce certain biases over time (Morales; Clarisó; Cabot, 2023).

Biases in AI systems can have critical consequences in the aerospace defense sector. For instance, biased threat assessment models may overlook certain risks or disproportionately flag benign activities, leading to operational inefficiencies or security vulnerabilities. Therefore, it is imperative to implement rigorous bias detection and mitigation strategies, ensuring that AI applications in this domain are fair and reliable.

Addressing these biases requires a multifaceted approach, including the use of diverse and representative datasets, transparent algorithmic design, and continuous monitoring and evaluation of AI systems. By acknowledging and proactively managing biases, the aerospace defense sector can harness AI's full potential while upholding ethical standards and operational integrity.

1.6.19 Mitigations for Biases on this Research

Due to its nature, this research is not susceptible to the impacts of gender, racial, ethnic, age, beauty, stereotype, or political biases. However, it may still be influenced by linguistic, emergent, technical, data, and algorithmic biases.

Regarding linguistic bias, this work anticipated the performance discrepancies of LLMs across different languages. The dataset employed for fine-tuning the 'gpt-3.5-turbo' model initially comprised requirements written in English and Portuguese. To mitigate this issue, prior to the fine-tuning process, all requirements written in Portuguese were translated using a "Pro" account of Grammarly, a highly reputable textual translation application.

Concerning data bias, efforts were made to utilize a dataset that was as balanced as possible with respect to the number of requirements per product type, as well as the distribution of MoCs required to fulfill the thousands of requirements comprising the training, validation, and test data.

Nevertheless, with regard to emergent, technical, and algorithmic biases, little can be done. LLMs are closed models, and their pre-training process and detailed functioning remain undisclosed. This directly relates to the issue of transparency. The mitigation of these points falls upon human operators, who remain essential for operation, decision-making, and especially supervision and review of the work conducted with the assistance of AI tools,

highlighting the importance of discussions surrounding accountability. Transparency and accountability are the subjects of the two subsequent Sections: 1.6.20 and 1.6.21.

1.6.20 Transparency in AI

Transparency in artificial intelligence (AI) is a critical factor in ensuring the ethical deployment and trustworthiness of AI systems, particularly in safety-critical sectors such as aerospace defense. The complexity and opacity of LLMs pose significant challenges to transparency, as these models often function as "black boxes," making it difficult to understand their decision-making processes.

The lack of transparency in AI systems can lead to several issues:

Unintended Biases: Without clear insight into how AI models process data, it becomes challenging to identify and mitigate biases that may be present in the training data or the model's architecture (Cambria *et al*., 2024).

Accountability: Opacity in AI decision-making complicates the assignment of responsibility when errors or unintended consequences occur, raising concerns about accountability in AI applications (Liao; Vaughan, 2023).

Trust and Adoption: Stakeholders may be reluctant to adopt AI technologies if they cannot comprehend or trust the underlying mechanisms, especially in domains where safety and reliability are fundamental (Liao; Vaughan, 2023).

To address these challenges, some strategies have been proposed:

Explainable AI (XAI): Developing methods that provide interpretable and understandable explanations of AI decisions can enhance transparency. Techniques such as feature attribution and model distillation aim to make AI systems more comprehensible to human users (Cambria *et al*., 2024).

Transparent Reporting: Implementing standardized documentation practices, such as AI FactSheets, can foster greater transparency by offering detailed insights into the development, capabilities, and limitations of AI models (Arnold *et al*., 2018).

Open-Source Initiatives: Releasing AI models and their training data to the public can promote transparency and allow for community-driven scrutiny and improvement. However, this approach must balance transparency with considerations of security and misuse (Heaven, 2023).

Enhancing transparency is essential in the context of aerospace defense, where AI applications must meet stringent safety and reliability standards. Implementing XAI techniques, adopting transparent reporting practices, and engaging in open-source collaborations can contribute to the development of AI systems that are not only effective but also more trustworthy and accountable. However, it seems impossible to completely eliminate all possible types of bias and transparency issues.

1.6.21 Accountability in AI

Accountability in Artificial Intelligence (AI) is a multifaceted concept that encompasses the ethical, legal, and professional responsibilities of individuals and organizations involved in the development, deployment, and utilization of AI systems. As AI technologies, particularly Large Language Models (LLMs), become increasingly integrated into various sectors, ensuring accountability is paramount to mitigate risks associated with biases and lack of transparency.

A critical aspect of AI accountability pertains to the professional responsibility of individuals who employ AI tools in their work. Drawing parallels from other domains, it is evident that professionals are expected to possess the requisite skills and knowledge to effectively utilize the tools pertinent to their roles. For instance, an engineer who fails to competently use software like Microsoft Excel, leading to a design flaw, is held accountable for the oversight. Similarly, a pilot who commits a navigational error due to inadequate proficiency with an Electronic Flight Bag (EFB) bears primary responsibility for the mistake. In both scenarios, while employers share some responsibility for providing appropriate training and resources, the ultimate accountability rests with the professionals who accepted responsibilities beyond their competencies.

This principle extends to the use of AI systems. Professionals integrating AI into their workflows must ensure they understand the capabilities and limitations of these technologies. Failure to do so can lead to erroneous outcomes, for which the individual is primarily accountable. Hohma (2023) underscores the importance of embedding AI ethics requirements at each step of the AI development lifecycle, emphasizing that accountability involves taking responsibility and providing justification for one's actions.

Moreover, the Organisation for Economic Co-operation and Development (OECD) highlights that organizations and individuals developing, deploying, or operating AI systems should be held accountable for their proper functioning in line with established principles

(OECD, 2024).

This underscores the shared responsibility between employers and employees. While organizations must facilitate training and establish clear guidelines, professionals are obligated to be honest about their competencies and seek necessary qualifications to fulfill their roles effectively.

1.6.22 Security Using LLMs: a Strategic Approach for Sensitive Data Management

The integration of LLMs into organizational workflows has revolutionized data processing, decision-making, and communication. However, for institutions handling classified or sensitive information, such as the Brazilian Air Force and its associated research institutes, sharing their documentation is not an option. They face significant challenges in adopting mainstream AI services like ChatGPT (OpenAI), Gemini (Google), and Claude (Anthropic) due to stringent data confidentiality requirements. This necessitates exploring alternative approaches that ensure data sovereignty while leveraging the benefits of LLMs.

To mitigate risks associated with external data transmission, deploying open-source LLMs on local intranets is a viable solution. Models like Llama (Meta), an open-source LLM, support local deployment within institutions, ensuring that sensitive data remains within the organization's secure environment. This approach aligns with privacy-preserving practices, as it avoids the transmission of sensitive information to third-party servers.

1.6.23 Advancements in Open-Source LLMs

The open-source community has seen significant advancements with models like Meta's Llama 3. Released under an open-source license, Llama 3 offers capabilities that rival proprietary models, providing organizations with the flexibility to customize and fine-tune the model without external dependencies. This democratization of AI technology enables entities handling sensitive information to harness advanced LLM capabilities while maintaining strict data control.

The DeepSeek models have recently demonstrated significant advancements in the field of artificial intelligence, as evidenced by benchmarking studies that compare their performance to leading models from OpenAI, Meta, Google, and Anthropic. Notably, the DeepSeek-R1 model has shown superior performance in benchmarks such as AIME, MATH-500, and SWE-Bench Verified (Deepseek-AI *et al.*, 2025), surpassing OpenAI's o1 in tasks

involving deterministic reasoning and mathematical problem-solving. These results underscore DeepSeek's innovative approach, leveraging techniques like Mixture of Experts - MoE (a machine learning technique that dynamically routes inputs through a subset of specialized neural network "experts," optimizing computational efficiency and model performance by activating only the most relevant components for each task) and reinforcement learning to optimize computational efficiency. Unlike many proprietary counterparts, DeepSeek-R1 is open-source, licensed under MIT (Massachusetts Institute of Technology), enabling broader adoption in both commercial and academic contexts. This accessibility, combined with competitive performance, highlights the potential of DeepSeek models to challenge the dominance of major AI players, even as the company operates with constrained access to high-end hardware resources—a testament to their emphasis on algorithmic optimization over sheer computational power.

1.6.24 User-Friendly Interfaces for LLM Interaction

Interacting with LLMs through intuitive interfaces enhances usability and accessibility. Open WebUI (former Ollama WebUI) (Baek, 2025), for instance, offers a customizable platform that emulates ChatGPT-like interactions. It supports various models, including those compatible with Ollama, and provides features such as Retrieval-Augmented Generation (RAG) and fine-tuning capabilities. RAG is a hybrid approach in natural language processing that combines the generative capabilities of large LLMs with the retrieval of relevant external knowledge to enhance response accuracy and relevance. In this framework, a retriever first identifies the most pertinent documents or data points from a predefined knowledge base based on the user’s query. These retrieved pieces of information are then integrated into the input of the generative model, which produces a final output that incorporates both the contextual understanding of the query and the retrieved factual knowledge. By dynamically grounding the generation process in up-to-date or domain-specific information, RAG overcomes the limitations of static model training, improving performance in tasks such as question answering, summarization, and decision support in specialized domains. This flexibility allows users to tailor the LLM's responses to specific tasks and domains, thereby improving efficiency and relevance.

1.6.25 Optimizing Performance: Model Selection and Fine-Tuning

Selecting appropriate models is crucial for balancing performance with available

hardware resources. For instance, models like Phi4 can operate on GPUs with as little as 12GB of memory, making them suitable for organizations with limited computational infrastructure. While these models may not match the comprehensive capabilities of larger models like GPT-4o, they can be fine-tuned to specialize in specific tasks relevant to the organization's domain.

1.6.26 Fine-Tuning LLMs for Specialized Tasks

Fine-tuning LLMs enables the adaptation of general-purpose models to specialized tasks, enhancing their performance in specific domains. Tools like WebUI facilitate this process by providing a low-code interface for fine-tuning models on custom datasets. This capability is particularly beneficial for organizations requiring LLMs to comprehend and process domain-specific terminology and contexts.

1.6.27 Implementing Retrieval-Augmented Generation (RAG)

RAG enhances the LLM's ability to generate contextually relevant responses by integrating external knowledge sources. WebUI's support for RAG allows users to augment the LLM's responses with information retrieved from specified knowledge bases or databases. This feature is crucial for tasks that demand up-to-date or specialized information, ensuring that the LLM's outputs are both accurate and pertinent.

1.6.28 Considerations for Local LLM Deployment

While local deployment of LLMs offers significant advantages in data security, several considerations must be addressed:

- **Hardware Requirements**: LLMs, especially larger models, demand substantial computational resources. Organizations must assess their hardware capabilities to ensure efficient model operation.
- **Data Compliance**: Implementing robust access controls and authentication mechanisms is essential to prevent unauthorized access and ensure compliance with data protection regulations.
- **Model Maintenance**: Regular updates and maintenance are necessary to keep the models current and effective, which may require dedicated personnel and resources.

1.7 Use of Generative AI

I acknowledge the use of AI software to rewrite, rephrase and/or paraphrase parts of this thesis to ensure the quality and standard of the English used. This thesis is a genuine account of the research I have undertaken, and the content can still be considered my own words, with all references cited accordingly.

1.8 Organization

This thesis is organized into seven chapters, each building upon the prior to address the research objectives and substantiate the proposed hypothesis.

Chapter 1 introduces the research topic, providing a comprehensive background, contextualizing the study within the current academic and industrial landscape, and establishing its significance. The research problem is clearly articulated, followed by the hypothesis and specific research objectives designed to confirm the hypothesis. It also explores the broader implications of AI applications in our society.

Chapter 2 presents the theoretical framework, offering a concise overview of the key concepts relevant to this research. This chapter serves as a foundation, covering essential topics such as System Theoretic Process Analysis (STPA), Large Language Models (LLMs), Prompt Engineering, and Automated Compliance in Aerospace Defense Systems, thus equipping the reader with the necessary theoretical background.

Chapter 3 comprises a literature review, critically analyzing recent works in the field and situating this research within the broader landscape of academic advancements in automated aerospace compliance and safety analysis. This chapter highlights both the contributions and gaps in existing literature, underscoring the originality and necessity of this study.

Chapter 4 details the methodology employed to achieve the research objectives outlined in Chapter 1. It describes the specific techniques, including the use of established Prompt Engineering techniques for STPA and the fine-tuning process for automated Means of Compliance (MoC) assignment, explaining their alignment with the study's goals.

Chapter 5 presents the research results. This chapter critically examines the limitations encountered, addresses the potential impacts of AI-based automation, and validates the methodology approached in Chapter 4.

Finally, Chapter 6 concludes the thesis by summarizing the research's contributions and significance within the field. This chapter reflects on the extent to which the research objectives have been met and offers insights for potential future work in the application of AI-driven analysis and compliance in aerospace defense systems.

Through this structured organization, the thesis aims to provide a comprehensive and cohesive narrative that guides the reader from foundational theories and literature to practical methodologies, results, and reflections on the study's broader impact.

2 Theoretical Framework

This chapter establishes the theoretical foundation essential for understanding the methodologies, analyses, and discussions in subsequent chapters. It provides a structured exploration of the key concepts, models, and techniques upon which this research is built, ensuring that readers unfamiliar with these areas gain a comprehensive understanding of each topic.

The chapter is organized into seven main Sections:

1. Systems Theory: This Section introduces the foundational concepts of Systems Theory, exploring the principles of interconnectedness, complexity, and feedback loops that characterize modern systems. These concepts are integral to comprehending the interdependencies in aerospace systems and lay the groundwork for System Theoretic Process Analysis (STPA), which will be discussed later.
2. Systems Engineering: Building upon Systems Theory, this Section delves into Systems Engineering, focusing on the processes and methodologies used to design, analyze, and manage complex systems throughout their lifecycle. Understanding Systems Engineering is crucial for recognizing the challenges and requirements that arise in aerospace defense systems, especially as they pertain to safety and compliance.
3. Means of Compliance (MoCs): This Section discusses MoCs, a cornerstone of aerospace certification and regulatory frameworks. It examines how MoCs are defined, assigned, and validated, particularly in the context of aerospace systems, and underscores the significance of accurate MoC assignment in ensuring regulatory compliance and safety.
4. System Theoretic Process Analysis (STPA): This Section provides an in-depth look at STPA, a methodology for identifying hazards and assessing safety within complex systems. It outlines the principles of STPA, explains its relevance to the aerospace defense sector, and discusses its role in eliciting requirements for safe system operation, setting the stage for its application in this research.
5. Lifecycle of Aerospace Systems in the Brazilian Air Force: Here, the unique aspects of the aerospace systems lifecycle within the Brazilian Air Force (FAB)

are examined. This Section highlights the stages of development, operation, and certification specific to FAB systems, providing critical context for understanding the application of STPA and compliance requirements in Brazilian defense projects.

6. Artificial Intelligence (AI): This Section introduces key concepts in AI, focusing on its role in automating processes, optimizing decision-making, and enhancing system safety. It provides an overview of AI techniques pertinent to this research, particularly in areas where AI aids complex decision-making in aerospace applications.
7. Large Language Models (LLMs): Finally, this Section explores the capabilities and limitations of Large Language Models (LLMs), with a particular emphasis on their potential to automate tasks related to STPA and compliance in aerospace systems. This Section offers insights into how LLMs, through Prompt Engineering and fine-tuning, can assist in requirements elicitation and MoC assignment, which are central to this thesis.

By establishing a detailed theoretical framework, this chapter equips readers with the necessary background to fully engage with the research questions, methodology, and findings presented in later chapters. Each Section progressively builds upon the previous, ensuring that even those with limited prior knowledge can follow the complex interplay of theories and techniques that underpin this study.

2.1 Systems Theory

Systems Theory is an interdisciplinary field developed through contributions from diverse scholars, with Ludwig von Bertalanffy frequently acknowledged as a foundational figure. His seminal work, **General System Theory** (Bertalanffy, 1968), established the principles of Systems Theory, promoting a holistic approach to understanding how the parts within complex systems interact and form an integrated whole. Rather than analyzing individual components in isolation, Bertalanffy emphasized that systems should be studied as interdependent structures, where relationships between elements are crucial to grasping system behavior as a whole. This holistic approach has since become a cornerstone in understanding various types of systems – from biological to social and engineered systems – and remains essential for assessing complex, adaptive systems in contemporary research

(Bertalanffy, 1968).

One of the fundamental concepts in Systems Theory is **emergent properties**. Emergent properties refer to characteristics or behaviors that arise from the dynamic interactions among system components. These properties are not attributes of the individual parts but emerge from the system as an interconnected entity (Georgiou, 2003). For example, in an organizational system, elements such as communication, collaboration, and workflow dynamics contribute to organizational culture – an emergent property that cannot be fully understood by isolating any single aspect of the organization. Systems Theory posits that emergent properties often exhibit behaviors that are unpredictable or nonlinear, thereby challenging traditional reductionist approaches (Georgiou, 2003).

Interconnectedness is another fundamental concept, emphasizing that all elements within a system are linked and that changes in one component often propagate through others. This interconnectedness is essential to understanding feedback loops, both positive and negative, that stabilize or amplify certain behaviors within systems. Positive feedback loops tend to reinforce change, while negative feedback loops serve to balance the system, bringing it back toward equilibrium. Feedback mechanisms are crucial for explaining how complex systems maintain stability or evolve in response to internal or external pressures (Forrester, 1976). In aerospace systems, for example, feedback mechanisms are critical in maintaining operational stability and safety, as they allow the system to adapt to variables that might otherwise compromise functionality.

Moreover, **open and closed systems** are key concepts within Systems Theory, as defined by Katz and Kahn (1978). An open system exchanges energy, information, or resources with its environment, making it susceptible to external influences. In contrast, a closed system is self-contained, with minimal interaction with its surroundings. Most real-world systems, including aerospace systems, operate as open systems, continuously adapting to changes in their operational environment, making them particularly relevant for defense applications where environmental unpredictability is a significant factor.

Complexity is intrinsic to Systems Theory and is particularly evident in systems with numerous interacting elements. Complexity often results in non-linear behavior, where small changes in one part of the system can have disproportionately large effects. This sensitivity to initial conditions is a defining feature of complex systems and is particularly pertinent to aerospace applications, where minor adjustments or failures in a subsystem can cascade through the entire system, potentially leading to safety-critical outcomes (Checkland, 1999).

The relevance of Systems Theory to Systems Engineering is profound. By emphasizing the interconnected and often unpredictable nature of components within a system, Systems Theory has shaped Systems Engineering's approach to managing complex projects, especially in fields requiring high safety and reliability, like aerospace engineering (Walden *et al*., 2015). Rather than isolating subsystems, engineers are encouraged to assess the system as an integrated whole, considering how changes in one area might influence others. This perspective is fundamental to System Theoretic Process Analysis (STPA), which is central to this research. By grounding STPA within a systems-theoretic framework, this thesis can address the complexities and interdependence characteristic of aerospace systems.

By establishing these foundational principles, this Section underscores Systems Theory's role in shaping modern approaches to safety, analysis, and engineering. These principles provide a framework for understanding the interconnected, complex, and often nonlinear nature of aerospace defense systems, setting the stage for a deeper exploration of Systems Engineering and the methodologies used in this research.

2.2 Systems Engineering

Systems Engineering (SE) is a structured, interdisciplinary approach focused on designing, integrating, and managing complex systems across their entire life cycle, addressing technical, economic, and social dimensions. SE is particularly vital in fields like aerospace and defense, where reliability and safety are non-negotiable. This Section delves into SE principles by integrating key perspectives from foundational texts, including the INCOSE Systems Engineering Handbook (Walden *et al*., 2015), ISO/IEC/IEEE 15288:2023, and additional influential SE literature, highlighting the evolution, principles, and applications of SE.

The INCOSE Systems Engineering Handbook defines SE as an approach that combines technical and management processes to deliver products that meet user needs, emphasizing performance, cost, and schedule constraints. Key SE processes include requirement analysis, system architecture, verification and validation, and system integration. The Handbook articulates SE's role in ensuring that complex systems operate as intended, focusing on managing the interdependencies among components. This is especially important in high-risk fields like aerospace, where precise coordination across all system elements is critical (Walden *et al*., 2015).

In parallel, ISO/IEC/IEEE 15288:2023 establishes a structured framework for SE across four primary groups: Technical Processes, Project Management Processes, Agreement Processes, and Organizational Project-Enabling Processes. This standard is widely used in aerospace, defense, and other industries where systems must undergo rigorous development and testing. It emphasizes adaptability, promoting the tailoring of SE processes to meet specific project demands. ISO 15288's framework enables structured systems lifecycle management from concept through decommissioning, ensuring that systems meet both functional and compliance requirements throughout their operational life (ISO, 2023).

2.2.1 Object-Process Methodology

Object-Process Methodology (OPM) is a comprehensive systems modeling paradigm that integrates the structural and behavioral aspects of systems into a unified framework. Developed by Dov Dori, OPM facilitates the representation of complex systems through a combination of graphical and textual elements, enhancing both human comprehension and machine interpretability (Dori, 2002).

At the core of OPM is the integration of objects and processes, where objects represent the structural components of a system, and processes denote the dynamic behaviors that transform these objects. This duality allows for a holistic depiction of systems, capturing both static and dynamic facets within a single model. The graphical component of OPM is the Object-Process Diagram (OPD), which visually illustrates the relationships and interactions between objects and processes. Complementing the OPD is the Object-Process Language (OPL), a subset of natural English that provides a textual narrative of the model, ensuring clarity and reducing ambiguity (Dori, 2002).

OPM's unique approach offers several advantages over traditional modeling languages. Unlike the Unified Modeling Language (UML), which employs multiple diagram types to represent different system aspects, OPM utilizes a singular diagrammatic form, streamlining the modeling process and mitigating the challenges associated with maintaining consistency across various diagram types. This singularity simplifies the modeling process and enhances the coherence of the system representation. Furthermore, OPM's bimodal representation—combining visual and textual descriptions—caters to diverse stakeholders, facilitating effective communication among system architects, designers, and domain experts.

The efficacy of OPM in conceptual modeling has been demonstrated across various domains. For instance, in the realm of MBSE, OPM has been employed to improve

conceptual modeling by incorporating stereotype features unique to its framework, thereby enhancing model construction, comprehension, and error reduction during design and simulation phases (Kohen; Dori, 2021).

The standardization of OPM as ISO 19450 has further solidified its position in the systems engineering community, providing a formalized framework for its application. This standardization facilitates the integration of OPM with other modeling languages, such as the Systems Modeling Language (SysML), enabling the creation of SysML views from OPM models and promoting interoperability among different modeling approaches (ISO, 2024).

2.2.2 Risk Management

Risk Management is another critical component of SE. In SE, Risk Management involves identifying potential project risks and implementing strategies to reduce them, thus ensuring greater reliability and minimizing the likelihood of system failure. The INCOSE Handbook underscores the importance of early-stage risk assessments, emphasizing the integration of mitigation plans from the conceptual design phase onward. This proactive approach is particularly impactful in aerospace and defense systems, where mission-critical failures carry high consequences (Walden *et al*., 2015).

2.2.3 System Thinking

Ultimately, SE practices are grounded in systems thinking, a holistic view that emphasizes the interrelationships among system components and their environments. Systems Thinking is an interdisciplinary framework that emphasizes understanding complex systems by focusing on the relationships, interdependencies, and interactions among system components rather than merely analyzing individual elements in isolation.

Originating from the General Systems Theory proposed by Ludwig von Bertalanffy (1968), Systems Thinking posits that a system's behavior cannot be fully understood by examining its parts independently. The whole exhibits emergent properties that arise from the dynamic interplay among its elements.

Senge (1997), in his seminal work *The Fifth Discipline*, highlights that Systems Thinking enables practitioners to identify feedback loops, delays, and non-linear relationships, which are crucial in predicting long-term outcomes and addressing root causes rather than symptoms. This approach is particularly valuable in engineering and management, where

complex, adaptive systems, such as socio-technical or organizational systems, require holistic strategies that account for interconnected processes and cascading effects (Sterman, 2011). By fostering a mindset that considers the broader context and the systemic nature of problems, Systems Thinking enhances the ability to create resilient and sustainable solutions, especially in fields like aerospace and defense, where system failures can have far-reaching consequences.

2.2.4 The Vee Diagram

The Vee Diagram is a well-established framework in Systems Engineering that visually represents the system development lifecycle, emphasizing the relationship between system decomposition and integration, as well as the progressive stages of verification and validation. Originally developed to structure the management of complex engineering projects (Forsberg; Mooz, 1992), the Vee Diagram, illustrated in Figure 2.1 is widely used across aerospace, defense, and other industries where robust, reliable systems are essential. Its "V" shape illustrates the left side as the decomposition phase, moving from high-level requirements to detailed design, while the right side represents the integration and validation phase, culminating in the verification of the system against requirements.

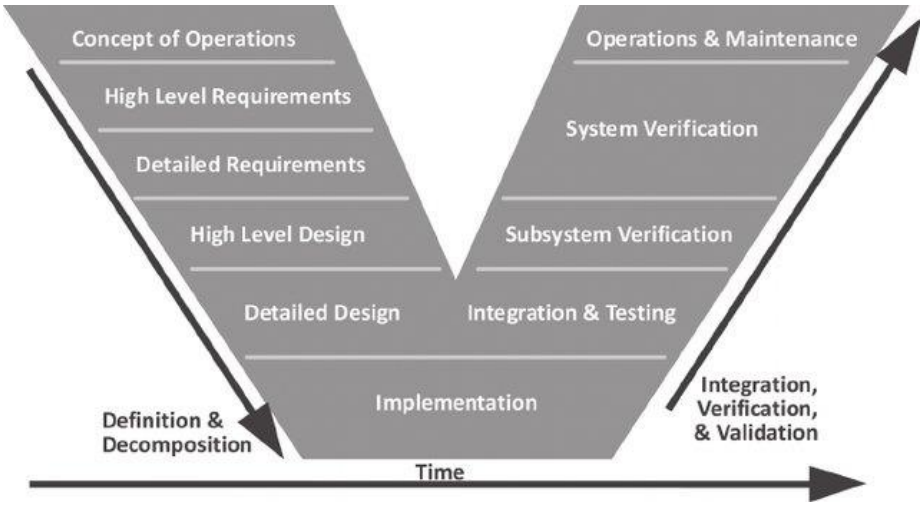


Figure 2.1 – Overview of V-model of systems engineering (Elm *et al*., 2008).

The left side of the Vee Diagram begins with Concept Definition, followed by System Requirements and Design Decomposition phases, guiding teams from abstract concepts toward detailed specifications for subsystems and components. At the bottom of the "V" is the Implementation phase, where system components are built or procured. The right side then involves System Integration and Verification and Validation (V&V) processes, where

components are progressively combined and tested to ensure the system functions as intended and meets initial requirements. This structure emphasizes the alignment between each stage on the left and the corresponding testing and integration on the right, thus reinforcing the importance of designing with validation in mind (Blanchard; Fabrycky, 2014).

The Vee Diagram aids Systems Engineering teams by providing a clear, structured approach to managing complexity over a system's lifecycle. By aligning each design phase with a corresponding testing phase, the Vee model encourages teams to consider early verification and validation, reducing the risk of costly rework in later stages (Estefan, 2008). This alignment also identifies potential issues early, facilitating proactive problem-solving before full integration. Furthermore, the Vee Diagram reinforces the iterative nature of Systems Engineering, allowing teams to revisit and refine requirements and designs as new insights emerge during development.

In summary, Systems Engineering, guided by frameworks like the INCOSE Handbook, ISO standards, and contributions from systems theory, MBSE, and HFE, is a robust approach for managing complexity and ensuring that systems meet rigorous performance, reliability, and safety requirements throughout their lifecycle. This theoretical foundation enhances system robustness and adaptability and addresses the unique demands of fields where failure is not an option, such as aerospace and defense.

2.3 Means of Compliance

From a European perspective, Means of Compliance, or MoCs, are the categorization of the means used to demonstrate compliance with the requirements (De Florio, 2016). A requirement can be met, for instance, by a flight test, an analysis, a simulation, or all of them. Table 2.1 depicts the classical MoCs and their significance for the European Union Aviation Safety Agency (EASA).

Table 2.1 – MoCs for aircraft requirements according to EASA (De Florio, 2016).

Type of Compliance	Means of Compliance	Associated Compliance Documents
Engineering evaluation	MoC0: - Compliance statement - Reference to Type Design documents - Election of methods, factors etc. - Definition	- Type Design documents - Recorded statements
	MoC1: Design review	- Description -Drawings
	MoC2: Calculation/Analysis	- Substantiation reports
	MoC3: Safety assessment	- Safety analysis
Tests	MoC4: Laboratory tests	- Test programmes - Test reports - Test interpretations
	MoC5: Ground tests	
	MoC6: Flight tests	
	MoC8: Simulation	
Inspection	MoC7: Design inspection/audit	- Inspection or audit reports
Equipment qualification	MoC9: Equipment qualification	- Note: Equipment qualification is a process which may include all previous means of compliance

According to the United States Federal Aviation Administration (FAA) (United States, 2017), Means of Compliance are detailed design standards that achieve the safety objectives embedded within the regulation. They serve as a mechanism utilized by an applicant to demonstrate adherence to airworthiness criteria and are acknowledged by the Administrator. A Mean of Compliance represents a singular approach, albeit not exclusive, for evidencing conformity with a regulatory mandate.

Establishing the MoCs involved in the certification of civil aeronautical products is relatively easy, albeit laborious (Gallina; Andrews, 2016). Extensive material guides aircraft manufacturers and their components throughout the certification process. The main aeronautical certification agencies worldwide offer this material for free download on their

websites. In fact, adopting MoCs different from those provided for in FAA Advisory Circulars, for example, increases the amount of detail necessary to avoid ambiguity in the certification's findings rationale (United States, 2023).

However, despite the many similarities between defense aerospace products and civilian aircraft, there is a stark difference in the availability of supporting materials regarding verification methods (Silva *et al*., 2018). This discrepancy significantly complicates the task of assigning MoCs to requirements for defense aerospace systems. Studies, analyses, meetings, and technical discussions are often indispensable for defining the necessary methods to meet requirements and setting acceptance criteria, among other challenges surrounding the verification and certification of such products.

To provide an understanding of the effort involved in defining a single requirement, a reasonable estimate is to consider the work of a team of approximately 12 individuals tasked with assigning the MoCs for an avionics system, for instance. It is not an exaggeration to suggest that such a task requires professionals with diverse technical backgrounds, such as experts in software development, software testing, hardware development, laboratory testing, flight testing, and safety assessment. Assuming just two specialists for each of these areas, we would have a team of 12 people. If this team spends, on average, only five minutes per requirement discussing and consulting standards to determine the appropriate MoCs, this translates to an effort of one man-hour per requirement. A generic avionics system typically has approximately 500 requirements. Thus, assigning MoCs for such a system would require an estimated 500 man-hours. An aircraft like the EMBRAER KC-390, on the other hand, can have tens of thousands of requirements!

2.4 System Theoretic Process Analysis (STPA)

STPA is a hazard analysis technique based on the System-Theoretic Accident Model and Processes (STAMP). This approach extends traditional accident causation models by incorporating system theory to address the complexities of modern systems (Leveson, 2002). In the aviation industry, probabilistic requirements are typically derived from the operational experience of similar systems (United States, 2024). However, such requirements are inadequate for software, given its deterministic nature and ubiquitous presence in aircraft systems. According to NASA (2011): "Software risk is the possibility of events by which the software used within a system may fail to successfully execute its mission-critical or safety-critical system functions, under the conditions that are to be covered according to the concept

of operation of the system designed to carry out the mission, and under the design envelope that is derived from, and consistent with, that concept of operation".

STPA addresses this gap by generating functional safety requirements for the entire system (Leveson; Thomas, 2018). STPA is an iterative process that evolves with the design, generating increasingly detailed requirements to mitigate Unsafe Control Actions (UCAs) identified during the analysis. This iterative nature allows analysts to refine the STPA analysis as needed throughout the design process (Rising; Leveson, 2018). The method's creator advocates for its application in the early stages of concept development (Leveson; Thomas, 2018). Figure 2.2 illustrates a framework for defining the purpose of the analysis, highlighting how hazards can be broken down into sub-hazards after identifying high-level system constraints.

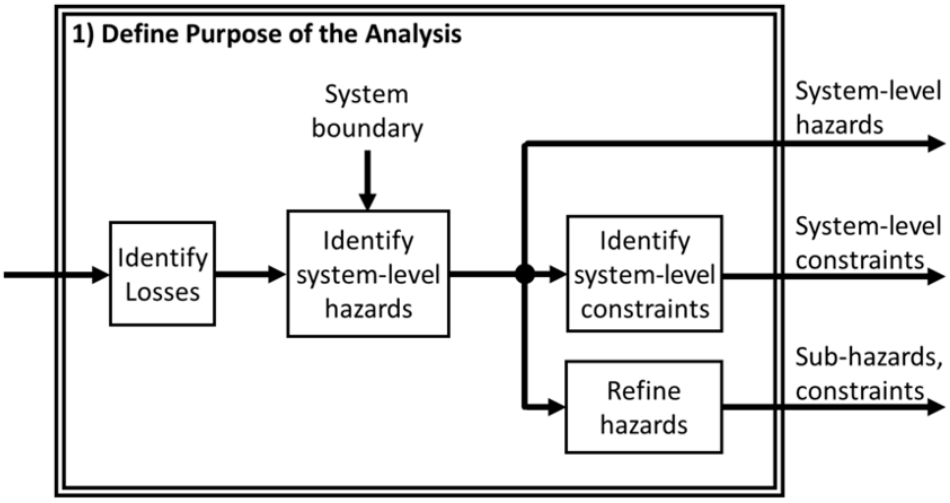


Figure 2.2 – Overview of defining the analysis purpose according to the STPA Handbook (Leveson; Thomas, 2018).

A concise overview of the four typical phases or steps of STPA (Leveson; Thomas, 2018) includes the four STPA phases:

Phase 1: Define the purpose of the analysis (including the system boundaries, the types of losses, and high-level hazards).

Phase 2: Model the system's control structure (identifying controllers, actuators, sensors, communication channels) and unsafe control actions (UCAs).

Phase 3: Identify loss scenarios (i.e., how UCAs could lead to hazards under certain conditions or system states).

Phase 4: Develop safety requirements and constraints to prevent or mitigate the loss scenarios.

Figure 2.3 illustrate these four phases.

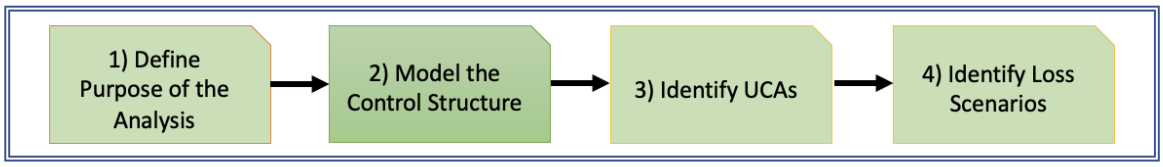


Figure 2.3 – The four phases of STPA (Leveson; Thomas, 2018).

The use of STPA over traditional risk analysis methods like FTA or Failure Modes and Effects Analysis (FMEA) is justified by its holistic approach, which addresses both component failures and unsafe interactions within complex systems (Mahajan; Bradley; Pasricha, 2017). Unlike traditional methods that often focus on linear cause-effect relationships and individual component failures, STPA employs a systems-theoretic approach, viewing the system as a dynamic control structure. This enables the identification of hazards arising from interactions among components, including software and human operators, which is crucial in modern, highly integrated systems. For example, research comparing STPA with FMEA and FTA has shown that STPA can identify additional causes of hazards that traditional methods might be missing. This includes issues related to software and system design as well as system integration (Ishimatsu *et al*., 2010). Additionally, STPA's capability to address complex interactions within a system makes it particularly effective in environments with high levels of integration and interdependencies, such as the aerospace and automotive industries (Bongirwar, 2021).

Moreover, STPA's systems-theoretic foundation allows it to consider both the functional and dysfunctional aspects of system interactions, providing a more comprehensive safety analysis. This method is also adaptable to modern engineering challenges, making it suitable for compliance with contemporary safety standards such as ISO 26262 in automotive systems (Bongirwar, 2021). Therefore, STPA's broader perspective and detailed analysis capabilities make it a superior choice for hazard analysis in complex systems (Thomas; Suo, 2015).

2.4.1 Key STPA Concepts

Accident / Loss: In STPA, the term *accident* (sometimes referred to as a *loss event*) denotes an undesirable outcome that negatively impacts people, equipment, the environment, or the organization. Examples in aerospace could include mid-air collisions, hull losses, or serious onboard injuries. A key distinction is that an *accident* or *loss event* is the realized consequence—i.e., it has already occurred or is in the process of occurring—whereas a *hazard* signifies a system state or condition that, if left uncorrected, could lead to an accident. In other words, the *loss* describes the eventual harm, while the *hazard* is the precursory situation that has the potential to produce that harm.

Hazard: A *hazard* is any system state or set of conditions that, under particular circumstances, could lead to an accident or loss. In STPA, hazards are defined not solely by component failures but by unsafe interactions or configurations that arise from the dynamic control relationships within the system. For instance, a hazard in an aircraft's flight management system might be "the inability to detect conflicting altitude commands," which could potentially lead to a loss event, such as a mid-air collision. The identification of hazards is fundamental in STPA because it informs the subsequent steps of analyzing how hazards can manifest and determining the controls or constraints necessary to prevent them.

Safety Constraint: A *constraint* is a safety-related requirement imposed on a system to prevent a hazard from leading to a loss event. These constraints can be both physical (e.g., enforced by hardware design) and procedural (e.g., enforced by operational policies or software logic). Unlike reliability requirements, which often emphasize the probability of component or subsystem failure, safety constraints in STPA target the systemic conditions and control actions that could create unsafe scenarios. Thus, while reliability requirements might specify a mean time between failures for a particular component, a safety constraint might dictate that "the autopilot shall not command a descent when the terrain avoidance system indicates an imminent ground collision risk." By addressing the broader context in which a system operates, STPA-derived constraints focus on preventing unsafe interactions rather than just mitigating component failures.

Unsafe Control Actions (UCAs): In STPA, an *unsafe control action* (UCA) is an action taken—or not taken—that could lead to a hazardous system state under specific conditions. Identifying UCAs is central to understanding how control loops in the system might inadvertently produce unsafe scenarios. For example, in a typical aerospace control

architecture, the flight management computer (controller) issues commands to the aircraft's control surfaces (actuators), which then affect the aircraft's flight path. A UCA might occur if the flight management computer fails to issue a corrective pitch-up command when the aircraft is approaching a critical angle of attack, thereby allowing a hazardous state to develop and increasing the risk of a stall. Alternatively, the flight management computer might issue a pitch-down command at an inappropriate time, placing the aircraft on a collision course with terrain. STPA identifies such actions by analyzing all possible control inputs, whether they are performed too early, too late, in the wrong order, or not at all. By systematically examining the control loops and their feedback mechanisms, analysts can pinpoint when a nominally valid action could become unsafe due to timing, context, or mismatches in communication among system components.

Loss Scenarios: In STPA, *loss scenarios* provide a structured explanation of how UCAs can evolve into actual accidents or losses. More specifically, they trace the causal pathways that connect identified hazards to potential system failures, human errors, or unexpected interactions in the operational environment. By examining the context, timing, and conditions under which a UCA occurs, analysts can illuminate the precise mechanisms that allow a hazard to materialize into a loss event. For instance, in an aircraft flight control system, a loss scenario might describe how a momentarily incorrect pitch command, combined with degraded sensor feedback and delayed pilot intervention, leads to a loss of stable flight and ultimately results in an accident. Thus, loss scenarios are critical for capturing the dynamic nature of control loops, including where and why they might break down or provide inadequate safety margins. By articulating these scenarios, STPA ensures that safety efforts target not just the hazard itself but also the full chain of events or conditions that would enable it to escalate into a system-wide failure.

Requirements: Once loss scenarios have been identified and traced back to their underlying hazards and UCAs, *requirements* are developed to prevent or mitigate these unsafe conditions. In STPA, these requirements may emerge at multiple levels:

1. **High-Level Safety Requirements** – Derived directly from the overall safety constraints that govern the acceptable boundaries of system operation. These might specify, for example, that "the control system shall ensure continuous monitoring of sensor integrity and provide immediate corrective actions when anomalies are detected." Such high-level requirements often guide system architecture and design decisions early in the development process.

2. **Detailed Requirements** – Elicitated from specific loss scenarios, which call for targeted corrective or preventive measures at a finer level of granularity. For instance, if a loss scenario highlights the possibility of late or missed pilot alerts in a degraded mode, a more detailed requirement might state that "the software shall generate a time-sensitive warning within one second of detecting conflicting guidance to ensure pilot awareness."

These STPA-derived requirements are distinct in their focus on mitigating unsafe interactions, rather than solely on preventing component failures. They extend beyond traditional reliability targets by emphasizing proactive control mechanisms, timely responses to anomalies, and comprehensive procedures that help maintain the system within safe operational limits. By systematically mapping each requirement to identified hazards and loss scenarios, STPA ensures traceability throughout the development lifecycle and supports continuous refinement of safety measures as the design evolves.

2.4.2 STPA versus Classic Safety Analysis Methods

Traditional hazard analysis methods, such as Failure Modes and Effects Analysis (FMEA) and Fault Tree Analysis (FTA), focus on identifying and mitigating risks primarily through linear cause-effect relationships and component-based reliability assessments. FMEA, for example, systematically examines potential failure modes within individual components and assesses their effects on the larger system, while FTA uses a top-down approach to identify combinations of failures that could lead to a particular undesired event (Ericson, 2015). Although these methods have been widely applied in various industries, they are often limited when dealing with complex, integrated systems that exhibit non-linear interactions and emergent behaviors. Unlike FMEA and FTA, STPA adopts a systems-theoretic perspective that views the system as a dynamic control structure rather than a collection of independent components (Leveson, 2016).

STPA differs by focusing on unsafe interactions and control actions that may arise from system-wide dependencies and environmental interactions. This shift is particularly crucial in modern safety-critical systems, such as those found in aerospace and defense, where software and human factors play a substantial role. Research by Ishimatsu *et al*. (2014) has demonstrated that STPA can identify additional causes of hazards that traditional methods may overlook, particularly in cases involving software or complex control algorithms. By encompassing both functional and dysfunctional interactions within the system, STPA is more

adept at revealing hazards associated with emergent properties, which are often invisible to linear analysis techniques (Leveson; Thomas, 2018). Consequently, while FMEA and FTA remain useful in certain contexts, STPA offers a more holistic and adaptable approach to hazard analysis in environments with high levels of integration and interdependencies, underscoring its applicability in contemporary engineering challenges.

2.4.3 STPA Practical Applications

The versatility of STPA extends beyond traditional aerospace applications, proving beneficial in a variety of complex systems that require high safety and reliability standards. In the aviation industry, STPA has been employed to assess hazards in flight control systems, engine control, and air traffic management, areas where system interactions and human factors are particularly intricate. One notable application is its use in analyzing NextGen air traffic management systems in the United States, where STPA helped identify interaction-based hazards that could arise from the integration of new automation technologies (Fleming *et al*., 2013). This example illustrates STPA's capacity to adapt to evolving technological landscapes and address safety risks associated with system upgrades.

Beyond aerospace, STPA has also demonstrated its efficacy in other sectors. In the automotive industry, STPA is utilized to meet the safety requirements set forth by ISO 26262, the international standard for functional safety in road vehicles (Bongirwar, 2021). With the increase in autonomous and semi-autonomous vehicle systems, STPA's ability to account for software-driven hazards and complex interactions has become invaluable. Furthermore, STPA has been applied in medical device manufacturing, where failures or unsafe interactions can have life-threatening consequences (Alemzadeh, 2016). For instance, in the development of infusion pumps, STPA was used to address potential hazards involving software-controlled dosing, thereby enhancing the safety and reliability of these critical medical devices. By highlighting the range of sectors in which STPA has been successfully implemented, these examples reinforce the method's adaptability and its critical role in addressing safety challenges across various high-stakes environments.

2.4.4 Limitations of STPA

While STPA provides a powerful framework for identifying hazards in complex systems, it also has certain limitations that should be acknowledged. One primary limitation is the need for significant expertise in system dynamics and control theory. STPA relies on

understanding how control actions propagate through hierarchical control structures, which may be challenging for practitioners without a background in systems thinking (Leveson; Thomas, 2018). Furthermore, STPA analyses can be resource-intensive, requiring extensive collaboration among stakeholders to define control structures and unsafe control actions comprehensively. This demand for interdisciplinary knowledge and stakeholder involvement can pose challenges in projects with limited resources or expertise (Ishimatsu *et al*., 2010).

Another limitation is the scalability of STPA in extremely large or complex systems. While STPA is effective for identifying high-level hazards and unsafe interactions, applying it to systems with thousands of components or highly decentralized control structures can lead to challenges in maintaining a manageable scope. As noted by Abdulkhaleq and Wagner (2015), analyzing complex cyber-physical systems with STPA may result in a high volume of potential hazards, complicating the prioritization and mitigation process. Additionally, STPA is not inherently designed to quantify risks probabilistically, a feature often required in certain regulatory environments. Although probabilistic risk assessments can be performed in conjunction with STPA, the method itself does not provide a built-in mechanism for risk quantification, which might limit its applicability in industries where quantitative risk data is a regulatory requirement (Leveson, 2016). Thus, while STPA offers comprehensive insights into system hazards, it is not without challenges, particularly when applied to very large systems or in settings demanding quantitative risk metrics.

Furthermore, unlike FMEA and FTA, STPA is not required by certification authorities and is not a substitute for them. This may discourage organizations interested in using STPA since they often need to exert considerable effort to comply with certification standards.

2.4.5 STPA Relationship to Safety and Certification Requirements

Although not mandatory, STPA's framework aligns closely with various safety standards and certification requirements across multiple industries, making it a valuable tool for regulatory compliance. In the aerospace industry, for instance, STPA can be integrated into the processes defined by DO-178C, a key standard for the certification of avionics software, as well as ARP4754A, which focuses on the development of aircraft systems (RTCA, 2011). By enabling a systematic analysis of unsafe interactions and control actions, STPA helps generate safety requirements that are essential for meeting regulatory criteria in these standards. For example, DO-178C emphasizes the need for rigorous hazard identification and control, which STPA addresses through its systems-theoretic approach to

hazard analysis, particularly in relation to software-driven interactions (Leveson, 2016).

In the automotive sector, STPA has been employed to support compliance with ISO 26262, the standard for functional safety in road vehicles. ISO 26262 mandates a detailed hazard analysis and risk assessment to ensure the safe operation of electronic and software-controlled systems (Bongirwar, 2021). STPA's ability to identify hazards not only from component failures but also from unsafe interactions within the control structure makes it particularly suited for generating the safety requirements necessary to achieve ISO 26262 compliance. Moreover, STPA has also been explored for applications in the railway industry, where safety standards such as EN 50126 require the development of safety requirements for complex, integrated control systems (Almeida; Fonseca, 2014). Through its structured hazard analysis process, STPA offers a method to systematically derive these requirements, thereby facilitating compliance with industry-specific safety standards. Overall, STPA's compatibility with key certification frameworks across aerospace, automotive, and rail industries highlights its efficacy in enhancing system safety and supporting regulatory compliance.

2.5 Lifecycle of Aerospace Systems in the Brazilian Air Force

The DCA 400-6, published by the Brazilian Air Force (FAB) in 2007, establishes a structured lifecycle management model for aeronautical systems and materials. This directive outlines the stages from conception through disposal, detailing processes and responsibilities to ensure the operational readiness, efficiency, and sustainability of systems throughout their lifecycle (Brasil, 2007).

The DCA 400-6 divides the systems' lifecycle into the following phases:

Conception Phase: This initial stage focuses on identifying operational needs or technological and economic opportunities that justify developing or acquiring new systems or materials. The first step is the creation of the NOP (*Necessidade Operacional* – Operation Need), a document formalized by the ODSA (*Órgãos de Direção Setorial e de Assistência Direta e Imediata ao Comandante da Aeronáutica* – Sectoral Management Bodies and Direct and Immediate Assistance to the Air Force Commander), which specifies identified deficiencies. This document is then reviewed by the Brazilian Air Force's General Staff (EMAER) to assess compatibility with strategic plans and budget allocations.

Feasibility Phase: In this phase, a thorough analysis of alternatives and lifecycle feasibility is conducted, encompassing technical, economic, and political assessments. The

objective is to define the optimal strategy for developing or acquiring the system, considering cost-effectiveness and potential partnerships with national or international companies. Evaluations in this phase also include opportunities for international cooperation and technology transfer. At this moment, the ROP (*Requisitos Operacionais* – Operational Requirements), a set of high-level/system-level requirements, is prepared.

Definition Phase: Upon demonstrating feasibility, the definition phase refines the system specifications and prepares the RTLI (*Requisitos Técnicos, Logísticos e Industriais* – Technical, Logistics, and Industrial Requirements), a breakdown of ROP requirements. Detailed engineering studies and simulations are carried out, followed by the selection of development or acquisition partners, with contracts stipulating localization and technology transfer clauses. This process ensures the system or material will meet the FAB's operational requirements.

Development/Acquisition Phase: The development phase involves executing the plans, including verification, testing, and system certification. Here, the AVOP (*Availação Operacional* - Operational Evaluation) assesses whether the system complies with functional, operational, and logistical requirements before production in mass. This process ensures that the system is fully prepared for production and deployment and meets all established standards.

Production Phase: Following successful development, the system enters production, with supplier contracts formalized, including logistics and equipment supply. The production is supervised by a GAC (*Grupo de Acompanhamento e Controle* – Monitoring and Control Group), ensuring the final product adheres to the FAB's contractual specifications and operational requirements. This phase is critical to maintaining uniformity and quality in the materials procured.

Deployment Phase: In the deployment phase, the FAB coordinates the system's entry into operational service, establishing the necessary logistics. Several sector-specific plans are developed, such as the Employment Support Plan, Supply and Maintenance Plan, and Infrastructure Plan, ensuring the system is fully equipped to support the FAB's operational needs, backed by a robust logistical structure.

Utilization Phase: This phase marks the system's operational use within the FAB, where maintenance and support strategies are implemented to ensure continuous performance. Monitoring logistics and maintenance allows the FAB to sustain availability and reliability,

minimizing downtime. Data on operational performance also contributes to future decisions regarding upgrades or deactivation.

Revitalization/Modernization Phase: To address wear and obsolescence, systems may undergo upgrades or modernization. This can involve technical modifications and logistical improvements that extend the system's lifecycle while maintaining relevance and effectiveness. Decisions on modernization are informed by detailed cost-benefit assessments and the FAB's strategic needs.

Deactivation Phase: When a system reaches the end of its service life, deactivation and disposal processes are enacted, effectively concluding its lifecycle. This involves careful planning to reallocate resources efficiently and discontinue the system according to legal and environmental standards. This phase ensures the FAB minimizes costs associated with obsolete systems, reallocating resources to more modern assets.

DCA 400-6 provides a structured framework for managing systems and materials within the FAB, emphasizing thorough planning and control at each stage of the lifecycle. Although robust, this directive could benefit from updates to incorporate modern lifecycle management practices, particularly with respect to evolving technologies and safety methodologies (Brasil, 2007). Thus, while DCA 400-6 remains a foundational tool, it can be revised to align with contemporary systems engineering practices and certification standards (Silva, 2021).

2.6 Artificial Intelligence (AI)

Artificial Intelligence (AI) is broadly defined as the field of computer science and engineering focused on creating machines capable of performing tasks that would normally require human intelligence. This includes not only simple data processing but also complex cognitive tasks such as learning, problem-solving, perception, and decision-making (Russell; Norvig, 2020). AI can be broken down into several key domains, each representing different aspects of human intelligence replicated by machines. These domains span from fundamental processes like Machine Learning (ML), which enables machines to learn from data to more intricate applications, such as Natural Language Processing (NLP) and Autonomous Systems, which allow machines to understand, generate language, and even navigate physical spaces independently.

One of AI's primary objectives is to increase efficiency and accuracy in decision-making processes across various industries. In sectors like healthcare, AI-driven systems analyze complex datasets to assist with diagnosis and treatment planning, potentially enhancing patient outcomes (Esteva *et al*., 2017). Similarly, in finance, AI applications streamline operations, detect fraud, and provide predictive insights that can influence market strategies (Ngai *et al*., 2011). In these contexts, AI systems augment human abilities, acting as tools that enhance productivity rather than replacing human input altogether.

The scope of AI is continually expanding due to advancements in data availability, computational power, and algorithmic development. AI today includes numerous subfields, such as computer vision, speech recognition, robotics, and cognitive computing. Each of these areas represents a specialized focus within AI research, collectively contributing to a broad understanding of intelligence (Russell; Norvig, 2020). For instance, computer vision enables machines to interpret and process visual data, allowing applications like autonomous vehicle navigation (Gao *et al*., 2024). Meanwhile, cognitive computing aims to simulate human thought processes in a computerized model, bridging the gap between human-like reasoning and AI decision-making (Megha; Madhura; Sneha, 2018).

Furthermore, AI systems can be categorized into different levels of intelligence: Narrow AI, which specializes in performing specific tasks, and General AI, a more ambitious concept that seeks to create machines capable of understanding and performing any intellectual task that a human can. AI applications are primarily narrow in scope, focusing on specialized functions, such as image classification or natural language understanding. General AI, also referred to as “strong AI,” remains largely theoretical, although it represents a long-term goal for some researchers (Goertzel, 2014).

AI has emerged as a transformative technology with the potential to reshape various aspects of society. The advent of autonomous systems, such as self-driving cars, represents one of AI’s most complex applications, requiring a synthesis of ML, computer vision, sensor fusion, and real-time decision-making (Russell; Norvig, 2020). Autonomous systems highlight the multi-faceted scope of AI, where machine learning algorithms analyze vast amounts of data to enable vehicles to navigate complex environments autonomously. This capability demonstrates AI's potential to perform tasks traditionally requiring high levels of human expertise and adaptability.

2.6.1 Brief History of AI

The origins of AI can be traced back to early theoretical work on computation and logic. Alan Turing, often regarded as the father of AI, proposed in his seminal 1950 paper *Computing Machinery and Intelligence* the idea of machines that could “think” by demonstrating intelligent behavior indistinguishable from that of a human. Turing introduced the concept of the “imitation game,” now known as the Turing Test, to evaluate a machine’s ability to exhibit intelligent behavior (Turing, 1950). This marked one of the first formal explorations into the possibility of artificial intelligence, laying the groundwork for future developments.

The formal establishment of AI as a field of study occurred in 1956 at the Dartmouth Conference, where researchers like John McCarthy, Marvin Minsky, and Claude Shannon gathered to explore “the possibility of creating machines that can solve problems currently reserved for humans” (Mccarthy *et al*., 2006). This conference was instrumental in defining AI’s objectives and catalyzing research that led to early breakthroughs in fields such as symbolic reasoning and rule-based systems.

During the 1960s and 1970s, AI research focused largely on symbolic AI and expert systems, which attempted to encode human knowledge into rule-based frameworks for tasks such as medical diagnosis and geological analysis (De Kleer, 1984). Expert systems like DENDRAL and MYCIN represented some of the first successful applications of AI, demonstrating that machines could assist in decision-making by following structured sets of rules. However, these systems were limited by the rigidity of rule-based logic and struggled with tasks that required flexibility or learning from data, eventually leading to what is now referred to as the "AI winter"—a period of reduced funding and interest due to unmet expectations and limited computational capabilities (Crevier, 1993).

The field was revitalized in the 1980s and 1990s with advancements in computational power and the development of neural networks, which introduced a new paradigm in AI research by attempting to mimic the architecture of the human brain. Early neural networks, such as the Perceptron developed by Rosenblatt in 1957, laid the groundwork for later breakthroughs in machine learning by demonstrating that machines could “learn” through exposure to data rather than pre-defined rules. However, it was not until the backpropagation algorithm was introduced in the 1980s that neural networks became feasible for complex problem-solving tasks, sparking renewed interest in AI (Crevier, 1993).

The rise of machine learning in the 1990s marked a significant shift in AI research. Instead of focusing solely on symbolic reasoning, researchers began exploring statistical methods and data-driven approaches that allowed machines to identify patterns and make predictions. This led to substantial progress in fields like speech recognition and natural language processing, where algorithms could be trained on large datasets to improve performance autonomously (Jurafsky; Martin, 2024). The availability of large-scale datasets, such as ImageNet, further accelerated machine learning research, enabling the development of deep learning algorithms that outperformed traditional AI techniques in areas like image classification and speech recognition (Deng *et al*., 2009).

The advent of deep learning in the early 2010s represented another transformative step for AI. Deep learning leverages multi-layered neural networks, or "deep networks," which can automatically learn hierarchical representations of data. This approach, pioneered by researchers such as Geoffrey Hinton, Yann LeCun, and Yoshua Bengio, demonstrated remarkable success in tasks like image and speech recognition, leading to a resurgence in AI research and applications (Lecun; Bengio; Hinton, 2015). Notably, the introduction of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) provided specialized architectures for handling image and sequence data, expanding the range of applications where AI could achieve human-level performance or better.

Most recently, the development of transformer-based architectures has revolutionized natural language processing and has enabled the creation of Large Language Models (LLMs). This architecture was introduced by Vaswani *et al*. (2017) in the paper *Attention is All You Need*, one of the greatest milestones of recent AI history. Transformers utilize self-attention mechanisms that allow models to process and generate language with unprecedented accuracy and coherence. LLMs, self-attention, and Transformers were essential for this work and, therefore, require a dedicated section (Section 2.7). These advancements have not only increased the scope of AI's applications but have also prompted discussions about the ethical and societal implications of powerful AI models (Bender *et al*., 2021).

2.6.2 Machine Learning

Machine Learning (ML) is a subfield of Artificial Intelligence (AI) that focuses on developing algorithms and statistical models enabling computers to learn from data and improve their performance without explicit programming. The core idea behind ML is to automate pattern recognition and decision-making processes by utilizing computational

models that analyze and interpret complex datasets (Mitchell, 1997). ML has become integral in a variety of applications, from image recognition and natural language processing to predictive analytics and autonomous systems (Goodfellow; Bengio; Courville, 2016).

ML techniques are generally divided into three main categories: supervised learning, unsupervised learning, and reinforcement learning, each with distinct mechanisms and applications. These categories form the foundational basis of ML and support a wide range of advanced AI systems, including Large Language Models (LLMs), where the learning process relies heavily on these ML paradigms (Bishop, 2006).

Supervised learning is one of the most widely used ML approaches, involving the training of models on labeled datasets where each data point is paired with the correct output. In this setup, the algorithm learns by minimizing the error between its predictions and the actual labels, adjusting its internal parameters through iterative optimization techniques like gradient descent (Goodfellow; Bengio; Courville, 2016). A prominent example of supervised learning includes classification tasks, where the model categorizes inputs into predefined classes, such as spam detection in emails or image classification (Lecun; Bengio; Hinton, 2015). Supervised learning is essential for tasks requiring high precision and interpretability, as it leverages vast amounts of labeled data to produce accurate, reliable predictions (Russell; Norvig, 2020). Figure 2.4 shows a classic and fun (practically an internet meme) sample of supervised learning for a classification problem. With enough amount of data, you can make an application learn to differentiate images so similar like this example.

Figure 2.4 – The classic ML's muffin-chihuahua classification problem (Nath *et al*., 2024).

Unsupervised learning, in contrast, deals with unlabeled data, meaning the algorithm identifies patterns or structures within the dataset without any guidance on expected outcomes. This approach is often used for clustering, dimensionality reduction, and anomaly detection (Prince, 2024). Clustering algorithms, for instance, group similar data points together based on specific characteristics, which are helpful in customer segmentation and image compression (Jain, 2010). Principal Component Analysis (PCA) and k-means clustering are classic examples of unsupervised learning methods that allow the discovery of underlying patterns and relationships within data (Bishop, 2006). Unsupervised learning is valuable for exploratory data analysis and discovering insights from datasets where labels are unavailable or too costly to obtain. Generative unsupervised models are designed to create new data samples that closely resemble the training data, making them statistically indistinguishable from it. Certain generative models define the probability distribution of the input data explicitly, allowing new samples to be generated by drawing from this defined distribution (Prince, 2024). Figure 2.5 brings some examples of their result.

Figure 2.5 – Generative models for images. On the left: two images were generated by a model trained on cat pictures. These are not real cats but samples produced by a probabilistic model. On the right are two images generated by a model trained on images of buildings (Adapted from Karras *et al*., 2020).

Reinforcement learning (RL) is a unique ML paradigm where an agent learns by interacting with an environment and receiving feedback through rewards or penalties based on its actions. The goal of RL is to maximize cumulative rewards over time, which requires the agent to develop policies that balance the exploration of new actions and exploitation of known strategies (Sutton; Barto, 2015). Unlike supervised and unsupervised learning, RL is particularly useful in sequential decision-making problems and environments with delayed rewards, such as robotics, gaming, and autonomous driving (Silver *et al*., 2016). The RL framework typically includes elements such as a policy (the agent's behavior), a reward function, and a value function to evaluate the expected return of states or actions (Sutton; Barto, 2015). RL's trial-and-error learning process makes it highly suitable for adaptable and resilient applications in dynamic, uncertain environments.

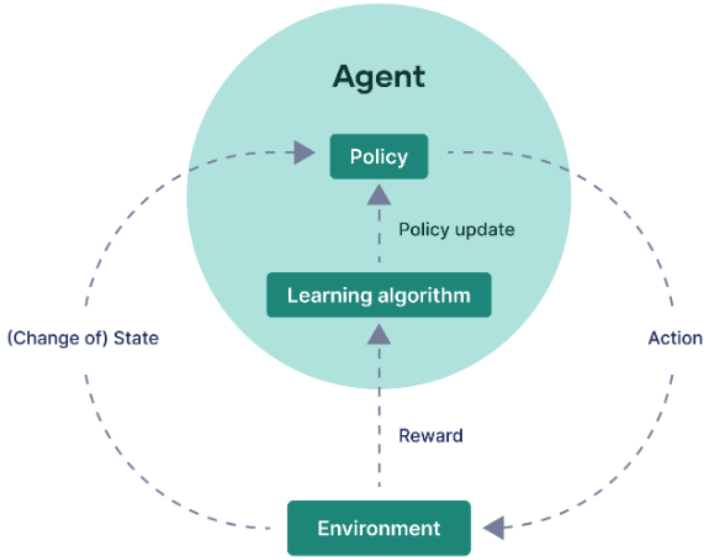


Figure 2.6 – The general framework of reinforcement learning (Nikolopoulou, 2023).

LLMs utilize all three machine learning types, combining unsupervised learning for general language understanding, supervised learning for task-specific optimization and reinforcement learning to improve human alignment and response quality.

2.6.3 Neural Networks

Neural networks are foundational to modern deep learning and form the underlying structure of Large Language Models (LLMs). These networks are computational models inspired by the human brain, composed of layers of interconnected nodes, or neurons, which process and transform information (Goodfellow; Bengio; Courville, 2016). The basic structure of a neural network consists of an input layer, hidden layers, and an output layer, with each layer containing nodes that perform specific calculations. Each node receives inputs, processes them using weights and biases, and applies an activation function to introduce non-linearity, allowing the network to capture complex patterns in data (Lecun; Bengio; Hinton, 2015).

The activation functions are essential for controlling the output of each neuron. Commonly used activation functions include the rectified linear unit (ReLU), sigmoid, and *tanh* functions. ReLU, for example, has become a standard in deep learning due to its ability to alleviate the vanishing gradient problem, which often impedes the training of deep networks (Lecun; Bengio; Hinton, 2015). These foundational elements – nodes, layers, and activation functions – allow neural networks to perform complex transformations on input data, making them suitable for a wide range of applications, from image recognition to natural language processing.

Several key architectures have been developed to address different types of data and tasks, each with distinct characteristics and advantages. The three most influential architectures are Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformers (Prince, 2024).

Convolutional Neural Networks (CNNs): CNNs are designed specifically for spatial data, making them particularly effective in image processing tasks. CNNs use convolutional layers, which apply filters to local regions of the input data, capturing spatial hierarchies and patterns. By learning spatial features, CNNs can recognize complex structures within images, such as edges and textures, with higher efficiency than traditional fully connected networks. CNNs are often applied in computer vision tasks like object detection and facial recognition, where spatial structure is crucial for performance. However, due to their design, CNNs are

less suitable for sequential data such as language, where temporal dependencies play a significant role (Prince, 2024).

Recurrent Neural Networks (RNNs) are tailored for sequential data, such as text, speech, or time-series data. Unlike CNNs, RNNs incorporate recurrent connections that allow them to maintain a memory of previous inputs, capturing temporal dependencies over time. This characteristic makes RNNs effective for tasks requiring context, such as language modeling and translation. However, RNNs suffer from the vanishing gradient problem, particularly when dealing with long sequences, which limit their effectiveness in handling long-term dependencies. To address this, variants like the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) were introduced, adding mechanisms to maintain information over longer sequences. Despite these advancements, RNNs still face scalability challenges, especially in processing large-scale data needed for training LLMs (Prince, 2024).

The Transformer architecture represents a breakthrough in neural network design, addressing many limitations of previous architectures, particularly in processing sequential data. Introduced by Vaswani *et al*. (2017), Transformers rely on self-attention mechanisms rather than recurrent connections, allowing them to capture relationships between words in a sequence regardless of their distance from each other. This architecture computes attention scores to focus on relevant parts of the input sequence, facilitating parallel processing of data and dramatically reducing training times. Transformers have become the standard in natural language processing, powering models like BERT (Devlin *et al.*, 2018) and GPT (Brown *et al*., 2020), and directly supporting the development of LLMs by enabling efficient training on massive datasets.

The evolution towards Transformer-based architecture has transformed the field of NLP, making it possible to develop language models capable of understanding and generating coherent text across diverse contexts. Unlike RNNs, which process data sequentially, Transformers' parallel processing capability makes them highly scalable, essential for training on the extensive datasets required for LLMs. Furthermore, the self-attention mechanism allows Transformers to capture complex dependencies within language, making them particularly adept at understanding nuanced contextual relationships critical for tasks such as translation, summarization, and question-answering.

2.6.4 Rise of the Robot Lawyers

The article "Rise of the Robot Lawyers" (Markovic, 2019) represents a watershed moment in this work. In 2021, after reading this article, the idea of applying NLP techniques to the verification processes of aerospace systems, explored in this thesis, was conceived.

Markovic examines the implications of artificial intelligence (AI) on the legal profession. It critiques the widespread narrative suggesting that AI will replace lawyers, arguing instead that while technological advancements will augment legal practice, they are unlikely to displace lawyers entirely. The author contends that many legal tasks require abstract reasoning, problem-solving, and contextual judgment, which remain beyond the capacity of current AI systems.

The author analyzes empirical data to show that lawyer employment and wages have grown steadily over the past two decades, contrasting the experience of the legal profession with industries like manufacturing, where automation has significantly reduced jobs. It further argues that legal automation has primarily affected routine tasks, such as document review, but has simultaneously created new opportunities in areas like data privacy and legal analytics.

Finally, Markovic also highlights normative considerations, emphasizing the societal role of lawyers in safeguarding justice and public interest, which cannot be fully replaced by AI. The article concludes that lawyers, far from being rendered obsolete, are uniquely positioned to thrive alongside AI by leveraging it as a tool to enhance efficiency and accessibility in legal services.

The similarities between the activities of legal professionals and aerospace systems compliance engineers are greater than one might initially assume. To construct a legal argument, a lawyer must thoroughly examine the case files and find support within the legal framework. Similarly, a compliance engineer must gather evidence from the results of various verification activities (e.g., analyses, calculations, simulations, tests, etc.) in order to create a technical rationale to demonstrate compliance with system requirements, adhering to a normative basis such as defense aerospace standards like MIL-STD-882E (United States, 2012) and MIL-HDBK-516C (United States, 2014).

However, candidate technology solutions to automate requirements elicitation and MoCs assignments in 2021 would have required a herculean endeavor and lack of novelty. By late 2022, an extraordinary tool had become publicly available and widely popular: ChatGPT.

This application captured global attention by showcasing the potential of Large Language Models (LLMs). The following Section delves into these models, explaining their mechanisms and exploring how to leverage their hyperparameters and characteristics to achieve optimal results.

2.7 Large Language Models (LLMs)

The rapid development and widespread adoption of LLMs like ChatGPT have profoundly impacted the field of Natural Language Processing (NLP). These models, trained on vast troves of textual data, have demonstrated remarkable abilities in tasks ranging from text generation to reasoning about complex concepts (Brown *et al*., 2020; Radford *et al*., 2019).

At the heart of LLMs lies an innovative approach to representing and processing natural language: word vectors. Word vectors, also known as word embeddings, are numerical representations of words that capture their semantic and syntactic relationships (Mikolov *et al*., 2013). This powerful technique allows language models to reason about words and their meanings in a more nuanced and context-sensitive manner, as evidenced by their ability to perform analogical reasoning tasks (Pennington; Socher; Manning, 2014).

Interestingly, recent studies have suggested that the training process of LLMs, which involves predicting the next word in a sequence, may lead to the emergence of high-level reasoning capabilities, such as theory of mind (Cross *et al*., 2024; He *et al*., 2023). This phenomenon, where complex skills appear to arise spontaneously as a byproduct of language modeling, highlights the potential for LLMs to exhibit increasingly sophisticated cognitive abilities (Lu *et al*., 2024).

However, the inner workings of these powerful models remain largely opaque, posing challenges for researchers seeking to fully understand their decision-making processes (Bau *et al*., 2020; Brown *et al*., 2020). Ongoing efforts to shed light on the mechanisms underlying LLM performance, such as the analysis of attention patterns and the role of different network components, are crucial for advancing our understanding of these systems and guiding future developments in the field (Liao; Vargas, 2024).

As the research community continues to explore the capabilities and limitations of LLMs, the integration of these models into real-world applications and the mitigation of potential biases, safety, and ethical concerns will be crucial areas of focus (Pressman *et al*.,

2024). The rapid progress in this field promises to have far-reaching implications for a wide range of domains, from natural language processing to artificial general intelligence (Hagos; Battle; Rawat, 2024).

2.7.1 How do LLMs Work?

Tokens are the basic units into which text is decomposed for processing by an LLM. Tokenization involves breaking down text into sub-word units, words, or characters, depending on the tokenizer's design. This process balances the model's ability to handle rare and common words efficiently (Prince, 2024).

For example, the sentence "The cat sat on the mat." might be tokenized as ["The", " cat", " sat", " on", " the", " mat", "."]. Tokens serve as the input data for the model, allowing it to convert raw text into numerical representations that can be processed computationally.

Embeddings transform tokens into continuous vector representations, allowing the model to process textual data numerically. Each token is mapped to a fixed-dimensional vector, capturing semantic and syntactic information.

Let V be the vocabulary of tokens, and d be the embedding dimension. The embedding function $E: V \rightarrow \mathbb{R}^d$ assigns each token $t \in V$ to a vector $\boldsymbol{e_t} \in \mathbb{R}^d$. The embeddings are typically learned during training and are stored in an embedding matrix $\mathbf{E} \in \mathbb{R}^{|V| \times d}$, where each row corresponds to a token's vector representation. Figure 2.7 provides a visual understanding of how the vectorization of tokens works. In simple terms, the embedding matrix converts words into numbers, and an LLM's neural network uses these numbers to learn and understand language through the mechanism of self-attention.

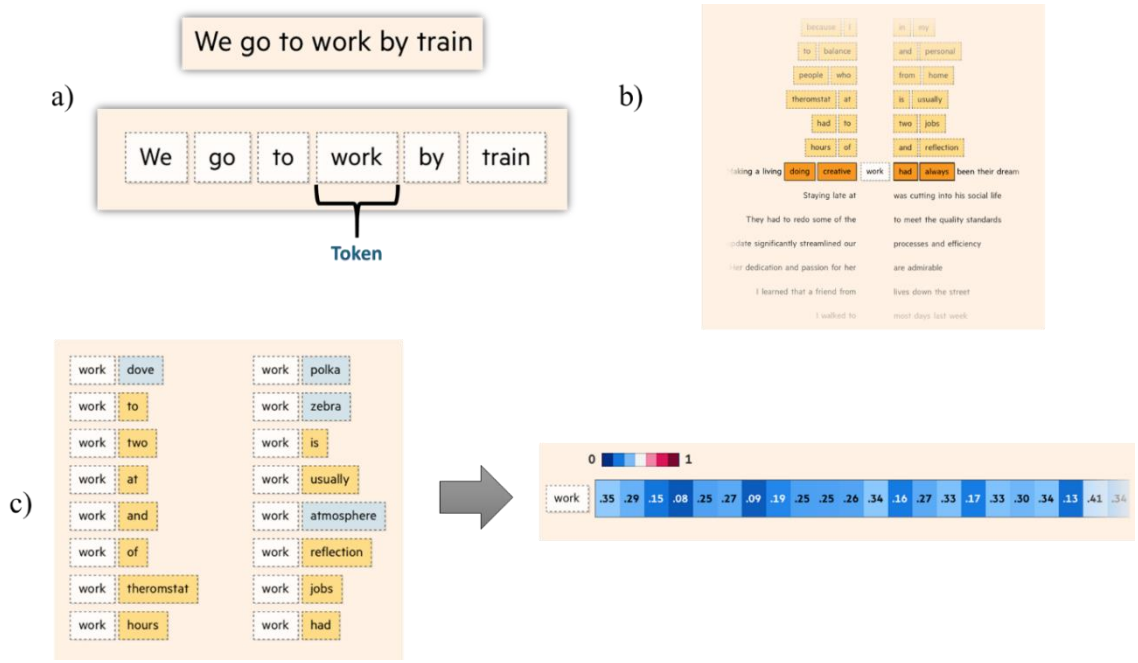


Figure 2.7 – Transforming words in vectors. a) Each text smaller unit is considered a token. b) The embedding function E transforms the frequency in which each token t∈V is found with the other tokens in V. c) Those frequencies make the embedding vectors (Financial Times, 2023).

Self-attention allows the model to weigh the importance of different tokens in a sequence relative to each other. It enables the model to capture dependencies irrespective of their distance in the sequence. Given an input sequence of embeddings $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]^{\mathrm{T}}$, where the exponent 'T' represents the transposition operator, the self-attention mechanism computes the output as:

1. **Linear Projections** – Compute queries **Q,** keys **K**, and values **V**:

$$\mathbf{Q} = \mathbf{X}\boldsymbol{W}^{\boldsymbol{Q}}, \mathbf{K} = \mathbf{X}\boldsymbol{W}^{\boldsymbol{K}}, \mathbf{V} = \mathbf{X}\boldsymbol{W}^{\boldsymbol{V}}, \tag{2.1}$$

where $\boldsymbol{W}^{\boldsymbol{Q}}, \boldsymbol{W}^{\boldsymbol{K}}, \boldsymbol{W}^{\boldsymbol{V}} \in \mathbb{R}^{d \times d_k}$ are weight matrices.

2. **Scaled Dot-Product Attention**:

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\mathbf{V} \tag{2.2}$$

The term $\sqrt{d_k}$ is a scaling factor to prevent the dot products from becoming too large, which could push the softmax function into regions with extremely small gradients. Queries **Q** represent the current token's intent. Keys **K** represent the tokens' attributes to be matched against queries. Values **V** are the information to be aggregated.

The **softmax** function is a mathematical function that converts a vector of real numbers into a probability distribution. It is widely used in machine learning (Prince, 2024), particularly in neural networks, to map raw output scores (also known as logits) to probabilities that sum up to 1. This makes it suitable for tasks like multi-class classification and attention mechanisms. In models like transformers, softmax is used to calculate attention weights, determining how much focus the model should give to different parts of the input. The softmax function is smooth and differentiable, which is essential for optimization algorithms like gradient descent used in training neural networks.

The input sequence is initially processed through the Input Embedding and Positional Encoding layers, producing an encoded representation for each word that captures both its semantic meaning and positional context. This encoded representation is then provided to the Query, Key, and Value parameters of the Self-Attention mechanism within the first Encoder. The Self-Attention mechanism computes a new encoded representation for each word, now incorporating attention weights that quantify the relevance of other words in the sequence to each specific word. As this data progresses through each subsequent Encoder in the stack, each Self-Attention module further enhances the word representations by integrating additional attention scores, thereby progressively refining the contextual understanding of each word in relation to the entire sequence. Figure 2.8 illustrates how, by consolidating all the queries into a single matrix, we transform the computation into a matrix multiplication rather than performing separate vector-to-matrix multiplications for each query individually. This approach ensures that each query is processed entirely independently of the others. The inherent parallelism is thus achieved effortlessly by employing matrix operations and supplying all the input tokens or queries simultaneously.

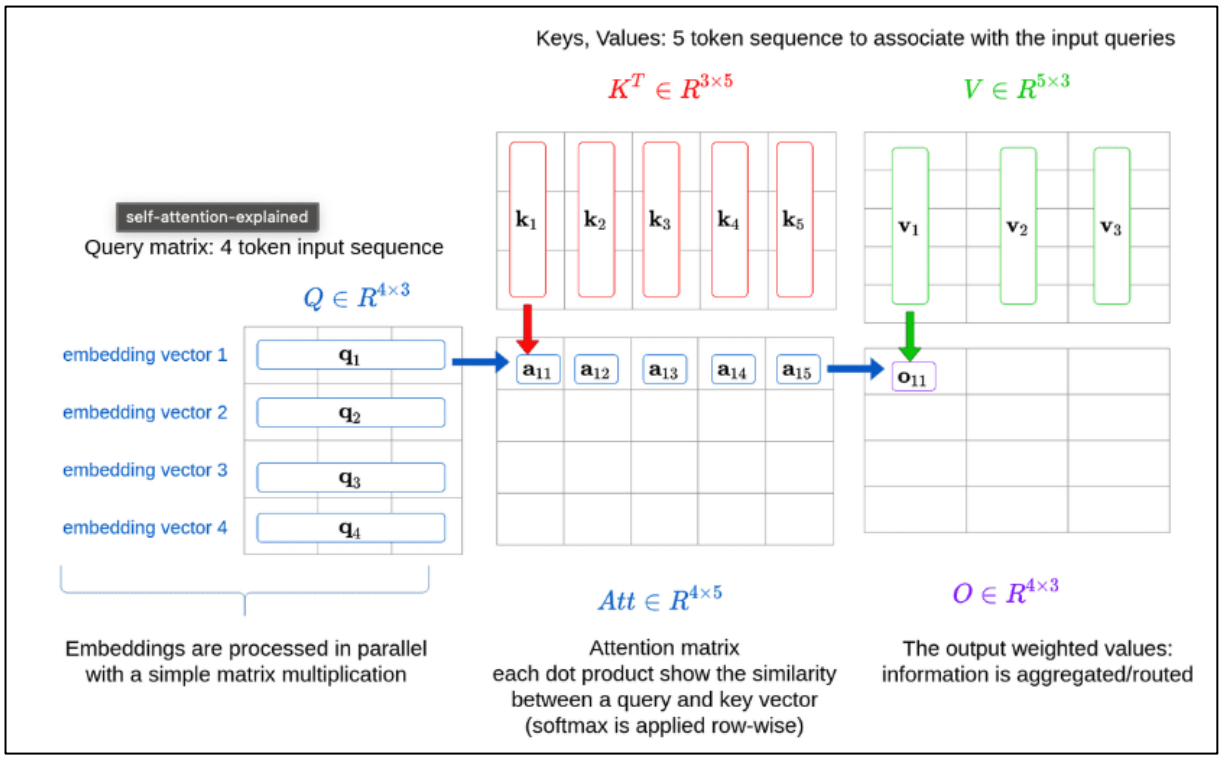


Figure 2.8 – Pictorial view of the self-attention mechanism (Adaloglou, 2021).

Multi-head attention extends the self-attention mechanism by allowing the model to focus on different representation subspaces. It enhances the model's ability to capture various aspects of the relationships between tokens. Given h heads, the multi-head attention is computed as:

1. Compute multiple heads:

$$\text{head}_i = \text{Attention}(\mathbf{Q_i}, \mathbf{K_i}, \mathbf{V_i}), i = 1, \dots, h \tag{2.3}$$

where $\mathbf{Q_i} = \mathbf{X}\mathbf{W_i^Q}$, $\mathbf{K_i} = \mathbf{X}\mathbf{W_i^K}$, $\mathbf{V_i} = \mathbf{X}\mathbf{W_i^V}$.

2. Concatenate and Project:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)\mathbf{W^O} \tag{2.4}$$

where $\mathbf{W^O} \in \mathbb{R}^{hd\times d_k}$is the output projection matrix. Figure 2.9 depicts how, by employing multiple attention heads, the model can capture different types of relationships and patterns in the data. Each head operates on a different subspace, enabling the model to attend to information at various positions and representation levels.

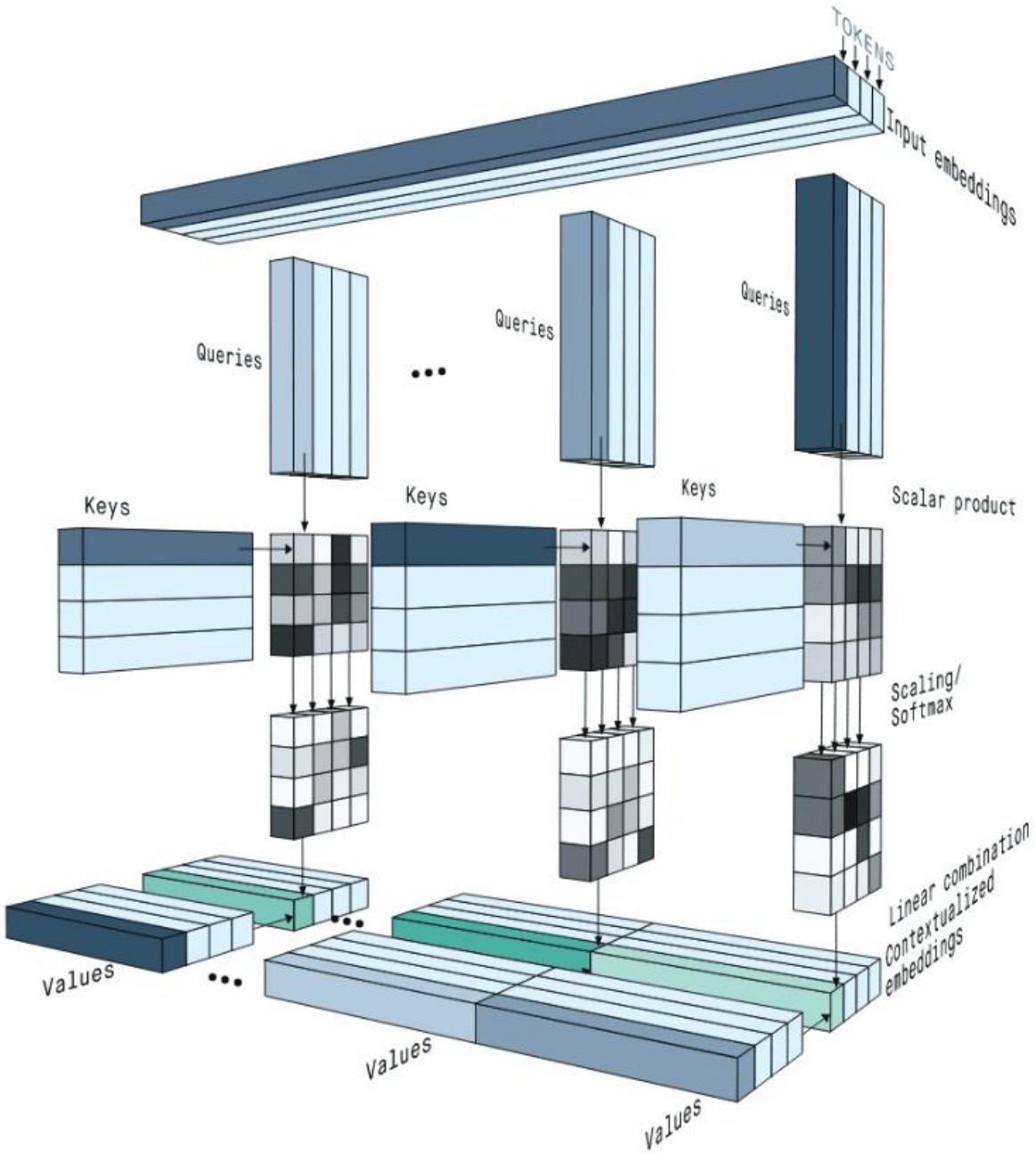


Figure 2.9 – Illustration of the Multi-head attention mechanism (Adaloglou, 2021).

2.7.2 Transformers

Transformers have emerged as a groundbreaking architecture in natural language processing (NLP). They offer a novel method for handling sequential data without relying on traditional recurrent or convolutional neural networks. The Transformer architecture consists of two main parts: the **encoder** and the **decoder**. Both are built by stacking multiple layers that share a common structure but serve different purposes in the overall model (Prince, 2024).

The encoder's role is to read and transform the input sequence into a rich, abstract representation that encapsulates the meaning and context of the data. Each encoder layer consists of:

1. Self-Attention Mechanism: This component enables the model to weigh the importance of different words in the input sequence relative to each other. Doing so captures the relationships and dependencies between words, regardless of their position in the sequence.
2. Feed-Forward Neural Network: A simple neural network is applied to each position individually following the self-attention layer. This network further processes the data, allowing the model to learn complex patterns and representations.

The decoder generates the output sequence, such as a translated sentence or a predicted next word. Each decoder layer includes:

1. Masked Self-Attention Mechanism: Similar to the encoder's self-attention, but with a mask applied to prevent the model from "seeing" future positions. This ensures that predictions for a position depend only on known outputs.
2. Encoder-Decoder Attention: This layer allows the decoder to focus on relevant parts of the input sequence by attending to the encoder's output. It aligns the input and output sequences meaningfully.
3. Feed-forward neural Network: Like the encoder, this network processes the combined information to produce the final output at each position.

Since Transformers do not inherently consider the order of sequence elements, positional encoding is introduced to provide information about the position of each word in the sequence. This is achieved by adding a positional vector to each input embedding, allowing the model to distinguish between words based on their order.

The attention mechanism is central to the Transformer's ability to handle sequences effectively. By calculating attention scores, the model determines which words in the sequence are most relevant to each other. This allows the model to focus on important relationships, such as subject-verb agreements or contextual cues, enhancing its understanding of the input data.

The most remarkable advantages of Transformers are:

1. Parallel Processing: Unlike RNNs, Transformers can simultaneously process all input sequence elements. This parallelism leads to faster training times and more efficient computation.
2. Long-Range Dependency Handling: The attention mechanism enables the model to capture relationships between distant words in a sequence, improving performance on tasks that require understanding context over long passages.
3. Scalability: Transformers are highly scalable and have been used to build very large models that achieve state-of-the-art results across various NLP tasks.

Transformers represent a significant advancement in sequence processing. By leveraging attention mechanisms and eliminating the need for recurrent structures, they offer a more efficient and effective means of modeling sequential data. Their ability to process entire sequences in parallel and capture intricate dependencies has opened new avenues for research and application in NLP and beyond. Understanding the fundamental workings of Transformers is essential for anyone looking to engage with modern developments in machine learning and artificial intelligence. Models like BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) are prominent examples of their successful applications (Prince, 2024).

2.7.3 Fine-Tuning

Fine-tuning is a key technique in the adaptation of LLMs to specific tasks or domains, enhancing their performance beyond the capabilities achieved through pre-training alone. This process involves adjusting the parameters of a pre-trained model using task-specific data, thereby enabling the model to specialize in particular applications while enhancing the extensive knowledge acquired during its initial training phase.

The fine-tuning process commences with a pre-trained LLM, which has been trained on a vast corpus of general data. This pre-training equips the model with a broad understanding of language structures and semantics. Subsequently, fine-tuning is performed using a smaller, task-specific dataset, allowing the model to adapt its parameters to the nuances of the target task. This approach is grounded in the principles of transfer learning, where knowledge from one domain is transferred to improve performance in another (Chen; Chen; Su, 2023).

The OpenAI documentation on fine-tuning is a detailed and user-focused guide that outlines the processes, tools, and best practices necessary to customize pre-trained OpenAI models for specific tasks or domains. Fine-tuning is presented as a powerful methodology for adapting models, such as GPT, to achieve higher performance on tasks requiring specialized knowledge, style alignment, or domain-specific capabilities. The guide is structured into key sections that encompass all stages of the fine-tuning lifecycle (OpenAI, 2024a).

Preparing the Dataset: The first step in fine-tuning involves creating a dataset tailored to the desired task. The guide emphasizes the importance of high-quality, representative data, formatted as JSONL files with "prompt" and "completion" pairs. Each data entry should clearly define the input (prompt) and the expected output (completion) to guide the model's learning process. Best practices include ensuring consistent formatting, avoiding ambiguous prompts, and balancing the dataset to prevent bias in outputs.

Uploading Data and Managing Projects: The documentation provides instructions for securely uploading datasets to the OpenAI platform. It introduces the concept of fine-tuning projects, allowing users to track experiments and configurations. The guide highlights how OpenAI's infrastructure facilitates data storage, versioning, and reuse across multiple fine-tuning iterations.

Executing the Fine-Tuning Process: Fine-tuning is initiated via the OpenAI API. Users are guided through command-line instructions to configure and start fine-tuning jobs. Key parameters, such as the number of epochs and batch size, can be adjusted to optimize performance. The documentation also provides recommendations for leveraging default settings effectively, especially for users less familiar with hyperparameter tuning. More details about the hyperparameters functions and operation are available at Section 2.7.4.

Evaluation and Model Deployment: The guide emphasizes the importance of rigorous evaluation to ensure the fine-tuned model achieves its intended behavior. OpenAI provides metrics for assessing the model's performance, including accuracy and error rates, and suggests testing the model on a validation dataset before deploying it in production. Additionally, users can seamlessly integrate the fine-tuned model into applications by deploying it via the OpenAI API.

Best Practices and Advanced Techniques: Throughout the guide, OpenAI stresses several best practices to maximize fine-tuning success:

1. Data Augmentation: Encouraging the creation of diverse datasets to enhance the robustness of the fine-tuned model.
2. Iteration: Advocating for iterative fine-tuning, where the model is evaluated and refined across multiple stages.
3. Parameter-Efficient Techniques: This section mentions strategies such as prefix-tuning or LoRA (Low-Rank Adaptation) for scenarios with limited computational resources.

The documentation highlights various practical applications for fine-tuning, such as creating domain-specific assistants, the accuracy of technical queries, or aligning model behavior with behavior to a brand's tone. Ethical considerations are also addressed, with OpenAI urging users to scrutinize dataset content for harmful biases and potential misuse.

2.7.4 LLMs Hyperparameters

The meticulous tuning of hyperparameters is fundamental to the successful training and deployment of LLMs. Each hyperparameter plays a distinct role in shaping the learning process, and their optimal configuration often requires empirical experimentation and validation. Understanding the interplay between these hyperparameters enables researchers and practitioners to develop models that are both efficient and effective in their designated tasks.

Epochs: An epoch refers to a complete pass through the entire training dataset during the learning process. The number of epochs determines how many times the model will iterate over the dataset. Selecting an appropriate number of epochs is essential; insufficient epochs may lead to underfitting, where the model fails to capture the underlying data patterns, while excessive epochs can cause overfitting, where the model learns noise and specificities of the training data, impairing its generalization to new data. Empirical studies suggest that the optimal number of epochs varies depending on the dataset and model architecture, necessitating experimentation to identify the most effective setting (Nath *et al*., 2024).

Temperature: In the context of LLMs, temperature is a hyperparameter that controls the randomness of predictions by scaling the logits before applying the softmax function. A lower temperature value results in more deterministic and focused outputs as the model becomes more confident in its predictions. Conversely, higher temperatures introduce greater randomness, allowing for more diverse and creative outputs. Adjusting the temperature is

particularly useful in tasks requiring a balance between creativity and coherence, such as text generation and conversational agents (Trad; Chehab, 2024).

Batch Size: Batch size denotes the number of training examples utilized in one forward and backward pass of the model. It influences the stability and efficiency of the training process. Larger batch sizes can lead to faster training and more stable gradient estimates but may require substantial computational resources. Smaller batch sizes, while more memory-efficient, can result in noisier gradient estimates, potentially necessitating a lower learning rate to maintain training stability. Research indicates that the choice of batch size can significantly affect model performance and training time, underscoring the importance of selecting an appropriate batch size based on the specific application and available resources (Aldin; Aldin, 2022) .

Learning Rate: The learning rate determines the step size at each iteration while moving toward a minimum of the loss function. It is a critical hyperparameter that affects the speed and convergence of the training process. A learning rate that is too high can cause the model to converge too quickly to a suboptimal solution, while a learning rate that is too low can result in a prolonged training process that may get stuck in local minima. Adaptive learning rate methods and learning rate schedules are often employed to adjust the learning rate dynamically during training, enhancing convergence and performance (Zhao; Gao; Fang, 2024).

2.7.5 Evolution of LLMs: Multimodal, Reasoning-Enhanced, and Beyond

In recent years, LLMs have undergone significant advancements, leading to the development of various specialized architectures. Among these, multimodal models and reasoning-enhanced models have garnered particular attention. This Section provides an overview of the primary types of LLMs, with a focus on multimodal models, reasoning-based models such as GPT-4o, DeepSeek-V3, GPT-o1 and DeepSeek-r1, and other notable architectures that have contributed to the evolution of LLMs.

Multimodal LLMs are designed to process and generate multiple forms of data, such as text, images, and audio, enabling a more comprehensive understanding and generation of content. These models integrate various data modalities to enhance their performance across diverse tasks.

A prominent example is OpenAI's GPT-4o, introduced in May 2024. GPT-4o is capable of analyzing and generating text, images, and sound, making it a versatile tool for

applications requiring a combination of these modalities. It has demonstrated state-of-the-art results in voice, multilingual, and vision benchmarks, setting new records in audio speech recognition and translation. Notably, GPT-4o scored 88.7% on the Massive Multitask Language Understanding (MMLU) benchmark, surpassing its predecessors (OpenAI, 2024c).

The integration of multiple data types allows multimodal LLMs to perform tasks that were previously challenging for text-only models, such as image captioning, audio transcription, and cross-modal content generation. This capability opens new avenues for research and application in fields like healthcare, education, and entertainment.

Reasoning-enhanced LLMs are designed to improve the model's ability to perform complex reasoning tasks by incorporating mechanisms that allow for step-by-step problem-solving. This approach enables the models to handle tasks that require logical deduction, mathematical calculations, and multi-step reasoning processes. OpenAI's GPT-o1, released in December 2024, represents a significant advancement in reasoning capabilities. Unlike its predecessors, GPT-o1 employs a "think, then answer" strategy, allowing it to deliberate before providing responses. This method enhances its performance in complex tasks, including competitive programming, mathematics, and scientific reasoning. GPT-o1 has demonstrated proficiency comparable to Ph.D. students on benchmarks in physics, biology, and chemistry (OpenAI).

The model's ability to "think" before responding is achieved through a process known as chain-of-thought prompting, where the model generates intermediate reasoning steps before arriving at a final answer. This technique has been shown to improve performance on tasks that require multi-step reasoning by allowing the model to break down complex problems into manageable parts.

DeepSeek-r1, introduced by the Chinese AI startup DeepSeek in January 2025, is another notable model in the realm of reasoning-enhanced LLMs. DeepSeek-r1 was developed using pure reinforcement learning (RL) without supervised fine-tuning, focusing on enabling the model to develop reasoning capabilities through self-evolution. The training process involved the use of Group Relative Policy Optimization (GRPO) as the RL framework to improve performance in reasoning tasks (Deepseek-Ai *et al*., 2025).

The model demonstrated remarkable reasoning capabilities, naturally emerging with numerous powerful and intriguing reasoning behaviors. However, it encountered challenges such as poor readability and language mixing. To address these issues and further enhance

reasoning performance, DeepSeek introduced DeepSeek-r1, which incorporates multi-stage training and cold-start data before RL. DeepSeek-r1 achieved performance comparable to OpenAI's GPT-o1 on reasoning tasks. The development of DeepSeek-r1 highlights the potential of reinforcement learning as a method for enhancing reasoning capabilities in LLMs, offering an alternative to traditional supervised learning approaches.

Beyond multimodal and reasoning-enhanced models, several other LLM architectures have made significant contributions to the field. Mixture-of-Experts (MoE) models, such as DeepSeek-V3, utilize a large number of specialized sub-models (experts) and dynamically select a subset of them for each input. DeepSeek-V3, for instance, is a strong MoE language model with 671 billion total parameters, with 37 billion activated for each token. This architecture allows for efficient inference and cost-effective training by leveraging Multi-head Latent Attention and DeepSeek-MoE architectures (Deepseek-Ai *et al*., 2024).

The MoE approach enables the model to allocate computational resources more effectively, focusing on the most relevant experts for a given task. This results in improved performance across a range of tasks while maintaining computational efficiency.

The open-source community has also contributed significantly to the advancement of LLMs. Models like Meta's Llama and DeepSeek's various releases have provided accessible alternatives to proprietary models, fostering innovation and collaboration within the research community. The success of open-source models demonstrates the potential for community-driven development to achieve performance comparable to, or even surpassing, that of proprietary models.

2.7.6 Tree of Thoughts

The Tree of Thoughts (ToT) technique is an evolution of Chain of Thought (CoT)-based reasoning, allowing an LLM to explore different paths before making a final decision. This improves the quality of answers, making them more informed and accurate, especially in complex problems.

The central idea of ToT is to structure the thought process as a decision tree, where each node represents a partial state of reasoning and branches represent different possible directions. Thus, instead of following a single linear path, as in CoT, the model can explore several alternatives and evaluate which one is most promising.

The process follows three main steps:

1. Dividing the problem into intermediate steps (tree nodes)
2. Instead of trying to solve a problem all at once, the model divides the issue into small interconnected parts, creating a set of possible thought states.
3. Expanding multiple thoughts for each step

For each node in the tree, the model generates different approaches or possible solutions. Each of these approaches can lead to different answers in the end. Table 2.2 presents the main characteristics of this Prompt Engineering technique as well as its differences in relation to CoT.

Table 2.2 – Main differences between CoT and ToT.

Characteristic	**Chain of Thought (CoT)**	**Tree of Thoughts (ToT)**
Type of reasoning	Sequential and linear	Exploratory and branching
Approach	Single step	Multiple approaches in parallel
Efficiency	May fail if one step is incorrect	More robust as it explores alternatives
Ideal application	Straightforward and well-defined problems	Complex problems with multiple solutions

3 Literature Review

The exploration of relevant literature forms the backbone of any rigorous academic research. It provides a foundational understanding of the field, identifies gaps, and situates the current study within the broader context of existing knowledge. This chapter outlines the methodological approach employed to conduct the literature review, emphasizing the combination of traditional and innovative techniques to ensure a comprehensive examination of the research landscape.

The review process incorporated conventional methods of bibliographic inquiry, leveraging renowned scientific databases such as Google Scholar, MDPI, IEEE Xplore, ScienceDirect, AIAA ARC, and others. These databases served as primary repositories for identifying peer-reviewed publications, conference proceedings, and high-impact papers central to the research themes.

In addition to these traditional methods, the innovative application "Connected Papers" was utilized to enhance the discovery of relevant works. Connected Papers is a visual and algorithmically driven tool designed to help researchers uncover and explore publications related to their field of study. Unlike citation trees, Connected Papers constructs a Force Directed Graph by analyzing approximately 50,000 papers and identifying those with the strongest conceptual and bibliographic connections to a chosen origin paper. Its similarity metric, based on co-citation and bibliographic coupling, clusters closely related papers visually, facilitating the identification of works that may not directly cite one another but share significant conceptual overlap.

This integration of traditional bibliographic searches and Connected Papers provided a multifaceted view of the literature, enabling the identification of seminal works and emerging studies that align with the focus of this research. By adopting this dual approach, the literature review ensures a robust and nuanced understanding of the field, setting the stage for this dissertation's contributions.

3.1 Use of Artificial Intelligence in the Aerospace Field

Artificial Intelligence (AI) has become a pivotal force in advancing aerospace engineering. It offers innovative solutions across various domains, including design, manufacturing, operations, and maintenance. The integration of AI methodologies has led to

significant improvements in efficiency, safety, and performance within the aerospace sector.

AI techniques, such as machine learning and optimization algorithms, have been employed in aircraft design to enhance aerodynamic performance and structural integrity. For instance, AI-driven design tools facilitate the exploration of complex design spaces, enabling engineers to identify optimal configurations that meet stringent performance criteria. This approach accelerates the design process and results in more efficient and innovative aircraft structures (Qiu; Zhao; Wang, 2023; Zou; Sun, 2021).

AI applications have also benefited aerospace manufacturing processes. Implementing AI in predictive maintenance allows for real-time monitoring of equipment health, leading to the early detection of potential failures and the optimization of maintenance schedules. This predictive capability reduces downtime and maintenance costs while enhancing the reliability of aerospace components (Insaurralde, 2020).

Furthermore, AI has been applied to enhance cybersecurity measures within aerospace systems. Machine learning models are utilized to detect and mitigate cyber threats, ensuring the integrity and security of critical aerospace infrastructure. The ability of AI to analyze vast amounts of data and identify anomalous patterns is crucial in safeguarding against sophisticated cyber-attacks (Garcia; Babiceanu; Seker, 2021).

Despite these advancements, the integration of AI in aerospace presents challenges, particularly concerning the explainability and transparency of AI models. The development of Explainable AI (XAI) aims to address these concerns by providing insights into the decision-making processes of AI systems, thereby enhancing trust and facilitating the certification of AI-driven applications in safety-critical aerospace environments (Sutthithatip *et al*., 2021).

These studies indicate that integrating artificial intelligence into aerospace engineering is a trend with no turning back. However, a question remains: what is the state-of-the-art use of artificial intelligence in requirements engineering, systems verification, and safety analysis in aerospace systems? The following sections of this chapter attempt to elucidate these points.

3.2 AI in Requirements Engineering

Requirements Engineering (RE) is a foundational discipline in software development, critical to ensuring the alignment between stakeholder needs and system functionalities. The increasing complexity of software systems has driven a surge in the adoption of Artificial Intelligence (AI) to streamline and enhance RE activities. This subsection synthesizes the

findings of two extensive systematic literature reviews: “Advances in Automated Support for Requirements Engineering: A Systematic Literature Review” (Umar; Lano, 2024) and “A Systematic Literature Review on Using Natural Language Processing in Software Requirements Engineering” (Necula; Dumitriu; Greavu-Șerban, 2024). Both offer comprehensive insights into the use of AI in RE.

Both reviews share a common goal: to survey the landscape of automated support for RE. However, they differ in their scope and focus. Umar *et al*. (2024) delve into the specifics of automated RE support tools, their outputs, targeted RE phases, and evaluation methodologies, providing insights into the automation of various RE phases, including elicitation, a critical aspect of this doctoral project. Conversely, Necula, Dumitriu and Greavu-Șerban (2024) offer a broader view of the integration of Natural Language Processing (NLP) and AI in RE. Their analysis covers a wider array of AI technologies, including machine learning, deep learning, and large language models, aligning with my research's utilization of prompt engineering and LLMs. They emphasize automation as a means to address traditional challenges, such as ambiguity, inconsistency, and inefficiency in requirements elicitation, analysis, and validation. Furthermore, both articles highlight AI's transformative potential to reduce manual effort and improve the precision of requirements engineering tasks.

The articles rely on overlapping foundational works, particularly those involving NLP and AI tools designed to support tasks such as requirements classification, ambiguity detection, and UML model generation. Both reviews underscore the predominance of NLP in automating requirements elicitation and validation. Techniques like Part-of-Speech tagging, syntactic parsing, and machine learning models are frequently cited as enablers of this automation. However, their specific selections of primary studies differ, with each review focusing on particular aspects of AI in RE.

While the thematic overlap between both works is substantial, the articles diverge in scope and methodological focus. Umar and Lano (2024) adopt a systematic literature review (SLR) to evaluate 85 studies, emphasizing tools' outputs, the phases of automated RE, and their evaluation methods. Necula, Dumitriu and Greavu-Șerban (2024) provide a broader analysis adopting PRISMA guidelines and using bibliometric tools like thematic mapping and co-citation analysis, exploring trends and thematic evolutions over three decades.

Both studies offer robust foundations for understanding the state of the art in AI-enabled RE. However, limitations persist. Umar and Lano emphasize the insufficient

Industrial Integration: the lack of widespread adoption, a challenge also relevant to the aerospace domain. Necula, Dumitriu and Greavu-Șerban (2024) critique the narrow scope of many existing tools, which often target specific RE phases without holistic integration.

Of the 85 studies reviewed by Umar and Lano (2024) and the 309 studies reviewed by Necula, Dumitriu and Greavu-Șerban (2024), none utilize STPA or its combination with LLMs. However, outside of these two systematic literature reviews on the use of AI in RE, some studies present results from using LLMs to apply the STPA technique without addressing their efforts to elicit system requirements. Nevertheless, let's delve into detail on these works in the next Section.

3.3 STPA With LLMs

The search for papers that addressed using LLMs for automating STPA through the main scientific databases and the Connected Papers application found 465 articles. However, any results obtained in papers from 2023 onwards can already be considered obsolete since they considered, at best, the GPT-3 benchmarking. Since the performance of GPT-3 is much lower than that of GPT-4 and other competing LLMs at the same performance level (e.g., Claude 3.5, Gemini 1.5 Pro, Llama 3.2, etc.), it is prudent to discard the results obtained before 2023. Considering this cut, 43 published papers remained to be analyzed. Of these, only one uses LLMs to actually assist in conducting an STPA.

The study titled "Hazard Analysis in the Era of AI: Assessing the Usefulness of ChatGPT-4 in STPA Hazard Analysis" explores the integration of Large Language Models (LLMs), specifically ChatGPT-4, into the Systems Theoretic Process Analysis (STPA) framework for hazard analysis in socio-technical systems (Charalampidou; Zeleskidis; Dokas, 2024). The research focuses on evaluating the efficacy of ChatGPT-4 in replicating and augmenting the STPA process, using the ROLFER (Robotic Lifeguard For Emergency Rescue) UAV system as a case study. The paper’s objectives are:

1. Evaluate ChatGPT-4’s contribution to Steps 3 and 4 of the STPA process, which involve Unsafe Control Actions (UCAs) and loss scenarios generation.
2. Compare outputs from ChatGPT-4 against those of a human safety analysis team.
3. Assess the time efficiency and practicality of using LLMs in hazard analysis.

Regarding the authors' methodology, the system under analysis (ROLFER) was comprehensively described to ChatGPT-4, enabling the model to understand operational mechanisms. Then, A step-by-step STPA was performed with ChatGPT-4, focusing on:

1. UCAs generation for 18 control actions.
2. Loss scenarios identification based on UCAs.
3. Safety specifications development to mitigate identified hazards.

The results were benchmarked against a verified human STPA analysis conducted over several weeks. Their key findings were:

1. UCAs Generation: ChatGPT-4 produced 138 UCAs for 18 control actions, and half of them were deemed invalid, misplaced, or contextually incorrect. Despite these errors, ChatGPT-4 identified UCAs overlooked by human analysts, leading to additional safety specifications. Generated UCAs often exhibited patterns, indicating potential biases in response generation.
2. Loss Scenarios: ChatGPT-4 efficiently generated structured loss scenarios but occasionally introduced unfounded assumptions about system functionality, reducing the validity of specific scenarios.
3. Safety Specifications: Specifications were directly linked to loss scenarios, but their validity was contingent on the accuracy of the preceding analyses.
4. System Understanding: ChatGPT-4 effectively posed 35 system-related questions, significantly aiding system comprehension and suggesting innovative improvements to the system, some of which aligned with best practices.
5. Time Efficiency: ChatGPT-4 drastically reduced the time required for hazard analysis, completing tasks in less than 8 hours compared to the 4-5 weeks needed for the human team.

The study demonstrates a pioneering application of LLMs in safety-critical domains. One of its notable strengths lies in its time efficiency, as it highlights how LLMs can significantly expedite hazard analysis processes that would otherwise require extensive manual effort. Additionally, ChatGPT-4 proved to be a valuable tool in enhancing system comprehension, with its questions facilitating deeper understanding and enriching brainstorming sessions. Furthermore, the study underscores ChatGPT-4's ability to uncover overlooked Unsafe Control Actions (UCAs), leading to the generation of new safety measures

and demonstrating its capacity to contribute meaningfully to system safety analysis.

On the other hand, the study revealed several weaknesses and limitations in ChatGPT-4's application to hazard analysis. A significant issue was the inconsistent validity of its outputs, with a high error rate in UCA generation, where 50% of the results were invalid or misplaced, and loss scenarios often relied on flawed assumptions, compromising the quality of safety specifications. Additionally, ChatGPT-4 exhibited a notable pattern bias, producing repetitive answer structures that reduced the diversity and richness of its outputs. Its lack of independence further limited its utility, as it could not validate results without human intervention, undermining its standalone applicability. Over time, the model's performance deteriorated, with the quality of responses declining in extended interactions, likely due to context limitations. Lastly, the model displayed an over-reliance on prompts, closely mirroring the wording provided, which raised concerns about its creativity and adaptability in generating nuanced or original content.

Finally, they indicate that future research should explore promising directions to enhance the use of LLMs in hazard analysis. A comparative analysis of alternative models, such as Google Bard (now upgraded to Google Gemini), could provide insights into their relative strengths and weaknesses, broadening the applicability of LLMs in safety-critical tasks. Refining prompt strategies is another critical area, as it offers the potential to reduce biases and improve the diversity and quality of generated responses. Expanding the application of LLM-assisted STPA to other safety-critical systems would further validate the methodology and demonstrate its versatility across different domains. Additionally, the development of hybrid frameworks that integrate LLMs with expert oversight could enhance reliability by combining automated efficiency with human expertise, creating a more robust approach to safety analysis.

3.3.1 Comparison Between the State-of-the-art and this Research

The work of Charalampidou *et al*. (2024) shares a core similarity in their objectives with this research, as it leverages ChatGPT-4 to automate aspects of the STPA process. Their study focuses primarily on identifying UCAs, loss scenarios, and safety specifications, and this thesis extends this foundation by also eliciting system-level requirements that go beyond hazard analysis, providing a more holistic contribution to system engineering. Both studies recognize the time efficiency offered by ChatGPT-4, and they validate, although in different ways, the outputs generated by ChatGPT-4 through comparison with human expertise.

Charalampidou *et al*. (2024) evaluate the UCAs and loss scenarios produced by ChatGPT-4 against those derived from a verified human STPA analysis, while my research compares the elicited requirements to those from a real system currently operational within the FAB, adding a layer of practical relevance to the assessment. Generating requirements adds an additional dimension to the time-saving potential, addressing broader system design needs.

Nevertheless, their article observed repetitive patterns in ChatGPT-4's responses (e.g., fixed numbers of UCAs per category), which were useful for developing prompts that disrupt these patterns, improving diversity in requirements elicitation and other outcomes explored in this research. The flaws they identified (e.g., hallucinations, and incorrect assumptions) helped me incorporate validation layers within the prompts.

3.4 LLMs in Systems Verification

The search for articles relating to LLMs and Systems Engineering, more specifically the Systems Verification phase, also yielded few results. Once again, results prior to 2024 were purposely neglected, given the accelerated rate of improvement of LLMs and the poor performance of models from 2023 and earlier when compared to recently released models.

Among the articles found, only two deserve to be brought to light in the discussions of this doctoral dissertation. One of them, "*Hardware Design and Verification with Large Language Models: A Literature Survey, Challenges, and Open Issues*", by Abdollahi *et al*. (2024), is a comprehensive literature review on the topic, which analyzed 54 research papers to assess the current role of LLMs in enhancing automation, optimization, and innovation within hardware design and verification workflows. Despite being a preprint, this article is very well structured and well-founded, in addition to being one of the most recent sources on the subject, having been made available on the platform *Preprint.org* on November 4th, 2024.

The second article that received attention from this research was written by Boeing researchers and has a reasonable intersection of objectives with this doctoral thesis. Depauw *et al*. (2024) work, "*Development of a Commercial Airplane Certification AI Digital Assistant*", introduces a machine learning-based digital airworthiness Certification compliance assistant tool "CertifAIer" designed to enhance the aircraft Certification process by leveraging LLMs capabilities. Their work, although only briefly touching on the topic of MoCs, does not provide any insight into how LLMs could automate the process of assigning them to system requirements. However, several insights proposed by Depauw *et al*. (2024) helped in planning

for the achievement of the goals of this doctoral thesis.

The next two subsections delve into detail about these two publications that influenced this work.

3.4.1 Literature Survey on Hardware Design and Verification With LLMs

The article "*Hardware Design and Verification with Large Language Models: A Literature Survey, Challenges, and Open Issues*" (Abdollahi *et al*., 2024) extensively reviews the integration of LLMs in hardware design and verification, identifying automation, optimization, and methodological advancements as key themes.

The literature on LLMs for hardware design and verification can be organized thematically as follows:

1. HDL Generation: This theme focuses on using LLMs to generate Verilog or VHDL code from high-level specifications.
2. Design Optimization: This theme explores the use of LLMs for optimizing hardware designs for PPA.
3. Verification: This theme includes tasks such as generating test benches, creating test cases, and identifying bugs.
4. Challenges: This theme addresses the challenges associated with applying LLMs for hardware design and verification.
5. Open Issues: This theme identifies open issues and areas for future research.

The reviewed studies demonstrate the versatility of LLMs in automating repetitive yet complex hardware design and verification tasks, such as generating Hardware Description Language (HDL) code and optimizing designs. The survey emphasizes how LLMs expedite verification workflows and enhance reliability by minimizing human errors, aligning with the broader aim of reducing manual oversight in engineering disciplines. Studies such as AutoChip and GPT4AIGChip highlight iterative refinement and automated synthesis, showcasing robust methodologies adaptable to other domains, including aerospace.

Nevertheless, the need for specialized training datasets to enhance model performance in niche applications, such as MoC attribution, remains a recurring challenge that has been underexplored by existing research. Current studies focus heavily on hardware-centric benchmarks, leaving systemic evaluation frameworks for broader compliance tasks under-

addressed. Additionally, few works address the interpretability of LLM outputs or the mechanisms to validate compliance with domain-specific regulations.

Several studies have explored the use of LLMs for automating and optimizing different aspects of hardware design. These include generating Hardware Description Language (HDL) code, optimizing design parameters, and automating verification tasks. For instance, LLMs have been used to generate Verilog or VHDL code from high-level specifications, reducing the manual effort involved in the design process. Moreover, LLMs have shown potential in optimizing designs for power, performance, and area (PPA) by exploring different design configurations and suggesting optimal solutions.

In hardware verification, LLMs have been used to generate test benches, create test cases, and identify potential bugs. This automation can significantly reduce the time and resources required for verification, leading to faster iterations and more reliable hardware products.

Despite promising advancements, challenges remain in applying LLMs to hardware design and verification. These include data scarcity, the need for specialized training, and integration with existing design tools. Hardware design complexity requires fine-tuning LLMs for specific tasks, which can be challenging due to the limited availability of domain-specific data. This need for fine-tuned LLMs for specific tasks highlights the relevant and timely nature of this doctoral research.

The surveyed works do not sufficiently address the nuances of regulatory alignment, which is critical for aerospace certification. While hardware design emphasizes circuit and component-level design, the unique complexities of aerospace systems, particularly in MoC attribution, remain underrepresented. Each methodology explored has its advantages and limitations. Fine-tuning effectively adapts LLMs for specific tasks but requires large, high-quality datasets. Developing new architectures can lead to better performance but requires significant expertise and resources. Integration with existing tools is crucial for practical application but can be challenging due to compatibility issues.

The integration of LLMs with Electronic Design Automation (EDA) tools reflects a trend toward collaborative AI-human workflows, promising parallels in collaborative compliance assessments. Efforts to develop domain-specific LLMs, such as Hardware Phi-1.5B, underscore the growing need for specialized models, from which aerospace engineering could benefit. The ethical use of LLMs in critical decision-making and their reliability in

high-stakes applications, such as defense systems, remain pivotal debates influencing their adoption.

The survey underscores LLMs' transformative potential in automating and optimizing hardware design and verification. However, the main studies in this area do not cover, or do so only partially, the use of LLMs in certifying aerospace systems. To date, only one published study on this topic has been published, and the next subsection will explore it.

3.4.2 Certification AI Digital Assistant

The article by DePauw *et al*. (2024) provides a comprehensive overview of the development and application of CertifAIer, a digital assistant leveraging large language models (LLMs) and generative AI to enhance the airworthiness certification process of commercial airplanes. While the research introduces several innovative approaches to streamlining compliance workflows, a detailed examination reveals its contributions, limitations, and potential gaps, particularly in light of the objectives of this doctoral research.

DePauw *et al*. (2024) highlight that the aerospace industry's digital transformation is driving the adoption of machine learning and generative AI, particularly large language models, to enhance certification processes by automating tasks and enabling real-time access to relevant information, improving efficiency and robustness. Figure 3.1 presents a schematic view of this trend.

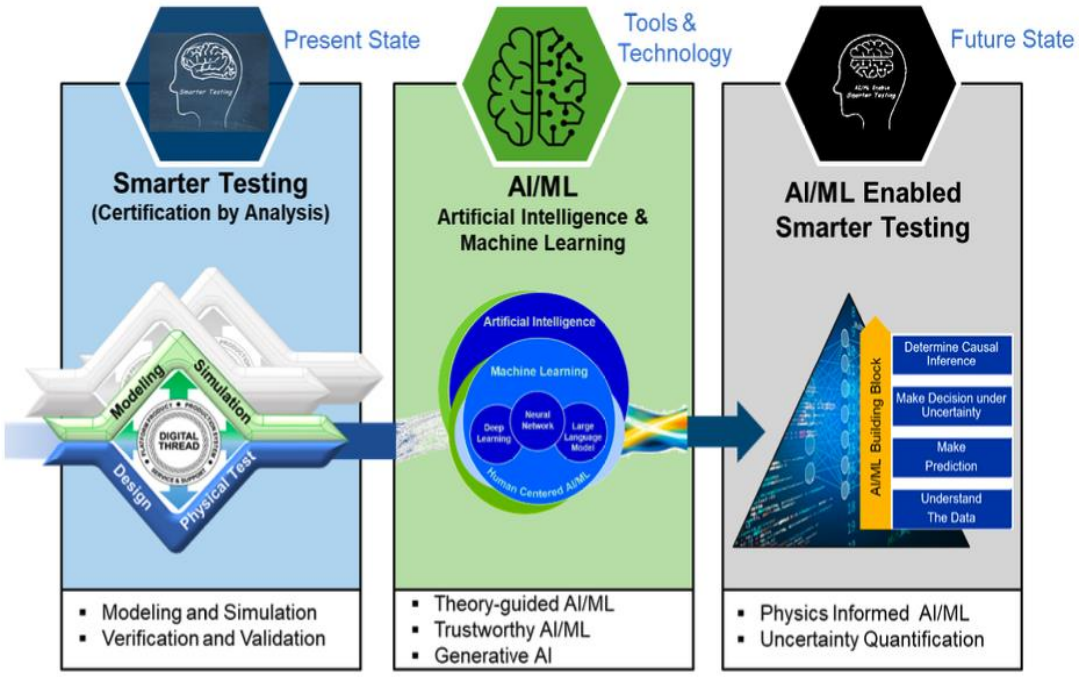


Figure 3.1 – Digital Boeing’s Framework (DePauw *et al*., 2024).

The authors extensively evaluated current technologies and tools to determine baseline capabilities and requirements. This process sought to pinpoint opportunities for improvement and incorporate machine learning techniques to enhance the tool's performance. Figure 3.2 shows the domain of this search plotted in the Vee diagram.

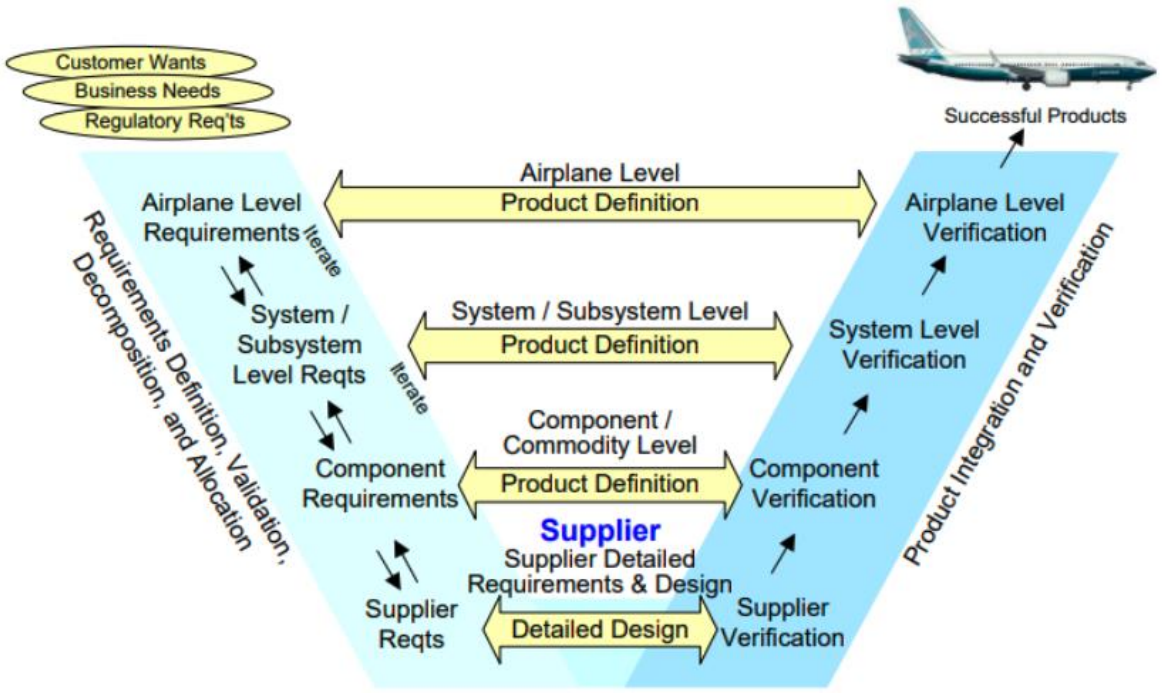


Figure 3.2 – Typical airplane development process follows system engineering Vee model (DePauw *et al*., 2024).

Their tool’s functionality centers on extracting documentation in various formats, converting it into a neutral format compatible with the LLM system, and training the model using both structured and unique content. Predefined document structures guide the training process, with human expertise addressing unique content to refine the model's development.

CertifAIer is deployed via an access-controlled web-based portal, enabling real-time interaction where users can submit natural language queries and receive detailed, accurate responses based on training data from regulations and compliance reports, like a kind of personalized ChatGPT. It features a feedback mechanism to refine future updates, a compliance report template generator tailored to specific airplane components, and embedded links to source materials for enhanced understanding. Designed for continuous updates, the tool incorporates feedback from users and regulators to remain current with evolving requirements. Access is facilitated through both a web interface for direct use and an API for seamless integration with other systems. Figure 3.3 depicts an illustration of the tool’s process and architecture.

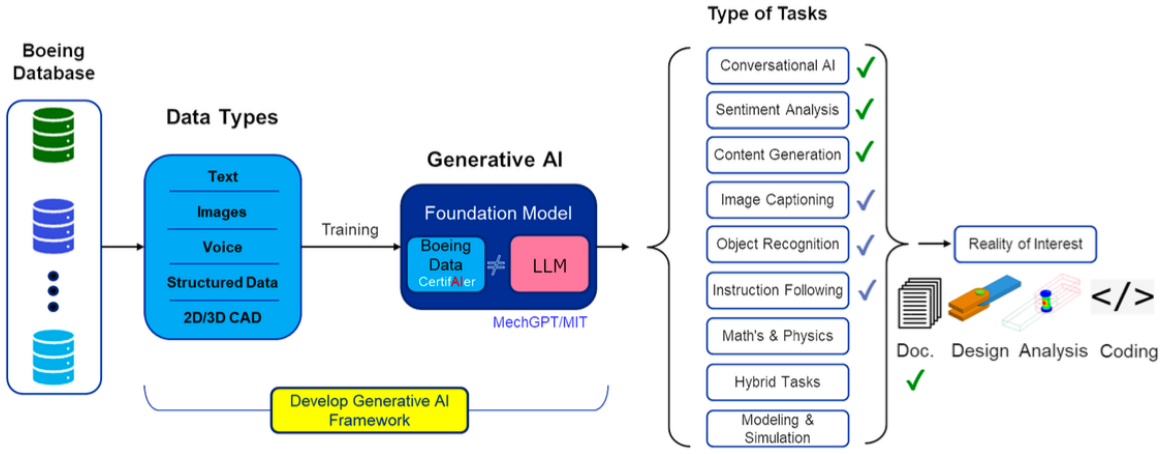


Figure 3.3 – Typical Generative AI Process and Architecture (DePauw *et al*., 2024).

DePauw *et al*. (2024) emphasize the importance of CertifAIer in modernizing the certification process through automation and digital transformation. Key strengths include:

1. Efficiency Gains: The tool demonstrates significant potential in automating document generation, real-time queries, and compliance verification, reducing the time-intensive nature of traditional certification processes.
2. Scalability: The integration of feedback loops and continuous updates ensures that the tool adapts to evolving regulatory frameworks, offering long-term viability.
3. Generative AI in Certification: The use of generative AI to create compliance report templates tailored to specific airplane components represents a novel application, fostering precision and consistency.
4. Stakeholder Engagement: The development process effectively incorporates inputs from manufacturers, regulators, and developers, ensuring alignment with industry needs.

Despite its innovative framework, the study reveals some limitations that restrict its applicability and breadth:

1. Focus on Document Management: While CertifAIer excels in automating compliance documents, it does not delve into the elicitation or validation of system requirements, a crucial step in safety-critical domains.
2. Generalization Across Domains: The reliance on historical compliance reports stored in legacy databases may limit the tool's scalability beyond its initial use cases, especially in defense or rapidly evolving technological fields.

3. Data Accessibility and Security: Challenges related to accessing antiquated databases and ensuring data confidentiality are acknowledged but remain unresolved, particularly in contexts requiring high-security protocols.
4. Absence of Safety-Centric Analysis: The tool does not integrate safety analysis methods such as STPA, which are essential for proactively identifying and mitigating system hazards.

The article's focus on automating compliance workflows through LLMs aligns with one of this doctoral research's key objectives – automating the assignment of Means of Compliance (MoCs). However, significant differences and gaps emerge when compared to this research:

1. Elicitation of Requirements: Unlike CertifAIer, which centers on compliance document management, this doctoral research advances the automation of requirements elicitation from STPA analyses. In contrast, CertifAIer has yet to explore the integration of safety analysis within AI-driven frameworks.
2. Fine-Tuning for MoCs: While CertifAIer relies on pre-trained LLMs for document generation, this research introduces fine-tuning techniques to improve the precision of MoC assignments, bridging a critical gap in automated certification processes.
3. Scope of Application: CertifAIer is tailored for commercial aviation, whereas this research expands the methodology to defense systems, addressing distinct compliance and safety challenges in military aerospace contexts.

The article underscores the growing adoption of LLMs and generative AI in regulatory compliance, highlighting trends such as:

1. Real-Time Compliance Tools: The development of interactive, query-based systems points to a shift toward agile certification models.
2. Adaptation to Regulatory Evolution: Continuous updates to LLMs ensure alignment with dynamic regulatory landscapes, an area where this research contributes by integrating feedback-driven improvement mechanisms.
3. Cross-Domain Applications: The potential expansion of LLM-based tools into aerospace and other safety-critical domains remains a prominent area of debate.

The state of the art presented in DePauw *et al*. (2024) lays the groundwork for exploring AI-driven certification tools but remains focused on document management and compliance efficiency. This doctoral research advances the field by addressing the automation of safety-critical requirements and compliance mappings, paving the way for a more comprehensive, safety-integrated approach to aerospace certification. By bridging these gaps, the research not only enhances the robustness of certification processes but also extends their applicability to high-stakes domains like defense, aligning with the broader objectives of aerospace safety and innovation.

4 Methodology

Building on the foundations established in the literature review, this chapter presents the methodological approach developed to address the three principal research objectives of this doctoral work: (1) the use of Large Language Models (LLMs) to automate the System-Theoretic Process Analysis (STPA) for eliciting system requirements in aerospace defense, (2) the fine-tuning of LLMs to automate the assignment of Means of Compliance (MoCs) to these requirements, and (3) the use of LLMs to generate safety reports. Figure 4.1 depicts a holistic overview of this work's methodology.

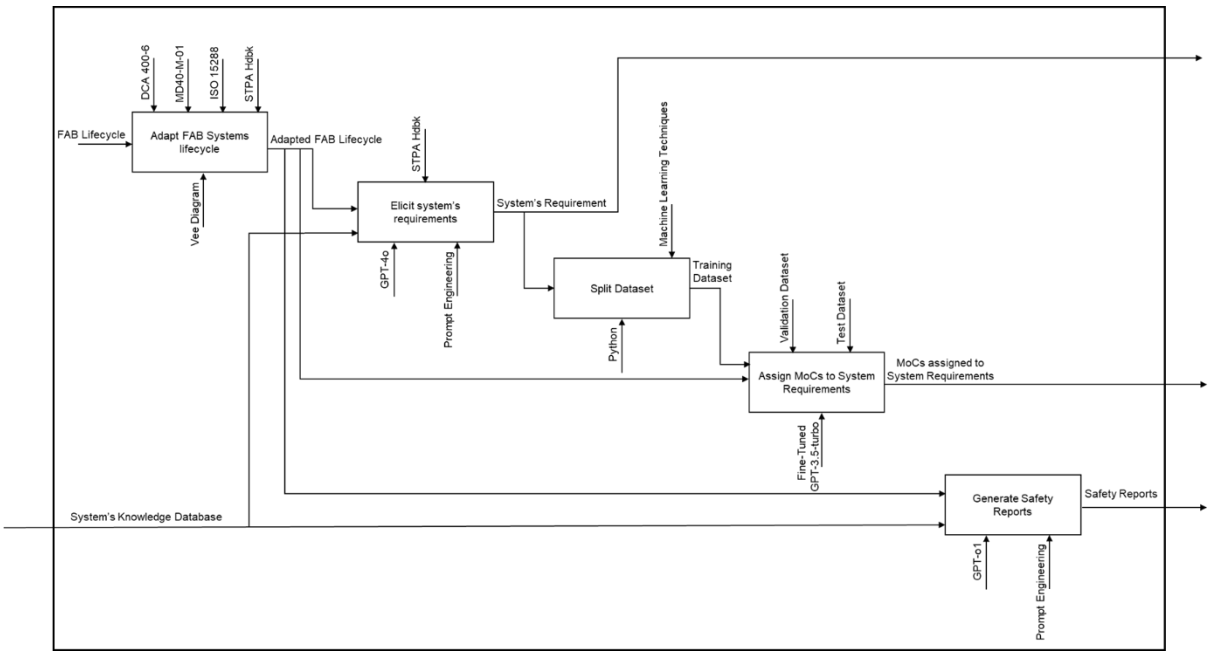


Figure 4.1 – IDEF0 diagram of this work methodology.

As detailed in the preceding Sections, recent studies have laid a promising groundwork for the integration of AI in STPA and compliance processes. Charalampidou *et al*. (2024) demonstrated how ChatGPT-4 could automate certain aspects of STPA, such as identifying Unsafe Control Actions (UCAs) and loss scenarios. However, their scope was confined to hazard identification and safety specifications. This research extends that work by utilizing the same generative AI paradigm to elicit system-level requirements—a more expansive application that addresses not only hazards but also broader design and operational needs.

Similarly, the work of DePauw *et al*. (2024) provides an essential precursor to exploring AI applications in aerospace certification, focusing on document management and

the streamlining of compliance-related workflows. However, this prior work largely neglects the automation of core safety-critical processes, such as the mapping of requirements to MoCs. Addressing this omission, the methodology in this thesis emphasizes a comprehensive approach: employing LLM fine-tuning to systematically assign MoCs to system requirements with a level of accuracy comparable to, or exceeding, that of domain experts. This contribution aligns with the broader industry goal of enhancing certification rigor while optimizing efficiency, particularly in high-stakes domains like aerospace defense.

By synthesizing these advancements, the methodology proposed here not only seeks to reduce the manual effort traditionally associated with STPA and compliance processes but also aims to enhance their consistency and scalability. The integration of LLMs into these workflows represents a significant shift in the way safety-critical systems are designed and certified, aligning with contemporary demands for agility and precision in defense and aerospace engineering.

In the Sections that follow, this chapter will detail the procedural frameworks, data sources, and validation techniques employed to achieve these objectives. Through a combination of experimental rigor and practical relevance, the proposed methodology aims to demonstrate the transformative potential of LLMs in safety-critical systems engineering.

4.1 DCA 400-6 Update With STPA

At the 2024 SIGE (Symposium on Operational Applications in Defense Areas) hosted by ITA, I presented a paper titled "*Enhancing Brazilian Aerospace Systems Lifecycle Directive*." In this work, I propose an update to the Brazilian Air Force Directive DCA 400-6, integrating principles from the Brazilian Ministry of Defense's Manual MD40-M-01 (Brasil, 2020), and aligning with the ISO/IEC/IEEE 15288:2023 standard (ISO, 2023). Both documents deal with Systems Lifecycle in a Systems Engineering framework. By also incorporating the STPA technique, the proposal for an updated directive seeks to enhance the lifecycle management framework of aerospace defense systems in the FAB, emphasizing modern practices, risk management, and comprehensive safety analyses.

The MD40-M-01 emphasizes continuous improvement, proactive risk management, and stakeholder involvement, aligning seamlessly with the ISO/IEC/IEEE 15288:2023 standard. These principles address gaps in the original DCA 400-6 directive, providing a robust foundation for adapting to the complexities of contemporary aerospace defense

systems. By aligning with these frameworks, the updated directive would enable compliance with international standards while fostering operational efficiency and safety.

This modernization effort also incorporates the ISO/IEC/IEEE 15288:2023's structured lifecycle management approach, emphasizing iterative development, validation, and verification processes. Together, these enhancements ensure that the revised directive reflects the latest systems engineering and lifecycle management advancements.

4.1.1 STPA for Eliciting ROPs and RTLIs

A key component of this update's proposal is the integration of an adapted Vee-Model, grounded in ISO/IEC/IEEE 15288:2023 and tailored to the needs of the Brazilian Air Force. This adapted framework incorporates STPA phases to align hazard identification and requirements definition with lifecycle milestones. The Vee-Model's structure is divided into the following key stages:

1. Phase 1 of STPA for ROP Derivation: during the initial conceptual phase, STPA Phase 1 is applied to identify system-level hazards and derive high-level Operational Requirements (ROP). These requirements are comparable to mission-level needs outlined in ISO/IEC/IEEE 15288:2023 and ensure alignment with operational goals and safety constraints.
2. Phases 2 to 4 of STPA for RTLI Development: subsequent STPA phases (2 to 4) are integrated into the decomposition and system definition stages to develop the Technical, Logistics, and Industrial Requirements (RTLI). These detailed requirements address unsafe control actions and establish comprehensive safety constraints for system implementation.

4.2 Why Performing an STPA on a UCAV

To justify the selection of an Unmanned Combat Air Vehicle (UCAV) as a case study for this doctoral thesis, it is important to address both the technological significance and the strategic implications of UCAVs in contemporary military operations. UCAVs represent a class of unmanned aerial systems that are revolutionizing the nature of modern warfare due to their versatility, cost-effectiveness, and ability to perform complex missions in increasingly contested operational environments (Jordan, 2021).

The ongoing evolution of global conflicts underscores the critical role that UCAVs play in reshaping military strategies. In particular, recent conflicts, such as the war in Ukraine, have highlighted the transformative impact of drones and UCAVs on the battlefield (Thompson, 2024). These systems have proven instrumental in providing real-time surveillance, precision targeting, and the capability to undertake high-risk missions without jeopardizing human lives. Notably, the widespread use of drones like the Turkish TB2 Bayraktar has demonstrated that relatively small and inexpensive UCAVs can achieve objectives traditionally reserved for more advanced and costly manned aircraft (Atherton, 2023). This shift underscores the increasing reliance on UCAVs as cost-effective force multipliers in modern warfare.

Moreover, UCAVs offer distinct operational advantages, including autonomous and semi-autonomous capabilities, enabling them to execute missions such as intelligence gathering, surveillance, reconnaissance, and direct engagement with minimal human intervention (Jordan, 2021). These attributes make UCAVs representative of the complexities and technological advancements that characterize modern aerospace systems, providing an ideal platform for the application of STPA.

The selection of a UCAV aligns closely with the objectives of this research, which seeks to explore the application of STPA in aerospace defense systems. As highly complex systems, UCAVs embody the integration of advanced technologies such as artificial intelligence, communication networks, and weapons systems, all of which require rigorous safety and reliability analyses. By focusing on a UCAV, this research ensures that the methodologies developed and validated through this study address the challenges of modern aerospace defense systems in their entirety.

The strategic importance of UCAVs extends beyond leading military powers. They have increasingly become accessible to middle powers, enabling countries (like Brazil) to enhance their military capabilities without the need for prohibitively expensive manned platforms (Milan; Bassiri Tabrizi, 2020). This democratization of advanced aerial combat technology underscores the necessity of robust safety and reliability frameworks to ensure these systems can be deployed effectively and safely across diverse operational contexts. For institutions such as the Brazilian Air Force, the adoption of UCAVs represents a critical step toward addressing strategic objectives while managing resource constraints. Given their complexity, operational significance, and growing prevalence in modern military strategies, UCAVs serve as an exemplary case for applying STPA.

4.3 Eliciting Requirements With LLMs & STPA

The elicitation of system requirements using LLMs within the framework of System-Theoretic Process Analysis (STPA) represents a key contribution of this research. By leveraging the generative and analytical capabilities of LLMs, specifically ChatGPT-4, the methodology outlined in this Section aims to address a longstanding challenge in aerospace systems engineering: the efficient and consistent derivation of system requirements from safety analysis. This process transcends traditional hazard analysis by generating actionable design requirements that directly address identified losses, hazards, and safety constraints, thereby creating a seamless bridge between safety considerations and system engineering practices.

The elicitation process builds upon the foundational steps of STPA. Particularly, the first phase of STPA involves defining system-level *Losses*, identifying the *Hazards* that may lead to those losses, and formulating *Safety Constraints* (intended to avoid the hazards occurrence), which can be derived to elicit systems-level requirements. These elements serve as the inputs for deriving high-level requirements that ensure system functionality while mitigating potential risks. The integration of LLMs into this process introduces a transformative capability to enhance efficiency, consistency, and scalability. Through iterative queries and prompt refinement, ChatGPT-4 is tasked with interpreting the contextual nuances of each STPA component and producing tailored requirements that align with the operational and safety objectives of the analyzed system.

A key enabler of this approach lies in the adaptability and precision of ChatGPT-4 when guided by carefully constructed prompts. Specific excerpts from the STPA Handbook (Leveson; Thomas, 2018) were instrumental in this process, providing the methodological rigor necessary to ensure that the outputs adhered to established safety engineering principles. The detailed prompts utilized in this study outlined the technical definitions and interrelations of *Losses*, *Hazards*, and *Safety Constraints*, ensuring that the generated requirements were both comprehensive and contextually accurate. Additionally, these prompts were designed to minimize common pitfalls associated with automated analyses, such as ambiguities in terminology or misinterpretation of causal relationships.

To derive requirements at lower hierarchical levels within the system (e.g., subsystems and components), it is necessary to extend the Prompt Engineering approach proposed herein. However, ChatGPT still fails to execute Phase 2 of STPA in a minimally satisfactory manner.

Therefore, it is recommended that this phase of the analysis be carried out using the traditional manual approach. Nevertheless, some degree of computational assistance can be incorporated into modeling the Hierarchical Control Structure (HCS) by leveraging Model-Based Systems Engineering (MBSE) tools, such as Capella, for instance.

Despite the difficulty LLMs face in modeling the HCS, they are even more valuable in Phases 3 and 4 of STPA than in its first phase. Identifying, mapping, and tracking the numerous Unsafe Control Actions (UCAs) and Loss Scenarios that naturally derive from a handful of Losses and Hazards is an arduous task. When subjected to appropriate prompt engineering techniques, ChatGPT-4 provides excellent suggestions for UCAs and Loss Scenarios while maintaining traceability throughout the analysis, making it an outstanding productivity tool.

Several studies highlight how much more labor-intensive it is to identify UCAs and Loss Scenarios. In Almeida's master's thesis (2024), from only four Losses and four Hazards, the analyst derived 40 UCAs and 541 Loss Scenarios. In "*Using STPA in Compliance with ISO 26262 for Developing a Safe Architecture for Fully Automated Vehicles*", Abdulkhaleq *et al*. (2017) identified 27 UCAs and 129 Loss Scenarios. In another representative work, Picanço *et al*. (2024) listed 145 "Loss Scenarios" from 6 "Losses" and 5 "Hazards". The deeper the STPA application aims to go, the exponentially larger the number of UCAs and Loss Scenarios required for a comprehensive and meaningful analysis becomes. This is precisely one of the critical areas where an LLM, as a productivity aid tool, adds substantial value.

The methodology employed here does not merely automate the STPA process but also seeks to validate its outputs against real-world benchmarks. Requirements elicited by ChatGPT-4 were compared to those derived from a real defense system currently operational within the Brazilian Air Force (FAB). This validation exercise highlights the practical relevance and applicability of the approach, demonstrating that AI-generated outputs can meet or exceed the expectations of domain experts in terms of accuracy and completeness. The results underscore the potential of LLMs to act as reliable collaborators in safety-critical domains, reducing the manual effort involved in requirement generation while maintaining high standards of quality and reliability.

In relation to item 4.1.2 of this doctoral thesis, Figure 4.2 demonstrates an approach aligned with the needs of the FAB. Therefore, Phase 1 of STPA can be utilized for the elicitation of system-level requirements, such as the ROPs. Upon completion of Phase 4,

subsystem/component level requirements comparable to the RTLIs of DCA 400-6 can be derived.

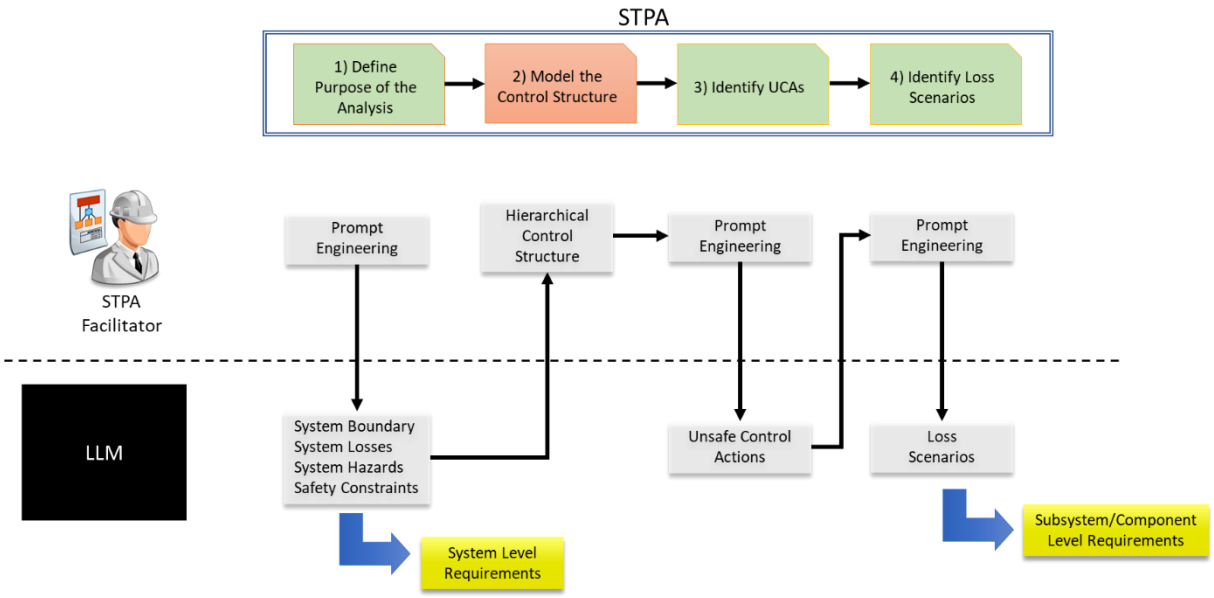


Figure 4.2 – Proposed workflow for acquiring both system and subsystem/component level requirements via STPA with LLM aid.

Another critical aspect of this work is the iterative refinement of LLM interactions to achieve optimal results. The iterative nature of querying ChatGPT-4 ensures that the requirements elicitation process is dynamic and responsive, enabling the incorporation of contextual feedback and domain-specific insights. By iteratively refining the prompts and evaluating the outputs against predefined criteria, the methodology establishes a robust framework for extracting high-quality system requirements in a structured and reproducible manner.

The subsequent discussion will delve into the principles of Prompt Engineering, a cornerstone of this methodology. By examining validated techniques for crafting effective queries, the next section will illuminate how targeted prompts can maximize the performance of ChatGPT-4. This will ensure that the elicited requirements are not only technically precise but also aligned with the broader objectives of aerospace systems engineering.

4.3.1 Prompt Engineering Best Practices

The newly emergent field of Prompt Engineering has met with some skepticism from conventional perspectives. While it may seem excessive to label these techniques as

"engineering", the reality is that computational language models obey rules like any other computer algorithm, and much can be structured in terms of how to achieve the best results from these models.

Sahoo *et al*. (2024) conducted a systematic survey of the main techniques and applications of Prompt Engineering for large language models (LLMs). After carefully analyzing the authors' results, I have concluded that the "Tree-of-Thought" approach is the most suitable for conducting an STPA with the assistance of ChatGPT-4o and ChatGPT o1-preview.

The "Tree-of-Thought" (ToT) approach is particularly well-suited for conducting an STPA with the assistance of ChatGPT-4o and ChatGPT o1-preview for several reasons. STPA is a comprehensive hazard analysis technique that examines the complex interactions within a system to identify potential safety issues and control flaws. This method requires a deep exploration of various system states, interactions, and potential failure modes, which aligns well with the iterative and expansive nature of the Tree-of-Thought approach.

Firstly, the ToT approach facilitates a structured exploration of multiple reasoning pathways. By conceptualizing the problem-solving process as a tree, it allows for the generation and evaluation of numerous potential solutions or scenarios simultaneously. This is crucial in STPA, where understanding the myriad ways in which system components can interact and potentially lead to hazardous states is essential. The tree structure identifies hidden relationships and emergent behaviors that might not be apparent through linear analysis.

Secondly, the ToT approach enhances the reasoning capabilities of large language models like ChatGPT-4o and ChatGPT o1-preview. These models can benefit from a framework that encourages divergent thinking and systematic exploration of possibilities. By employing the ToT methodology, the AI can delve deeper into each branch of thought, ensuring that no critical pathway is overlooked. This depth of analysis is particularly important in safety-critical systems, where overlooking a potential hazard can have severe consequences.

Moreover, the tree-of-thought approach's iterative nature aligns with the iterative processes inherent in STPA. Both methodologies emphasize continuous refinement and reassessment of the system model based on new insights. The ToT approach allows the AI to revisit previous nodes in the thought tree, incorporating new information or correcting earlier

assumptions. This dynamic reassessment is vital for accurately modeling complex systems and their potential failure modes.

Additionally, using the ToT approach with advanced models like ChatGPT-4o and ChatGPT o1-preview leverages their ability to handle complex, context-rich information. These models are designed to maintain coherence over extended interactions and can manage the intricacies of a thought tree with multiple branches and depth. This capability ensures that the analysis remains consistent and comprehensive throughout the STPA process.

Finally, the ToT approach supports collaboration between human analysts and AI. It provides a transparent framework where the reasoning process of the AI can be reviewed, validated, or augmented by human experts. This collaborative dynamic enhances the reliability of the analysis, and fosters trust in the AI's contributions to the STPA.

4.4 Teaching an LLM How to Attribute MoCs

This Section details the methodology employed in the research to automate the assignment of Means of Compliance (MoCs) to aerospace defense system requirements, leveraging the capabilities of Large Language Models (LLMs). The objective was to develop a computational application capable of accurately predicting the appropriate MoCs for a given requirement based solely on its textual description.

In the realm of aerospace defense systems, the verification and validation of requirements are critical to ensuring compliance with stringent safety and functional standards. A fundamental component of this process involves the assignment of appropriate MoCs, which serve as evidence of adherence to specified requirements. Traditionally, this task is highly labor-intensive, demanding significant time and expert judgment, particularly given the absence of standardized MoC guidelines tailored for defense systems.

Adopting the approach proposed in this research in human processes has the potential to offer numerous advantages, such as:

1. **Efficiency and Speed**: AIs can process and analyze large volumes of data much faster than a human can (Elkins; Sood; Rumpf, 2022).
2. **Continuous Availability**: Unlike humans, AIs can operate 24 hours a day, 7 days a week, without the need for breaks or rest, significantly enhancing productivity and responsiveness in critical processes (Wang; Liu, 2023).

3. **Error Reduction**: In repetitive tasks or those requiring high precision, AIs tend to make fewer errors than humans, as they are not affected by factors such as fatigue, stress, or distraction (Zdravković; Panetto; Weichhart, 2022).
4. **Complex Data Analysis**: AIs can identify patterns and insights in complex and voluminous datasets that would be virtually impossible for humans to analyze manually (Gichoya *et al*., 2022).
5. **Cost Reduction**: Although the initial investment in AI may be high, in the long term, it can reduce operational costs, especially in tasks requiring a large amount of human labor (Wang; Qiu, 2023).
6. **Human Performance Improvement**: AIs may work as extensions of human capabilities (Tran *et al*., 2020).

4.4.1 A Supervised Machine Learning Approach

The methodology leveraged a curated dataset of approximately 4,000 labeled requirements to fine-tune the LLM, enabling it to predict the most appropriate MoCs based solely on the textual description of each requirement. This approach, illustrated at Figure 4.3, follows a classical Machine Learning workflow comprising several key stages:

1. Data Preprocessing: The initial phase involved preprocessing the data used to train the LLM. This included converting the data from various formats (PDF, DOC, etc.) into a standardized XLSX format and then transforming it into a JSONL file, a format suitable for training LLMs.
2. Dataset Preparation: The processed dataset was then shuffled and split into three distinct sets: training, validation, and test. A classic division of 70% for training, 15% for validation, and 15% for testing was employed.
3. Model Selection and Fine-tuning: The 'gpt-3.5-turbo' model from OpenAI was selected for fine-tuning due to its advanced natural language processing capabilities. The model was fine-tuned using the prepared training dataset, with each instance consisting of a requirement's text and its corresponding MoCs.
4. Validation and Parameter Optimization: The fine-tuned model underwent a validation phase using the validation dataset. The number of epochs, a key hyperparameter, was adjusted to optimize the model's performance. The accuracy

of the model improved significantly with an increase in epochs, reaching an optimal point at around 15 epochs, where it demonstrated an accuracy of 80.78% when submitted to the validation dataset.

5. Testing: The final stage involved testing the trained model using the unseen test dataset. The model achieved an accuracy of 80.18%, demonstrating its effectiveness in automating the MoC assignment process and mitigating the possibility of overfitting, which occurs when a machine learning model learns the training data too well, including its noise and anomalies, resulting in reduced ability to generalize to unseen data.

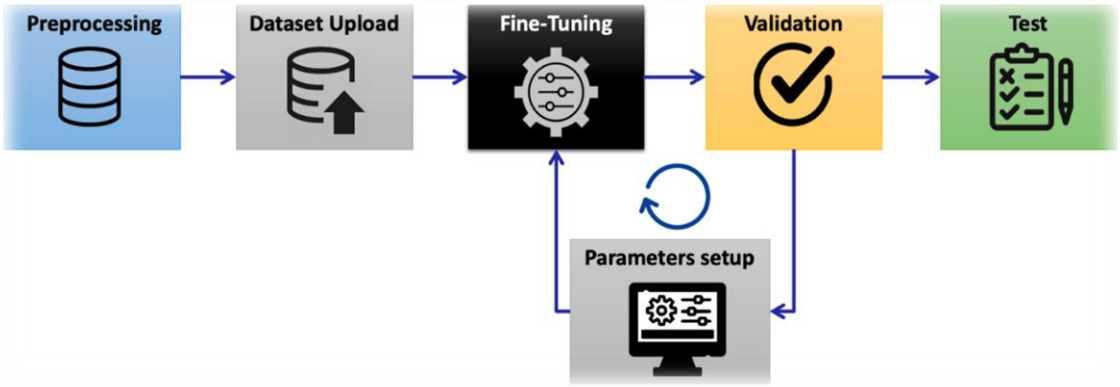


Figure 4.3 – The steps of LLM training: Preprocessing, Dataset Upload, Fine-Tuning, Validation and Test.

To provide a systemic understanding of the process proposed in this work to automate the task of assigning MoCs, it was prepared an OPM (Object Process Methodology) (DORI, 2002) model using the OPCloud (Opcloud, 2022) tool. Figure 4.4 displays the OPM model of the proposed approach. The detailed model description in a formal language is available at Appendix C through the OPL (Object Process Language) description generated by OPCloud exportation tool.

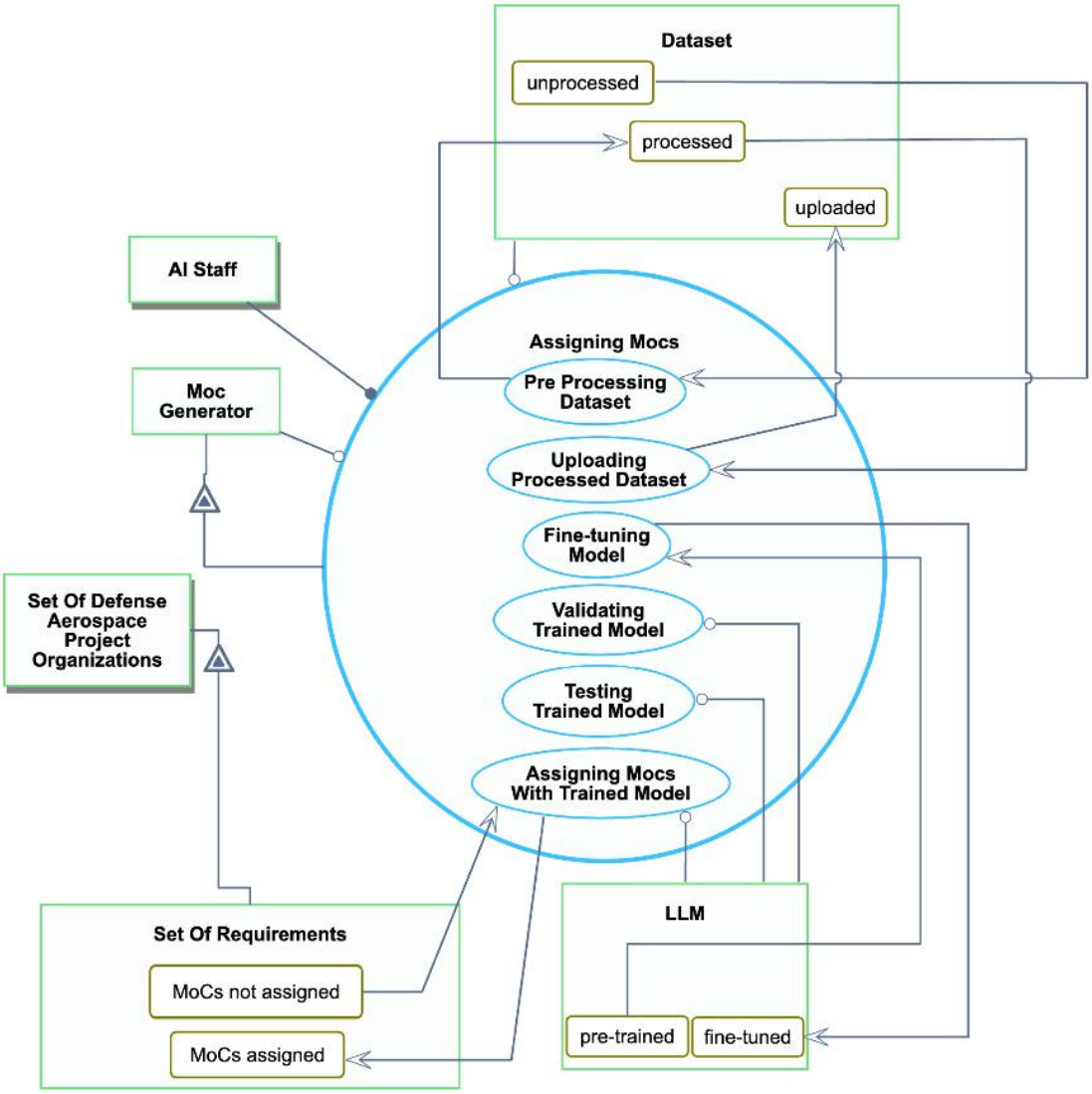


Figure 4.4 – OPM model of the proposed approach.

Considerable effort was expended in extracting information from their conventional databases. It is common for part of the data to be in file types such as PDF, DOC, ODT and other formats. After converting everything to the XLSX format and eliminating other information irrelevant to the research problem, it was necessary to convert this large spreadsheet into the JSONL format (similar to the JSON format, with the peculiarity that each entry must occupy exclusively only one line). A script was written to perform this task, and its pseudocode can be found in Appendix D.1. Figure 4.5 summarizes the Preprocessing formatting phase undertaken.

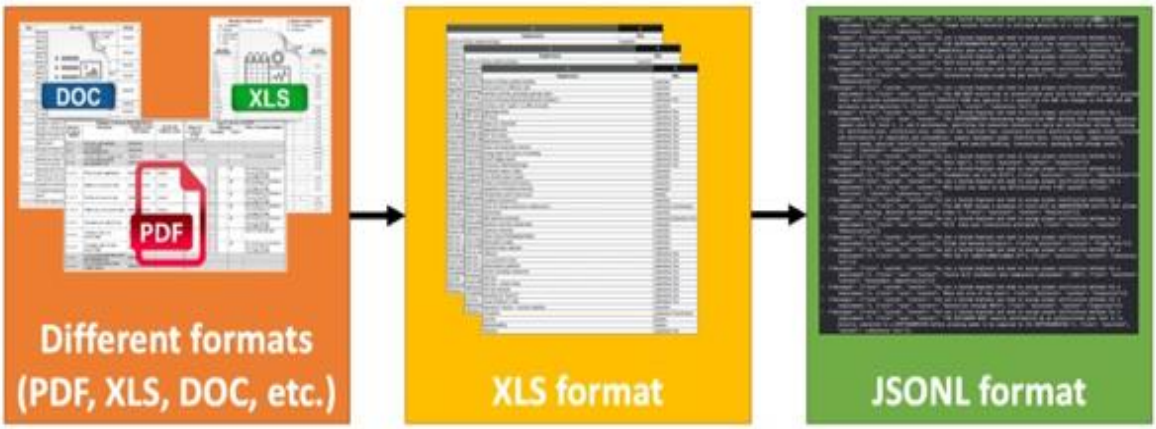


Figure 4.5 – Schematic view of the preprocessing formatting process.

Another critical aspect of the approach employed was the redistribution of dataset instances through the shuffling technique. In this study, the data were initially grouped by product or compliance matrix. However, shuffling is a crucial step in data preprocessing to ensure that the model is not biased by the original order of instances in the dataset. This process helps prevent overfitting and enhances the model's generalization capability. Subsequently, the dataset was split following a classical machine learning division: 70% of the data was allocated for training, 15% for validation, and 15% for testing, as supported by various studies (Chang *et al*., 2020; Tabibu; Vinod; Jawahar, 2019; Willemink *et al*., 2020). Figure 4.6 illustrates the processes of shuffling and data splitting.

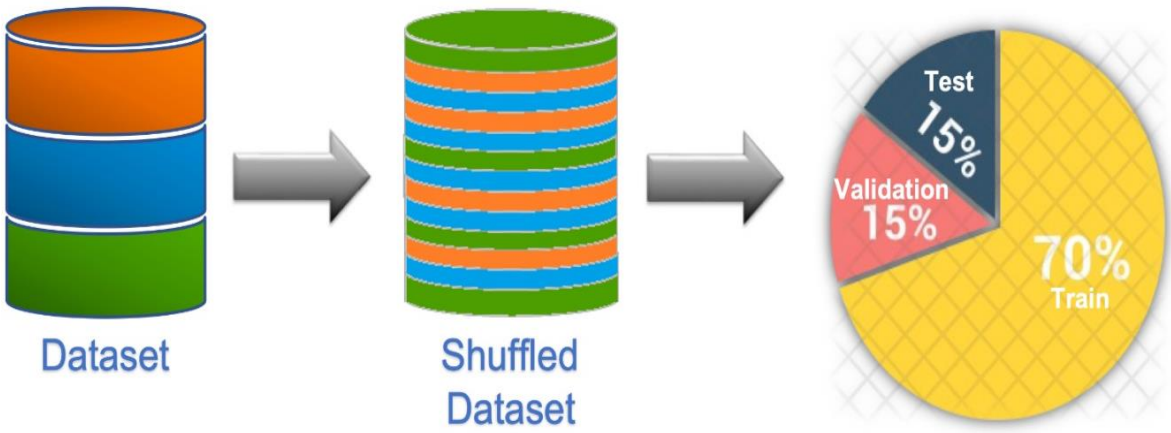


Figure 4.6 – Shuffling and data splitting before submitting the dataset to the LLM fine-tune.

The model employed for fine-tuning was the “gpt-3.5-turbo” from OpenAI, accessed via API. This model utilizes a specific template for data input, akin to other models. Some models initially used as concept demonstrators for this work employ inputs of the type:

{“prompt”: “prompt message”; “completion”: “model answer”}

The terms “prompt” and “completion” vary from one model to another, yet this template is the most commonly accepted by LLMs. The “gpt-3.5-turbo” model employs a slightly different template, incorporating not only the prompt input and the model’s response but also a systemic entry that contextualizes the task. Each entry in the JSONL file should exhibit the following structure:

{“messages”: [“role”: “system”, “content”: “Systemic/contextualization message”, “role”: “user”, “content”: “prompt message”, “role”: “assistant”, “content”: “model answer”]}

After all these adjustments, it was possible to initiate the next phase, where we upload the Dataset. Once again, a new Python script was responsible for establishing the connection with the OpenAI API and sending the data, and its pseudocode is also included in Appendix, D.3.

Subsequently, the most critical phase of this approach: train a pre-trained LLM in natural language related to numerous fields of human knowledge and provide it with an emphasis, a specialty. The model was presented to 2790 inputs and their respective outputs. This causes a series of weights and internal parameters of its neural network to be retrained for a specific context, and our expectation is that the model will be capable of producing valid outputs when presented with new inputs. The script pseudocode is available in Appendix D.4, Subsection B, and it was used to connect with the OpenAI API and perform the training. The default hyperparameters of the training function “*fine_tuning.jobs.create*” from the “*openai*” library were used.

The “gpt-3.5-turbo” model received the Dataset instances of requirements’ text and their respective methods of verification. Since the training was made via API, it takes place within OpenAI’s GPUs, which greatly facilitates the task. Training LLMs locally requires a whole setup and configuration of the machines that will perform the training, costing human, power supply, and computational resources. An unimaginable list of issues can occur with the configurations of these machines, and outsourcing this activity is of great help. However, it is necessary to consider the issue of the secrecy of the information being handled. It’s very likely that OpenAI (as well as other competitors like Claude3.5 and Gemini) respects its user privacy policy (OpenAI, 2024b), but it is impossible to be certain.

After the training was completed, we made some queries to the model, now trained, to define Means of Compliance MoCs given the text of a requirement. The results seemed consistent. In the first query, the model already correctly identified the verification method of the requested requirement and, when queried two more times, remained consistent with the result presented. At this point, the model seemed ready to undergo the validation phase.

Once again, it was necessary to prepare a script to validate the model. Its pseudocode can be found in Appendix D.5. Upon completing the script execution, we obtained an initially result: an accuracy of 47.92%, correctly identifying 266 MoCs out of 555 requested requirements.

A new training session was conducted, this time by altering the number of epochs from 3 (the default value) to 5. This simple action entails the dataset traversing the neural network an additional two times, thereby reinforcing the weights of the parameters in the "gpt-3.5-turbo" model that are pertinent to achieving the training objectives. This is attributable to the model being afforded more opportunities to learn from the dataset. However, there is a juncture beyond which further increments in the number of epochs may not only cease to yield significant improvements but may also begin to precipitate overfitting. For a dataset of the magnitude of a thousand entries, such as the one we employed, the conventional practice is to commence Fine-Tuning with 3 epochs (Dodge *et al*., 2020; Stollenwerk, 2022). Training tutorials, like those from "Hugging Face" (Huggingface, 2024), advocate for the gradual escalation in the number of epochs to attain enhanced outcomes, all the while exercising caution to avert overfitting. A high degree of accuracy in the validation and test sets suggests that the model is generalizing well, which mitigates concerns regarding overfitting (Suzuki; Matsuzawa, 2022; Yao; Koller, 2023).

As a consequence of the increase in the number of epochs, the accuracy of the trained model jumped to 77.47%, accurately predicting the MoCs of 430 requirements out of 555 in the validation dataset. This promising advancement indicated that it was worthwhile to invest in increasing the number of epochs, until such an increase no longer contributed to the model's performance improvement.

A new round of validation was conducted, this time using a number of epochs equal to 10. This resulted in an accuracy of 79.81%, correctly predicting the MoCs of 443 requirements. The outcome was better but not significantly different from the result with 5 epochs. We trained the model again, raising the number of epochs to 15 and, as a result, the model achieved an accuracy of 80.72%, correctly predicting the MoCs of 448 requirements

from the validation dataset. At this point in the study, the model's performance appeared to exhibit asymptotic behavior, indicating the reaching of some methodological limit.

To confirm our suspicion, we executed the validation script once more, training the model with 20 epochs. The model correctly assigned the MoCs of 448 requirements once again, maintaining an accuracy level of 80.72%. Figure 4.7 presents the graph where accuracy points versus the number of epochs were plotted.

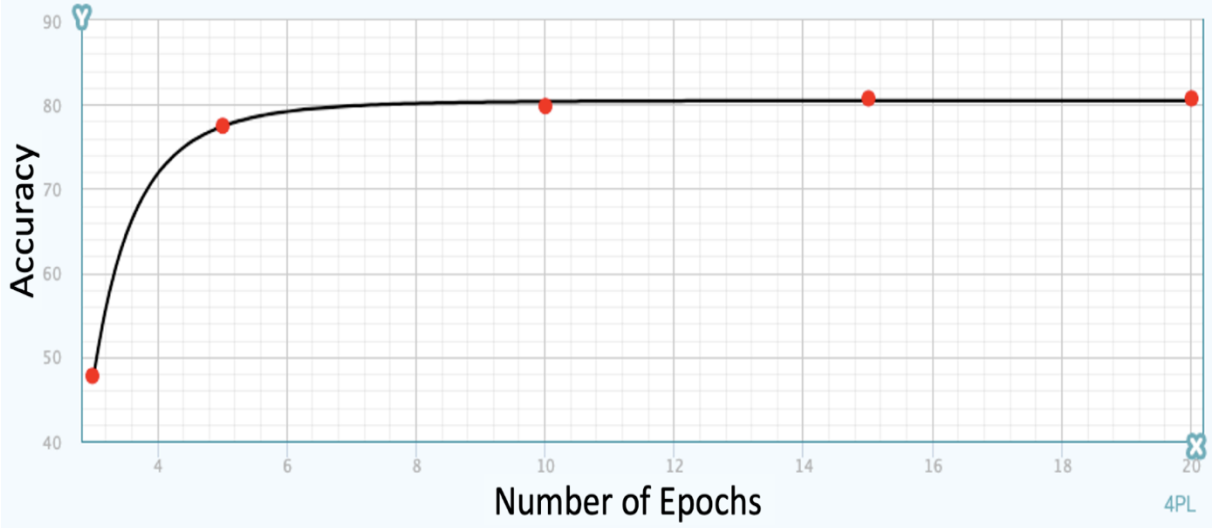


Figure 4.7 – Accuracy Vs Number of Epochs used in the five training sessions.

The asymptotic behavior of the curve in Figure 4.7 strongly suggests that continuing to increase the number of epochs may waste computational effort and cause overfitting, as discussed before. Consequently, the next phase was conducted, utilizing the test dataset to measure the performance of the trained model.

The Python script used for testing was identical to the one utilized for validation, merely loading a distinct dataset that had been segregated for this purpose at the outset of the study. Similar to its predecessor, the Test Dataset also consisted of 555 requirements from aerospace defense systems. Although slightly lower than that achieved in the validation phase, the result remained significant: an **accuracy of 80.18%**, having correctly assigned the MoCs to 445 requirements.

Thus, in practical terms, the fine-tuned GPT-3.5-turbo model has become specialized in assigning MoCs to requirements of aerospace defense systems—an activity that demands significant effort from development teams. The training dataset consists of labeled data, comprising thousands of requirements along with their corresponding MoCs. Essentially, the

pre-trained model was exposed to these thousands of examples, and through the fine-tuning process, it "learned" to assign MoCs to this type of product using only the requirements' texts as input. At its core, fine-tuning involves making small adjustments to the weights of the model's neural network layers, effectively tailoring it to this specific task. The validation and testing phases further confirm the model's ability to assign MoCs to new requirements, mitigating the possibility that its strong performance is merely a result of overfitting.

The outcomes achieved using the proposed approach appeared sufficient to justify comparing them with the classical method of evaluating which MoCs should be utilized to fulfill a requirement. Section 5.3.1 discusses this comparison.

4.5 Using LLMs to Generate Safety Reports

This Section documents an experiment conducted to evaluate the potential of Large Language Models (LLMs), specifically GPT-4o, for generating safety reports, such as the Preliminary Hazard Analysis (PHA), but also useful for Functional Hazard Analysis (FHA), Failure Mode and Effects Analysis (FMEA), and other similar safety assessments. The study leverages technical documentation from a real-world aerospace product—the BEF-1502 electronic fuze—kindly provided by Mac Jee, a Brazilian defense company. The goal was to assess the capabilities of LLMs to produce accurate, structured, and standards-compliant safety analyses, and to explore how these tools could complement traditional engineering workflows.

4.5.1 Case Study Description

The experiment involved utilizing GPT-4o to generate a PHA for the BEF-1502 electronic fuze, a sophisticated electromechanical device developed by Mac Jee. This system was chosen due to its nor too complex neither too simple architecture, serving as a good mid-level complexity example. The PHA process adhered strictly to the guidelines of MIL-STD-882E, a military standard widely recognized for system safety analysis. The objective was to determine whether LLMs could produce a PHA that met the rigorous expectations of military airworthiness authorities.

To perform such an experiment, the following materials were utilized:

1. **Technical Documentation**: The detailed technical specifications of the BEF-1502 electronic fuze provided a comprehensive understanding of the system's

components, energy sources, interfaces, controls, and operational environments in a premature development stage, exactly when the PHA should be made available. The authorization for using such documentation is available at Appendix B.2.

2. **MIL-STD-882E**: This standard served as the methodological framework for conducting the PHA.
3. **Prompt Engineering Techniques**: Techniques such as chain-of-thought reasoning, few-shot prompting, and explicit formatting instructions were employed to guide GPT-4o in generating the analysis.

A carefully crafted prompt (available at Appendix, session A.2) was developed to guide GPT-4o in generating the PHA. The prompt included the following instructions:

1. **System Overview**: Begin by analyzing the components, energy sources, interfaces, controls, and operational environments based on the technical documentation.
2. **Hazard Identification**: Identify hazards using the MIL-STD-882E framework, focusing on areas such as system components, ordnance, hazardous materials, environmental factors, and human factors engineering.
3. **Hazard Analysis**: For each hazard, document the function causing the hazard, specific failure modes, associated mission phases, and potential severity and likelihood of occurrence.
4. **Risk Assessment**: Evaluate risks using MIL-STD-882E's Risk Assessment Codes (RACs).
5. **Risk Mitigation**: Propose mitigation measures in line with the system safety design order of precedence.

The output was expected in a tabular format, including columns for Function ID, Function, Failure Mode, Hazard Description, Mission Phase, Severity, and Risk Mitigation.

GPT-4o was tasked with generating the PHA based on the provided prompt. The model processed the BEF-1502 technical specifications and MIL-STD-882E guidelines to identify and analyze hazards systematically. The resulting PHA included 20 hazards, each associated with a specific function of the electronic fuze, and detailed mitigation measures, and it is available at Table 4.1.

Table 4.1 – PHA from technical documentation and prompt engineering with GPT-4o.

Function ID	Function	Failure Mode	Hazard Description	Mission Phase	Severity	Risk Assessment Code (RAC)	Proposed Mitigation
F1	Power supply from FZU to FMU	Battery failure	Fuze does not receive power, fails to arm	Pre-release	I	2	Use redundant battery systems; periodic battery health checks.
F2	Arming time selection	Selector malfunction	Incorrect arming time leads to premature detonation	Pre-release	I	3	Implement selector switch redundancy; verify setting before mission.
F3	Delay to function	Delay timer failure	Improper delay results in unwanted detonation timing	Post-release	II	2	Use triple redundant delay hardware timers; perform software validation tests.
F4	Safe distance parameter	Incorrect parameter value	Detonation occurs within unsafe distance	Arming	I	3	Use sensors to verify distance before arming; software cross-checks for parameter validity.
F5	Connection between FZU and FMU	Cable damage	Loss of communication or power	Pre-release	III	4	Use reinforced MK70 cables; regular pre-mission inspections.
F6	DSU-33 Proximity Sensor Integration	Signal failure	Proximity detonation fails	Airburst	III	4	Perform integration testing with DSU-33; introduce error correction in signal processing.
F7	Voltage Booster (40V to 1200V)	Circuit overload	Component damage or inability to detonate	Arming	II	3	Design circuit protection with surge protectors; validate booster under operational conditions.
F8	Supercapacitor charge	Capacitor fails to charge	Fuze cannot detonate	Arming	III	4	Use high-reliability capacitors rated for operational conditions.
F9	Environmental exposure	Extreme temperatures	System failure due to thermal stress	Storage / operation	III	4	Ensure storage within specified temperature range; use thermal protection materials.
F10	Software-driven arming logic	Software bug	Arming sequence executed incorrectly	Arming	II	2	Perform rigorous software safety and failure mode testing; adhere to NATO-AOP-52 guidance.
F11	FZU safety activation pin	Inadvertent removal	Unintended arming of fuze	Pre-release	I	2	Redesign activation mechanism to require deliberate and forceful action for pin removal.
F12	Proximity sensor input	Signal interference	False or no detonation signal	Airburst	III	3	Shield sensor circuits; use signal filtering algorithms.
F13	Detonator initiation	LEEFI failure	No detonation occurs	Impact	II	2	Use redundant detonators; validate LEEFI reliability with extensive testing.
F14	Mechanical integrity	Loose closure ring	Loss of fuze during flight	Pre-release	III	4	Torque closure rings to specification during installation; use locking mechanisms.
F15	Operator error in setting delay parameters	Incorrect delay setting	Mission fails due to premature detonation	Pre-release	II	3	Provide clear, labeled controls; introduce software checks to validate operator inputs.

Table 4.1 – PHA from technical documentation and prompt engineering with GPT-4o (cont.)

Function ID	Function	Failure Mode	Hazard Description	Mission Phase	Severity	Risk Assessment Code (RAC)	Proposed Mitigation
F16	High Voltage Charge Bank	Overcharging	Component damage or fire	Arming	I	3	Install overcharge protection circuitry; conduct routine maintenance on the charging system.
F17	MK70 Cable connector	Loose or faulty connection	Loss of power or signal	Pre-release	III	4	Use connectors with locking mechanisms; inspect cable connections during pre-mission checks.
F18	Human factors	Inadequate training	Operator errors during setup	Pre-release	III	4	Develop comprehensive training programs; implement checklists for setup and installation procedures.
F19	Impact sensor	Sensor malfunction	Failure to trigger detonation upon impact	Impact	III	4	Use redundant impact sensors; calibrate sensors before each mission.
F20	Built-in self-tests	Failure to detect faults	Fuze enters mission with undetected defects	Pre-release	II	3	Develop robust self-test protocols; implement fail-safe modes to disable defective systems.

5 Results

This chapter delves into the results obtained from the methodological approaches outlined in Chapter 4. It provides a comprehensive analysis of the outputs generated by LLMs in automating key aspects of aerospace systems engineering, specifically focusing on the elicitation of system requirements, the automated assignment of MoCs, and the generation of safety reports. The chapter critically examines the performance of LLMs in comparison to traditional methods and expert evaluations, highlighting both the transformative potential and current limitations of AI in this safety-critical domain.

The analysis presented here will illuminate the extent to which LLMs can streamline complex engineering processes, enhance consistency, and potentially surpass human capabilities in certain tasks. It will also address the challenges and nuances encountered in leveraging AI for aerospace applications, emphasizing the complementary role of human expertise in ensuring robust and reliable outcomes. By dissecting the research results, this chapter aims to provide a deeper understanding of the implications for aerospace systems engineering, paving the way for future research and development in this rapidly evolving field.

5.1 FAB Lifecycle Directive Proposal

This Section deals with the tangible outcomes of the proposed framework, highlighting the specific enhancements and modifications introduced to the DCA 400-6 document. It details how the integration of STPA and the adapted Vee-Model, incorporating principles from MD40-M-01 and ISO/IEC/IEEE 15288:2023, may guide FAB's Directive review to obtain a document more robust, safety-focused, and aligned with international standards. The discussion will showcase the practical implications of these changes, emphasizing their potential to improve the lifecycle management of aerospace defense systems within the Brazilian Air Force.

5.1.1 Vee-Model Proposal for the Brazilian Air Force

The adapted Vee-Model (Figure 5.1) introduces critical milestones derived from ISO/IEC/IEEE 15288:2023 and MD 40-M-01, ensuring traceability and systematic progression throughout the system lifecycle. The key milestones are:

1. MCR (Mission Concept Review): The MCR affirms the mission need and examines the proposed mission's objectives and the concept for meeting those objectives.
2. ASR (Alternative Systems Review): This review assesses alternative solutions and ensures that the selected system concept meets the operational needs and requirements. It evaluates the feasibility and risks associated with different options.
3. SRR (System Requirements Review): This milestone ensures that system requirements are complete, feasible, and verifiable. It confirms that the requirements are correctly defined and meet the needs of stakeholders.
4. SFR (System Functional Review): This review focuses on the system's functional baseline, verifying that all functional requirements are properly defined and allocated. It ensures that the system's functional architecture can meet the specified requirements.
5. PDR (Preliminary Design Review): This review assesses the preliminary design against the system requirements. It ensures that the design approach meets all functional and performance requirements and is ready to proceed to detailed design.
6. CDR (Critical Design Review): The CDR confirms that the detailed design meets all system requirements with acceptable risk and is ready for full-scale development. It evaluates the design maturity and completeness.
7. TRR (Test Readiness Review): This milestone ensures that the system and its components are ready for testing. It verifies that the test procedures, facilities, and configurations are prepared for execution.
8. SVR (System Verification Review): The SVR verifies that the system meets all specifications and requirements. It usually occurs together with the FCA (Functional Configuration Audit) to ensure that the final system configuration matches the documented specifications and requirements.
9. PRR (Production Readiness Review): This review assesses the readiness of the system for production. It ensures that the production processes, tools, and facilities are in place and capable of producing the system to the required specifications.

10. PCA (Physical Configuration Audit): The PCA verifies that the physical configuration of the system matches the documented design and requirements. It is typically performed before system delivery to ensure all configuration items are properly documented and controlled.

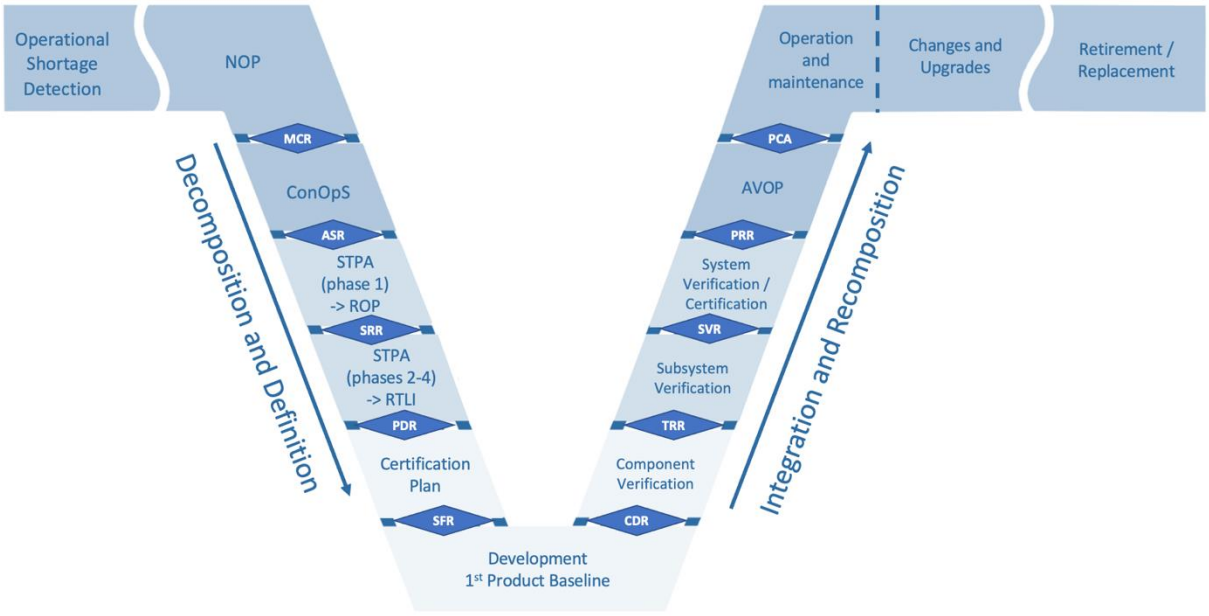


Figure 5.1 – Proposed Vee-Model for DCA 400-6.

These milestones are interwoven with STPA activities to ensure that safety constraints and hazard analyses are iteratively refined and validated at each stage. The integration of STPA into DCA 400-6 offers several advantages:

1. **Enhanced Hazard Analysis**: STPA's focus on both component failures and systemic interactions ensures comprehensive hazard identification and mitigation.
2. **Iterative Safety Refinement**: STPA's iterative nature allows for continuous improvement of safety constraints, adapting to evolving operational and technical requirements.
3. **Alignment with Best Practices**: The directive achieves global compatibility by adhering to ISO/IEC/IEEE 15288:2023 and MD 40-M-01 principles, facilitating international collaboration and standardization.

To implement the proposed framework effectively, a structured, phased approach is recommended:

1. **Training and Familiarization**: Comprehensive training programs are essential for equipping stakeholders with the knowledge and skills required to apply the Vee-Model and STPA methodologies effectively.
2. **Pilot Projects**: Initial implementation in selected pilot projects will provide valuable insights into the model's practicality and areas for refinement.
3. **Iterative Refinement**: Feedback from pilot projects will inform iterative adjustments to the model, ensuring its robustness and adaptability.
4. **Full-Scale Deployment**: Upon validation, the framework can be rolled out across all relevant projects, supported by continuous monitoring and updates to maintain alignment with evolving standards.

The proposed update to the DCA 400-6 represents a significant advancement in lifecycle management for aerospace defense systems. By integrating principles from MD 40-M-01, aligning with ISO/IEC/IEEE 15288:2023, and embedding STPA into the Vee-Model, the updated directive would potentially ensure a modern, rigorous, and safety-focused approach to system development. This framework not only addresses the operational and technical complexities of contemporary aerospace systems but also establishes a proactive foundation for continuous improvement and risk management.

5.1.2 Where Does this Work Apply?

This Section presented a framework designed to serve as the implementation environment for this study within the FAB, aligning the primary standard governing the lifecycle of aerospace systems with the leading systems engineering benchmarking in the Western world. In this context, it is important to highlight how the concepts explored in Sections 5.2, 5.3, and 5.4 integrate into the Vee Diagram in Figure 5.1:

1. The elicitation of requirements through the execution of the first phase of STPA enables the capture of the ROP.
2. The elicitation of requirements through the complete execution of STPA (upon completion of phase 4) enables the capture of the RTLI.
3. The assignment of MoCs, using the method presented in Section 4.4, facilitates the construction of the compliance matrix, which must be included in the Certification Plan.

4. The generation of Safety Reports is applicable from the development of the RTLI until the completion of the Verification/Certification process, as such documents remain dynamic and subject to modifications until the Product Baseline reaches its final configuration, which only occurs at the PRR.

5.2 Results of STPA Using an LLM

This Section delves into the results obtained from the application of STPA using an LLM. It presents a detailed analysis of the LLM's performance in generating key STPA components, including the identification of losses, hazards, and safety constraints, for a specific case study. The evaluation of the LLM-generated outputs against real-world benchmarks, along with expert feedback, provides insights into the effectiveness and limitations of leveraging AI in conducting STPA for aerospace defense systems.

5.2.1 ChatGPT Answers

The prompt text available in Appendix A.1 was used to obtain the complete first phase of the STPA. This prompt was tested in both ‘4o’ and ‘o1-preview’ models. The outcome of ChatGPT o1-preview was superior and is available in Figure 5.2 to Figure 5.8.

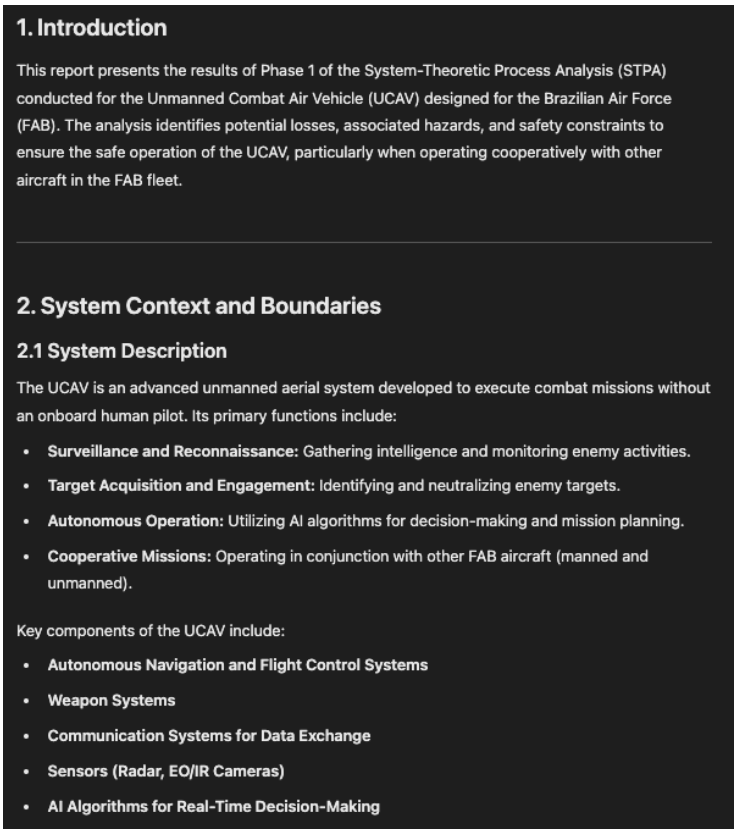
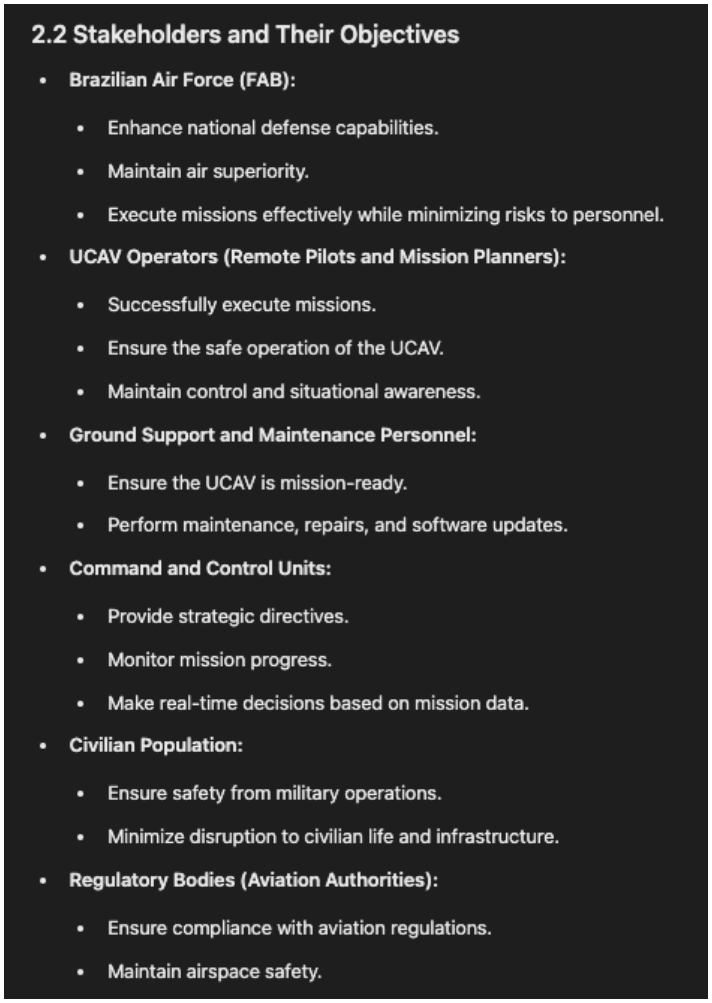


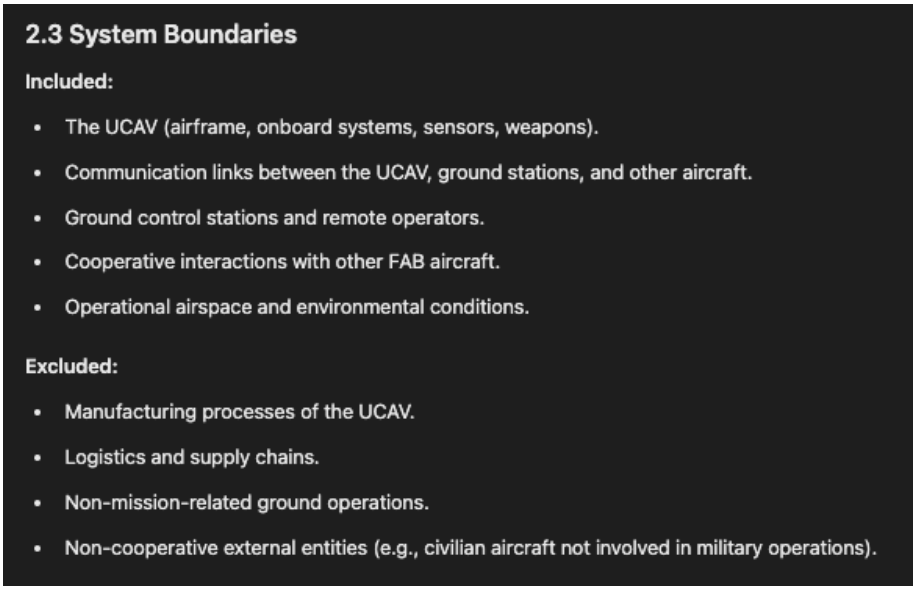
Figure 5.2 – Introduction and System Description produced by ChatGPT o1-preview.



2.2 Stakeholders and Their Objectives

- **Brazilian Air Force (FAB):**
  - Enhance national defense capabilities.
  - Maintain air superiority.
  - Execute missions effectively while minimizing risks to personnel.
- **UCAV Operators (Remote Pilots and Mission Planners):**
  - Successfully execute missions.
  - Ensure the safe operation of the UCAV.
  - Maintain control and situational awareness.
- **Ground Support and Maintenance Personnel:**
  - Ensure the UCAV is mission-ready.
  - Perform maintenance, repairs, and software updates.
- **Command and Control Units:**
  - Provide strategic directives.
  - Monitor mission progress.
  - Make real-time decisions based on mission data.
- **Civilian Population:**
  - Ensure safety from military operations.
  - Minimize disruption to civilian life and infrastructure.
- **Regulatory Bodies (Aviation Authorities):**
  - Ensure compliance with aviation regulations.
  - Maintain airspace safety.

Figure 5.3 – Stakeholders and their objectives produced by ChatGPT o1-preview.



2.3 System Boundaries

Included:

- The UCAV (airframe, onboard systems, sensors, weapons).
- Communication links between the UCAV, ground stations, and other aircraft.
- Ground control stations and remote operators.
- Cooperative interactions with other FAB aircraft.
- Operational airspace and environmental conditions.

Excluded:

- Manufacturing processes of the UCAV.
- Logistics and supply chains.
- Non-mission-related ground operations.
- Non-cooperative external entities (e.g., civilian aircraft not involved in military operations).

Figure 5.4 – System Boundaries produced by ChatGPT o1-preview.

3. Identifying Losses

Loss ID	Description	Stakeholders Impacted
L-1	Loss of human life (military personnel and civilians)	FAB, UCAV Operators, Civilian Population
L-2	Injury to people	FAB, UCAV Operators, Civilian Population
L-3	Loss or damage of the UCAV	FAB, Ground Support Personnel
L-4	Unintended engagement of friendly forces (fratricide)	FAB, Other FAB Aircraft, Command Units
L-5	Mission failure (failure to accomplish mission objectives)	FAB, Command Units, UCAV Operators
L-6	Violation of airspace regulations	Regulatory Bodies, FAB
L-7	Compromise of sensitive information (e.g., data interception)	FAB, National Security Agencies
L-8	Damage to property or infrastructure	Civilian Population, Regulatory Bodies
L-9	Adverse environmental impact	Civilian Population, Environmental Agencies

Figure 5.5 – Identified Losses produced by ChatGPT o1-preview.

4. Identifying Hazards

Hazard ID	Description	Linked Losses
H-1	UCAV unintentionally fires weapons at friendly or neutral targets	L-1, L-2, L-4
H-2	UCAV collides with another aircraft	L-1, L-2, L-3
H-3	UCAV crashes into populated areas due to system failure	L-1, L-2, L-3, L-8
H-4	UCAV operates at unsafe altitudes over civilian areas	L-2, L-8
H-5	Failure of sense-and-avoid systems leading to near-miss incidents	L-2, L-6
H-6	Loss of communication leading to improper fail-safe responses	L-3, L-5, L-6
H-7	System malfunction causing loss of control	L-3, L-5
H-8	Misidentification of targets due to sensor errors	L-1, L-4
H-9	Incorrect data sharing causing faulty situational awareness	L-4, L-5
H-10	Navigation system errors leading to off-course flight	L-5, L-6
H-11	Inaccurate mission planning data	L-5
H-12	UCAV enters restricted airspace without clearance	L-6
H-13	Failure to comply with altitude and route constraints	L-6
H-14	Communication links are intercepted or jammed	L-5, L-7
H-15	UCAV is captured by enemy forces	L-3, L-7
H-16	UCAV crashes into civilian structures	L-2, L-3, L-8
H-17	Weapon malfunction causing unintended detonation	L-1, L-8
H-18	Fuel leakage during crash causing environmental contamination	L-9
H-19	Unexploded ordnance left in operational areas	L-8, L-9

Figure 5.6 – Identified Hazards produced by ChatGPT o1-preview.

5. Identifying Safety Constraints

Constraint ID	Description	Linked Hazards
SC-1	UCAV must ensure weapons are only fired at positively identified enemy targets.	H-1, H-8
SC-2	UCAV must maintain safe separation from all other aircraft using collision avoidance systems.	H-2, H-5
SC-3	UCAV must execute controlled emergency landings in unpopulated areas in case of system failures.	H-3, H-16
SC-4	UCAV must operate at designated safe altitudes over civilian areas.	H-4
SC-5	UCAV must have reliable sense-and-avoid systems to detect and evade other airspace users.	H-2, H-5
SC-6	UCAV must enter a predefined safe mode upon loss of communication, maintaining compliance with airspace regulations.	H-6
SC-7	UCAV must have redundant control systems to maintain control in case of malfunctions.	H-7
SC-8	UCAV must validate target identification with multiple sensor modalities before engagement.	H-1, H-8
SC-9	UCAV must ensure data integrity and accuracy when sharing information with other units.	H-9
SC-10	UCAV must have accurate navigation systems with periodic cross-checks to stay on course.	H-10
SC-11	Mission planning data must be verified for accuracy before mission execution.	H-11
SC-12	UCAV must have geo-fencing capabilities to prevent entry into restricted airspace.	H-12
SC-13	UCAV must comply with assigned altitude and route constraints during all phases of flight.	H-13
SC-14	UCAV's communication links must be encrypted and resilient to interception and jamming.	H-14
SC-15	UCAV must have self-destruct or data purge capabilities to prevent sensitive information compromise if captured.	H-15
SC-16	UCAV must avoid flight paths over civilian structures when possible.	H-3, H-16
SC-17	UCAV's weapons must undergo checks to prevent malfunction and unintended detonation.	H-17
SC-18	UCAV must have fuel containment measures to prevent leakage in case of a crash.	H-18
SC-19	UCAV must ensure all deployed ordnance is tracked and detonated or recovered.	H-19

Figure 5.7 – Identified Safety Constraints produced by ChatGPT o1-preview.

6. Iterative Refinement

An iterative review was conducted to ensure completeness and accuracy:

- **Additional Considerations:** No additional losses, hazards, or constraints were identified beyond those listed.
- **Consolidation:** Overlapping hazards and constraints were analyzed for potential consolidation but were retained to address specific aspects of system safety.
- **Traceability:** Ensured clear linkage between losses, hazards, and safety constraints for effective risk mitigation.

---

7. Conclusion

This STPA Phase 1 analysis has identified critical losses, hazards, and safety constraints for the UCAV system. Implementing these safety constraints is essential to mitigate risks and ensure the safe and effective operation of the UCAV, particularly during cooperative missions with other FAB aircraft.

Figure 5.8 – Iterative Refinement and Conclusion produced by ChatGPT o1-preview.

5.2.2 Elicited Requirements

After completing the entirety of Phase 1 of the STPA, it was possible to derive requirements from the 'Safety Constraints' identified by ChatGPT. The initial iteration of Phase 1 of the STPA typically occurs when knowledge about the system is limited. Ideally, a system designed using the STPA technique has not yet undergone detailed specification. As a result, the output at this stage of the process consists of high-level requirements comparable to the Operational Requirements (ROPs) outlined in the Brazilian Air Force Directive DCA 400-6.

To test the hypotheses of this doctoral thesis, I needed to design an experiment to validate the obtained results. To this end, I sought assistance from the Directorate of the IFI to gain access to compliance matrices of aerospace defense products. Fortunately, compliance matrices are information present on the Certification Plan of Aerospace products, which are documents of an ostensive nature, not under the protection of any document access restriction. These data were crucial for the experiments presented in this chapter. Appendix B.1 includes the authorization for using this information in the training and validation of the approaches proposed in this study.

Considering the selection of the case study discussed in Section 4.2, it was essential to choose a subsystem that was present both in the material provided by the IFI and in the fictitious UCAV used to assess ChatGPT's capabilities in eliciting requirements. Additionally, the focus was on selecting requirements related to system safety, aligning with the central themes of this research. Ultimately, a subsystem that is universally necessary for any UCAV, regardless of its potential configuration, was chosen: the Ground Control Station (GCS).

A GCS serves as the main interface between human operators and UCAVs, facilitating comprehensive command and control over the vehicle's operations. Typically, a GCS comprises hardware and software components that enable operators to plan missions, monitor real-time telemetry data, and manage payloads. The hardware includes control interfaces, communication systems, and display units, while the software provides functionalities for flight planning, navigation, and system diagnostics. The GCS is responsible for transmitting control commands to the UCAV and receiving data streams, including sensor outputs and status updates, ensuring seamless communication throughout the mission. The criticality of the GCS in UCAV operations cannot be overstated; it ensures precise navigation, effective mission execution, and rapid response to dynamic operational scenarios. Moreover, the GCS

plays a vital role in safety management by enabling operators to detect and mitigate potential hazards in real-time, thereby safeguarding both the UCAV and surrounding environments. The design and functionality of GCSs are extensively discussed in the literature, highlighting their integral role in the efficacy and safety of UCAV missions (Sadraey, 2024).

The material provided by the IFI included only 13 safety-related requirements for a GCS. Therefore, to ensure statistical parity in the comparison, ChatGPT was tasked with deriving the same number of safety requirements for the GCS of our fictitious UCAV, based on the "Safety Constraints" identified during the STPA process.

The 13 real UAV requirements selected to this experiment were:

1. The GCS design shall aim for full compliance with the most severe safety requirements within specified environmental conditions.
2. The GCS design shall consider engineering safety issues.
3. The GCS shall make efforts to eliminate or control all identified hazards to an acceptable level of risk.
4. The GCS design shall consider operational safety issues.
5. The GCS shall undergo verification through safety engineering analysis.
6. The criteria in accepting the residual risks and establishing necessary corrective actions will be defined, in conjunction with normative safety documents.
7. The GCS shall eliminate or mitigate System catastrophic events.
8. The GCS design shall contribute to the safety integration effort by controlling and mitigating hardware and software hazards that may affect UAV System integrity.
9. The risk assessment model is intended to be used along with the GCS and UAV System Hazard analyses.
10. The GCS design shall incorporate the accumulated experience of verified safety engineering criteria in complex UAV integrated systems.
11. The GCS shall ensure the highest degree of safety, consistent with its operational requirements, throughout its life cycle.
12. The resolutions will be reflected in the Safety Assessments Reports in terms of severity and probability of occurrence.

13. The resolution of identified land and air hazards in the GCS constituents will be defined and documented.

The 13 ChatGPT-generated requirements for a fictitious UCAV based on Phase 1 of STPA were:

1. The GCS operation shall request comprehensive training programs and implement standard operating procedures to minimize the potential for human error.
2. The GCS design shall implement reliable power supply solutions, including backup systems, to prevent operational interruptions due to power loss.
3. The GCS design shall implement robust authentication and access control mechanisms to ensure that access to UAV control systems is restricted to authorized personnel only.
4. The GCS design shall ensure software reliability through rigorous testing, regular updates, and prompt resolution of identified issues.
5. The GCS safety analysis shall establish and maintain a robust risk management framework that ensures all risks are assessed, managed, and documented appropriately.
6. The GCS safety analysis shall incorporate comprehensive hazard identification and mitigation processes, including regular reviews and updates, to address all identified safety risks.
7. The GCS design shall implement measures to protect against adverse environmental conditions and ensure safe operational continuity.
8. The GCS design shall adhere to all relevant safety standards and undergo regular audits and updates to ensure compliance.
9. The GCS design shall implement robust cybersecurity measures to protect against cyber-attacks and vulnerabilities.
10. The GCS safety policy shall conduct thorough and accurate safety assessments and maintain comprehensive reporting practices to ensure all safety-related information is documented and reviewed.
11. The GCS shall maintain robust communication channels and implement backup systems to ensure continuous and reliable communication with the UAV/UCAV.

12. The GCS operation shall request the development and enforcement standard operating procedures and the provision of ongoing training to ensure personnel are adequately prepared for their roles.
13. The GCS design shall ensure the reliability of all system components through rigorous testing, regular maintenance, and prompt resolution of technical issues.

After the selection process, they were compiled into a list of 26 requirements and included in a survey answered by 13 experts in Systems Engineering and Aerospace Systems Certification. The order of the requirements was randomized, and **no identifiers were provided that would allow respondents to distinguish between requirements generated by ChatGPT and those belonging to the real system**. The name and other identifications of the real UAV used in this study are preserved to protect the manufacturer's intellectual property and reputation.

5.2.3 Why a Survey?

Expert review is a well-established practice in engineering, often employed to evaluate artifacts and ensure quality. Pressman (2010) highlights expert review as a crucial validation technique, where specialists examine software artifacts to identify potential issues and ensure adherence to standards. Similarly, Sommerville (2015) emphasizes the importance of expert review in validating software, recognizing its ability to uncover defects and improve the overall quality of the final product.

In the context of requirements engineering, Nuseibeh and Easterbrook (2000) underscore the challenges associated with eliciting and validating requirements. They stress the importance of involving stakeholders, particularly domain experts, throughout the process. Maciaszek (2001) further reinforces the necessity of validating requirements with stakeholders, including experts, as a critical step in requirements analysis and system design.

The use of an expert survey aligns with these established practices. This approach provides an exempt and external validation, enhancing the trustworthiness of the generated requirements.

The survey method offers a structured approach to gathering expert feedback. It allows for a systematic comparison of the requirements generated through the ChatGPT-4 and STPA approaches against those of a real-world UAV system. This comparative analysis provides valuable insights into the effectiveness of the proposed approach and helps identify areas for

potential improvement.

5.2.4 Method's Limitation

It is important to acknowledge the challenges encountered in leveraging LLMs for the initial phases of STPA, particularly in modeling the hierarchical control structure (HCS). While LLMs excel at natural language processing and text generation, their capabilities in understanding and representing complex system architectures, which are crucial for HCS modeling, are still under development. This limitation can hinder the effectiveness of LLMs in providing comprehensive support for the early stages of STPA.

However, this limitation can be mitigated by tools like the "STPA Viewpoint for Capella", which offer a structured environment for HCS modeling, providing functionalities for defining system components, their interactions, and control loops. This Capella's add-on is a software tool that provides an STPA perspective for the Arcadia method and Capella modeling tool. By integrating the STPA approach into the Capella environment, it enables the conduction of hazard analysis early in the system design lifecycle. The tool supports the creation of control structure diagrams, hazard tables, and links to system models, facilitating a comprehensive safety assessment. This integration promotes a proactive approach to safety, allowing engineers to identify and mitigate potential hazards before they manifest in the physical system. The add-on enhances the capabilities of Capella by incorporating the STPA methodology, contributing to the development of safer and more reliable systems. Therefore, "*STPA Viewpoint for Capella*", can greatly facilitate the task of the STPA analyst in modeling the HCS.

While this study provides valuable insights into the potential of ChatGPT-4 and STPA for requirements elicitation, it is essential to acknowledge certain limitations. Firstly, the sample size of 13 requirements may be considered limited. However, each requirement underwent a comprehensive evaluation process, incorporating feedback from 13 experts across nine distinct criteria. This in-depth analysis of individual requirements, coupled with the aggregated comparison, provides a robust assessment despite the limited sample size. Additionally, it is crucial to consider the specific context of aerospace systems. Due to their complexity and criticality, even a small sample of requirements can offer significant insights into the overall quality and characteristics of the generated requirements.

Secondly, the number of experts involved in the study, while representative of a substantial expert panel, could be perceived as a limitation. However, the selection of

experienced professionals in systems engineering and aerospace certification mitigates this concern. The participants possess extensive domain knowledge and practical experience, working at remarkable organizations in the Aerospace Systems Engineering discipline. This ensures that their evaluations are well-informed and reliable. Furthermore, the focus on a specific domain, aerospace systems, allows for a more targeted expert selection, ensuring that the participants have the necessary expertise to assess the nuances of the generated requirements.

Finally, the inherent subjectivity of expert judgment can be considered a potential limitation. While the study employed a structured survey with clearly defined evaluation criteria to mitigate this subjectivity, it is crucial to acknowledge that individual interpretations and biases may still exist. However, using a diverse expert panel helps minimize the impact of individual biases, as the aggregation of multiple perspectives provides a more balanced and objective assessment.

5.2.5 Survey Outcome

The experts participating in this study had varying levels of experience. The study aimed to replicate the composition of a real-world Systems Engineering team with a heterogeneous and representative distribution. Figure 5.9 and 5.10 provide insights about the profiles of the participating experts.

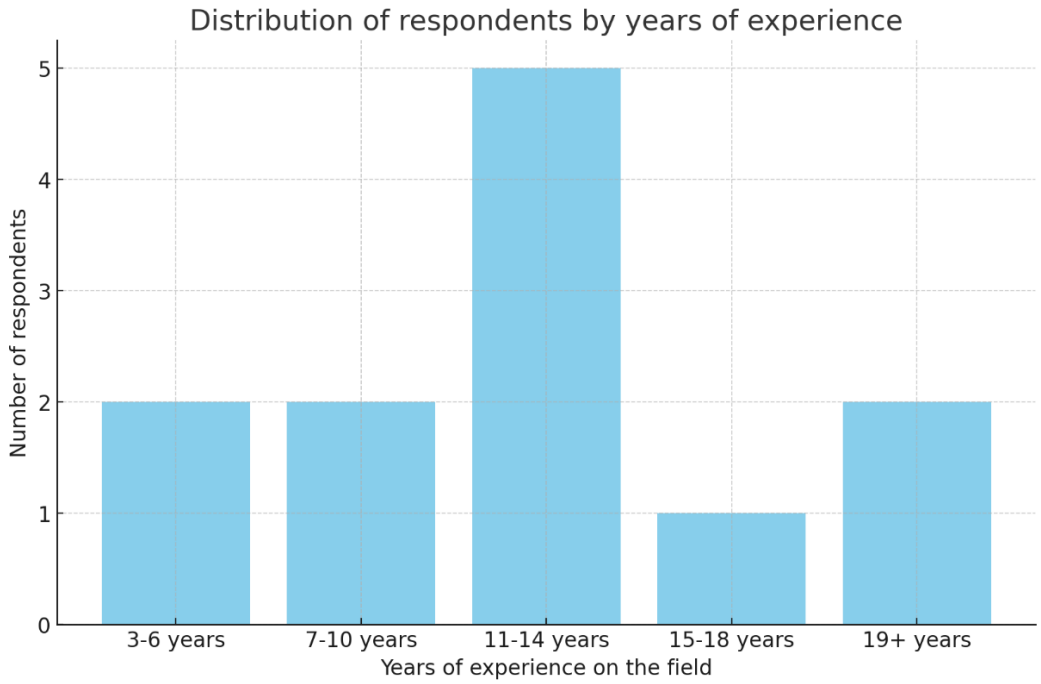


Figure 5.9 – Experience of Systems Engineering experts participating in the survey.

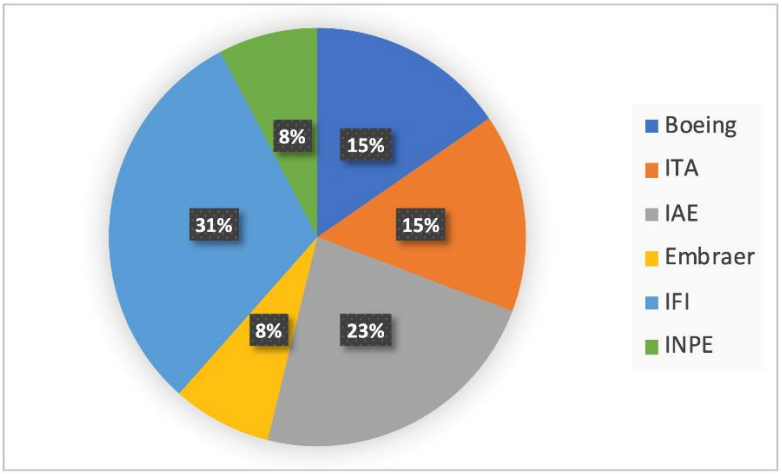


Figure 5.10 – Organizations where the experts who evaluated the requirements work.

The survey was conducted using a Google Forms questionnaire, which is available at https://forms.gle/YmtaeFNszTfwzXrH9> and in Appendix B.3. The experts' answers, which are the raw data used to illuminate the insights discussed in this chapter, are available at https://docs.google.com/spreadsheets/d/1t649xg_6lJjY0sqTmnkFTCKzEq_-Uqd7N2q_GP-O_74/edit?usp=sharing>.

The method used to evaluate the requirements is based on the principles of requirements engineering outlined by Robert Halligan (Halligan; Cpeng, 1993) Halligan's framework emphasizes the importance of requirements' quality attributes that are critical for the success of any engineering project: Correctness, Completeness, Consistency, Clarity, Non-ambiguity, Connectivity, Singularity, Testability, Modifiability, and Feasibility. However, the Connectivity attribute was neglected in favor of the survey's nature and its purpose to prevent experts from realizing the origin of each requirement. Connectivity refers to the property whereby all of the terms within the requirement are adequately linked to other requirements and to word and term definitions, so causing the individual requirement to properly relate to the other requirements as a set. The requirements' presentation order was shuffled and each requirement deattached from any relationship with the others. This made impossible the task to evaluate the connection among them. The definitions of the other nine evaluated attributes are:

1. Correctness refers to an absence of errors of fact in the statement of requirement.
2. Completeness requires that the requirement contain all of the information necessary, including constraints and conditions, to enable the requirement to be implemented such that the need will be satisfied.

3. Consistency requires that a requirement not be in conflict with any other requirement, nor with any element of its own structure.
4. Clarity requires that the requirement be readily understandable without semantic analysis.
5. Non-Ambiguity requires that there be only one semantic interpretation of the requirement.
6. Singularity refers to the attribute whereby a requirement cannot sensibly be expressed as two or more requirements having different subjects, verbs and/or objects.
7. Testability refers to the existence of a finite and objective process with which to verify that the requirement has been satisfied.
8. Modifiability requires that necessary changes to a requirement can be made completely and consistently.
9. Feasibility requires that a requirement be able to be satisfied:
    (a) within natural physical constraints;
    (b) within the state-of-the-art as it applies to the project; and
    (c) within all other absolute constraints applying to the project.

Following the collection of survey responses, the number of experts who assigned the presence of each attribute to each requirement was tallied. The requirements were then separated into their two original groups: those generated with the assistance of ChatGPT and those originating from the actual UAV. Finally, the average of these counts was calculated per group. Figure 5.11 summarizes the gain from using the approach presented here (STPA + ChatGPT) in comparison to requirements belonging to a real-world system developed in a traditional manner. Among all the assessed attributes, the requirements from a real comparable system had a better performance only in one property (Singularity), and ChatGPT presented a better outcome in eight attributes: Correctness, Completeness, Consistency, Clarity, Non-ambiguity, Connectivity, Testability, Modifiability, and Feasibility.

Figure 5.11 – Spyder chart synthesizing the improvement perceived by experts over ChatGPT generate requirements.

This analysis underscores the potential of leveraging advanced language models, like ChatGPT, in conjunction with established systems engineering techniques, such as STPA, to generate high-quality requirements. The findings indicate that this approach not only yields requirements comparable to those derived from traditional methods but also presents opportunities for enhanced efficiency and innovation in the requirements development process. This outcome serves as a compelling foundation for the subsequent exploration of automating the assignment of MoCs to aerospace defense system requirements. By extending the capabilities of LLMs through fine-tuning, the aim is to streamline further and optimize the compliance process, ensuring the development of robust and reliable aerospace systems. The following section delves into the intricacies of this automated MoC assignment approach, detailing the methodology and evaluating its effectiveness.

5.3 Results of Using LLM to Attribute MoCs

This Section presents the results of utilizing a Large Language Model (LLM) to attribute Means of Compliance (MoCs) to aerospace defense system requirements. It analyzes the performance of a fine-tuned LLM in accurately assigning MoCs based on textual descriptions of requirements. The evaluation includes a comparison of the LLM's performance against human experts in the field, highlighting the potential of AI to streamline compliance processes while maintaining or exceeding expert-level accuracy.

5.3.1 Performance Comparison: Trained Model Vs. Human Experts

A new survey (available on Appendix B.4) was prepared, following the method already discussed in Subsection 5.2.3. This time, 14 aerospace development and/or certification experts were asked, tasking them with attributing MoCs to 20 specific requirements from the dataset. The survey was also distributed to the selected experts via Google Forms, and it is available at:

- <https://forms.gle/3ynwZxSw2Qe3mhfL8>

A Support Material was prepared to standardize the understanding over each MoC meaning, regardless the experts' technical background. It is available at:

- https://drive.google.com/file/d/1awwM5C6ZHzYJEmlwshAiPMWTnvpnjMIL/view>

Those requirements were carefully chosen since most of the requirements on the original Dataset are too short, providing little to no context for the experts to better comprehend the requirements needs. Furthermore, we sought to choose requirements in order to balance the MoCs involved, thus preventing a certain type of MoC from appearing much more than others.

The experts possess a variety of backgrounds and lengths of experience, yet all have held or currently hold key positions in the development or certification of aerospace defense products. Each has participated in actual tasks of assigning or reviewing MoCs in Brazilian aerospace defense projects. Figure 5.12 displays a chart depicting the distribution of experts by their length of experience in the development and/or certification of aerospace products: 22% have 6 to 10 years of experience; 50% have 11 to 15 years of experience; 14% have 16 to 20 years of experience; and the last 14% have more than 30 years of experience.

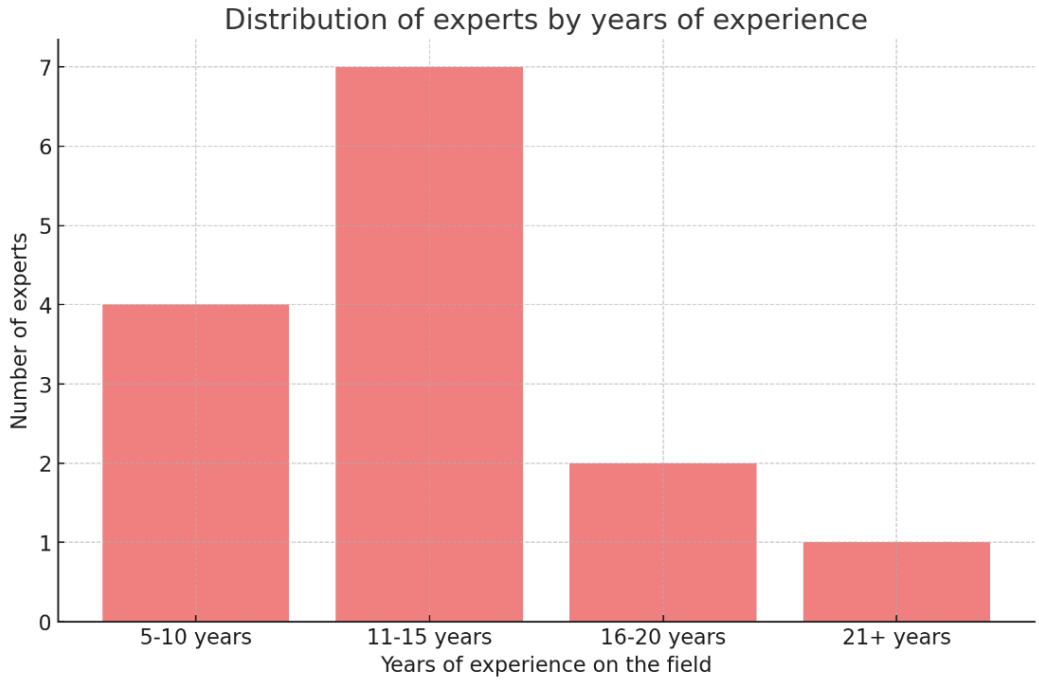


Figure 5.12 – Experts experience on the field.

Figure 5.13 shows the organizations where the surveyed experts work: 14% work for ITA, the Aeronautics Institute of Technology; 50% work for IFI (Industrial Fostering and Coordination Institute), the Brazilian Military Airworthiness Authority; 22% work for IAE, the Brazilian Space and Aeronautics Institute; 7% work for ICEA, the Brazilian Institute of Airspace Control; and 7% of them work for INPE, the Brazilian National Institute for Space Research.

Figure 5.13 – Development and/or Certification Experts' organizations.

Interestingly, the performance of these experts was significantly inferior to that of our trained model. Their accuracy average was only 6.07%. This discrepancy may stems from two factors:

1. Lack of Context: The mere text of a requirement often fails to convey its full scope, with the understanding profoundly influenced by the system's context.
2. Collaborative Nature of MoC Assignment: Typically, the assignment of MoCs is a collaborative endeavor, involving a multidisciplinary team that consults standards, engages in discussions, and reaches a consensus on the applicable MoCs. In contrast, our survey limited experts to their individual expertise and the Support Material.

5.3.2 Method's Limitation

Despite these challenges, the study's outcomes do not detract from the survey's value. On the contrary, they highlight the subjective nature of MoC assignment and suggest that artificial intelligence can map this subjectivity using just the requirement text. This capability significantly streamlines the MoC request process for new requirements, obviating the need for additional context and extensive normative research.

While the results advocate for the potential of AI to economize on human and time resources substantially, they also underscore the indispensability of expert judgment, especially for critical requirements, warranting verification team scrutiny. Therefore, while this AI-driven approach offers a promising avenue for enhancing efficiency in MoC attribution, it should complement rather than supplant expert involvement.

5.4 Results of Using LLM to Generate Safety Reports

This Section presents the results of utilizing an LLM to generate safety reports for aerospace systems. It provides a detailed analysis of the LLM's performance in generating a Preliminary Hazard Analysis (PHA) for a real-world aerospace product, the BEF-1502 electronic fuze. The evaluation includes expert reviews and iterative improvements, highlighting the LLM's strengths and limitations in producing accurate, structured, and standards-compliant safety analyses.

5.4.1 Validation and Iteration

To validate the generated PHA, the document was reviewed by four former employees from IFI. These experts used the IFI's Technical Analysis Form (FAT) to assess the PHA as if it were a real submission for certification. Their evaluations concluded that the PHA was of high quality, and it would be considered approved with restrictions (requiring some adjustments). Their FATs are available at Appendix E.

The primary feedback from the experts included:

1. Missing legend for severity categories, RAC categories, and mission phases.
2. Lack of qualitative probability values for hazards.
3. Absence of hazards related to electromagnetic interference/compatibility.
4. Missing consideration of human error during pre-installation inspection.
5. No mention of hazards related to storage conditions.

Using the feedback from the FATs, the PHA was revised and improved. Additional hazards were identified, including:

1. Electromagnetic interference/compatibility.
2. Human error during pre-installation inspection.
3. Improper storage conditions.

A second iteration of the PHA was produced, addressing all identified issues. This revised version, available in Table 5.1, was submitted to one of the original reviewers, Captain Vitor Bourguignon, who had eight years of experience analyzing similar documents at the IFI. Captain Bourguignon, despite the minor inaccuracy made by GPT-4o over the three additional identified hazards (the last three hazards got already existing identifiers), would fully approve the revised PHA, noting that it was now compliant with the requirements of MIL-STD-882E. He also pointed out the necessity of renumbering the additional hazards with exclusive identification. Since the PHA is used as an input for further safety assessment, these identifiers will follow the hazards throughout the evolution of their safety assessment during the system development in documents such as Functional Hazard Analysis (FHA), System Hazard Analysis (SHA) etc.

Interestingly, in this second iteration, GPT-4o also provided a legend for some PHA attributes (available in Table 5.1), although this was not requested.

Table 5.1 – Revised PHA obtained from former IFI employees' feedback.

Function ID	Function	Failure Mode	Hazard Description	Mission Phase	Severity	RAC	Proposed Mitigation	Probability
F1	Power supply from FZU to FMU	Battery failure	Fuze does not receive power, fails to arm	Pre-release	I	2	Use redundant battery systems; periodic battery health checks.	Low
F2	Arming time selection	Selector malfunction	Incorrect arming time leads to premature detonation	Pre-release	I	3	Implement selector switch redundancy; verify setting before mission.	Low
F3	Delay to function	Delay timer failure	Improper delay results in unwanted detonation timing	Post-release	II	2	Use triple redundant delay hardware timers; perform software validation tests.	Low
F4	Safe distance parameter	Incorrect parameter value	Detonation occurs within unsafe distance	Arming	I	3	Use sensors to verify distance before arming; software cross-checks for parameter validity.	Low
F5	Connection between FZU and FMU	Cable damage	Loss of communication or power	Pre-release	III	4	Use reinforced MK70 cables; regular pre-mission inspections.	Low
F6	DSU-33 Proximity Sensor Integration	Signal failure	Proximity detonation fails	Airburst	III	4	Perform integration testing with DSU-33; introduce error correction in signal processing.	Low
F7	Voltage Booster (40V to 1200V)	Circuit overload	Component damage or inability to detonate	Arming	II	3	Design circuit protection with surge protectors; validate booster under operational conditions.	Low
F8	Supercapacitor charge	Capacitor fails to charge	Fuze cannot detonate	Arming	III	4	Use high-reliability capacitors rated for operational conditions.	Low
F9	Environmental exposure	Extreme temperatures	System failure due to thermal stress	Storage	III	4	Ensure storage within specified temperature range; use thermal protection materials.	Low
F10	Software-driven arming logic	Software bug	Arming sequence executed incorrectly	Arming	II	2	Perform rigorous software safety and failure mode testing; adhere to NATO-AOP-52 guidance.	Low
F11	FZU safety activation pin	Inadvertent removal	Unintended arming of fuze	Pre-release	I	2	Redesign activation mechanism to require deliberate and forceful action for pin removal.	Low
F12	Proximity sensor input	Signal interference	False or no detonation signal	Airburst	III	3	Shield sensor circuits; use signal filtering algorithms.	Low
F13	Detonator initiation	LEEFI failure	No detonation occurs	Impact	II	2	Use redundant detonators; validate LEEFI reliability with extensive testing.	Low
F14	Mechanical integrity	Loose closure ring	Loss of fuze during flight	Pre-release	III	4	Torque closure rings to specification during installation; use locking mechanisms.	Low
F15	Operator error in setting delay parameters	Incorrect delay setting	Mission fails due to premature detonation	Pre-release	II	3	Provide clear, labeled controls; introduce software checks to validate operator inputs.	Low
F16	High Voltage Charge Bank	Overcharging	Component damage or fire	Arming	I	3	Install overcharge protection circuitry; conduct routine maintenance on the charging system.	Low
F17	MK70 Cable connector	Loose or faulty connection	Loss of power or signal	Pre-release	III	4	Use connectors with locking mechanisms; inspect cable connections during pre-mission checks.	Low

Table 5.1 – Revised PHA obtained from former IFI employees' feedback (cont.).

Function ID	Function	Failure Mode	Hazard Description	Mission Phase	Severity	RAC	Proposed Mitigation	Probability
F18	Human factors	Inadequate training	Operator errors during setup	Pre-release	III	4	Develop comprehensive training programs; implement checklists for setup and installation procedures.	Low
F19	Impact sensor	Sensor malfunction	Failure to trigger detonation upon impact	Impact	III	4	Use redundant impact sensors; calibrate sensors before each mission.	Low
F20	Built-in self-tests	Failure to detect faults	Fuze enters mission with undetected defects	Pre-release	II	3	Develop robust self-test protocols; implement fail-safe modes to disable defective systems.	Low
F6	System susceptibility to electromagnetic interference	EMI-induced failure	Unintended behavior due to external electromagnetic interference	Pre-release	II	2	Shielding and EMI testing in compliance with MIL-STD-461	Medium
F7	Pre-installation inspection	Dropped component undetected	Component damage not identified due to lack of inspection after drop	Pre-release	II	3	Mandate inspection if dropped; include checklists	Low
F8	Storage conditions	Improper storage environment	Exposure to moisture or extreme temperatures degrades functionality	Storage	III	4	Use controlled storage conditions as per MIL-STD guidelines	Low

Table 5.2 – PHA's legend generated by GPT-4o.

Category	Description
Severity	I: Catastrophic, II: Critical, III: Marginal, IV: Negligible
RAC - Risk Assessment Codes	1: High, 2: Serious, 3: Medium, 4: Low
Mission Phase	Storage, Pre-release, Arming, Post-release, Airburst, Impact

5.4.2 Strengths of GPT-4o

The experiment highlighted several strengths of GPT-4o in generating safety analyses:

1. **Efficiency**: The model rapidly produced a comprehensive PHA, significantly reducing the time required compared to manual efforts.
2. **Standards Compliance**: The output aligned closely with MIL-STD-882E guidelines, demonstrating the potential of LLMs to support rigorous engineering standards.
3. **Structure and Clarity**: The tabular format and detailed analyses were well-organized and easy to interpret.

5.4.3 Limitations of GPT-4o

Despite its strengths, GPT-4o exhibited limitations:

1. **Error in Identifier Assignment**: The model failed to assign unique identifiers to three newly added hazards, inadvertently reusing existing identifiers (F6, F7, and F8).
2. **Reliance on Prompt Quality**: The accuracy and completeness of the analysis were highly dependent on the clarity and detail of the prompt.
3. **Human Oversight Requirement**: The need for expert review and iterative improvements underscored the importance of human involvement in the safety analysis process.

5.4.4 Implications for Aerospace Certification

The results demonstrated that LLMs could serve as powerful tools for accelerating the development of safety analyses. However, their role should be seen as complementary rather than autonomous. Human analysts remain essential for validating and refining the outputs, ensuring accountability and compliance with certification requirements.

6 Conclusions

The findings of this research demonstrate that integrating artificial intelligence into aerospace systems engineering presents transformative opportunities and complex challenges. Building upon the methodologies and results presented in the preceding chapters, this discussion aims to critically analyze the broader implications of employing LLMs in the aerospace defense sector, particularly for tasks such as system requirements elicitation and the automated assignment of MoCs.

6.1 Discussions on Research Findings

This Section examines the implications of the research findings in the context of the doctoral thesis objectives, research problem, and hypothesis. By exploring the results in detail, the discussion highlights the significance of the contributions to aerospace defense systems engineering, the advancements achieved in utilizing LLMs for requirements elicitation and MoCs assignments, and the broader impact of this work on the field.

6.1.1 Addressing the Research Problem

The research problem posed the question: "*How Large Language Models (LLMs) can automate the application of System Theoretic Process Analysis (STPA) to elicit aerospace defense systems' requirements, automatically assign their Means of Compliance (MoCs), and generate safety reports while maintaining or exceeding the current performance level of experts on the field*?"

This study's findings unequivocally presented strong evidence that LLMs can serve as robust tools in automating STPA-based requirement elicitation and MoC assignments. By achieving or exceeding expert-level performance in key areas of requirement quality and accuracy, the study validated the feasibility of the proposed approach. Automating these traditionally manual and resource-intensive processes marks a significant step toward streamlining aerospace systems engineering.

6.1.2 Evaluating the Hypothesis

The hypothesis of this research stated that "Large Language Models (LLMs), when guided by Prompt Engineering Techniques for System Theoretic Process Analysis (STPA),

can effectively automate the elicitation of aerospace defense systems' requirements, accurately assign Means of Compliance (MoCs) to these requirements through fine-tuning techniques, and generate reasonable safety reports' drafts, achieving or exceeding the performance accuracy of field experts in the aerospace defense industry."

This hypothesis was tested through three primary methodologies:

1. **Requirement Elicitation via STPA and Prompt Engineering**: The results of Phase 1 of STPA, conducted with the assistance of ChatGPT-4 and refined using Prompt Engineering, show a marked improvement in requirement quality compared to traditional methods. Across nine critical quality attributes, the AI-generated requirements surpassed real-world requirements in eight attributes, demonstrating enhanced correctness, completeness, consistency, clarity, non-ambiguity, testability, modifiability, and feasibility. These findings confirm the hypothesis that Prompt Engineering can guide LLMs to effectively elicit high-quality system requirements.
2. **Fine-Tuned MoC Assignment Model**: The fine-tuned 'gpt-3.5-turbo' model exhibited a remarkable accuracy of 80.18% in assigning MoCs to aerospace defense requirements, far exceeding the average accuracy of 6.07% demonstrated by human experts in a comparative survey. Although the experiment was conducted outside of ideal conditions, which would have required a collaborative and simultaneous effort by experts in assigning MoCs, it nevertheless demonstrated the high aptitude of LLMs in mapping the subjectivity of such a specific systems engineering task with minimal information (solely the textual requirements). AIs possess an immense capacity for pattern recognition, far surpassing human capabilities. Studies like "*AI recognition of patient race in medical imaging: a modelling study*" by Gichoya *et al*. (2022) reveal the unimaginable capabilities of these computational tools. In this paper, the authors demonstrate how AI was able to identify the race of patients from medical images (e.g., X-rays and computed tomography scans), which contain no discernible racial indicators detectable by human experts. Therefore, the results from the MoC assignment survey strongly support the hypothesis that fine-tuning enables LLMs to meet or surpass expert-level performance in MoC assignments.
3. **LLM-Driven Safety Report Generation:** The experiment conducted with GPT-4o for generating a Preliminary Hazard Analysis (PHA) demonstrated the

capability of LLMs to produce structured, standards-compliant safety reports aligned with the MIL-STD-882E framework. Using detailed technical documentation of the BEF-1502 electronic fuze, Prompt Engineering techniques, and iterative validation, the LLM generated a high-quality PHA with 20 identified hazards, mitigation measures, and risk assessments. Expert reviews, performed by former IFI professionals, assessed the initial PHA as "approved with restrictions," citing minor issues such as missing legends and overlooked hazards. A revised version of the PHA, addressing expert feedback, received full approval from one reviewer with over eight years of experience, affirming its compliance with MIL-STD-882E. This study highlights the effectiveness of Prompt Engineering in guiding LLMs to generate safety reports, providing significant potential for complementing traditional safety assessment workflows while reducing time and resource burdens. These results strongly support the hypothesis that LLMs, when properly guided, can meet the rigorous expectations of aerospace safety standards.

6.1.3 Achieving Research Objectives

Each of the six research objectives was addressed through the findings, as outlined below:

1. **Development and Evaluation of Prompt Engineering for STPA Requirement Elicitation**: the study successfully developed a methodology leveraging Prompt Engineering to guide ChatGPT-4 in performing Phase 1 of STPA. The elicited requirements were validated against those of a real aerospace system, demonstrating comparable or superior quality. This achievement fulfills the objective of ensuring the relevance and accuracy of the method in a real-world aerospace context.
2. **Fine-Tuning LLM for Automated MoC Assignment**: a fine-tuned LLM model was developed and trained using a dataset of over 2,700 labeled requirements. The model achieved high accuracy in assigning MoCs, showcasing its ability to streamline compliance processes for aerospace defense systems.
3. **Benchmarking LLM-driven STPA Against Traditional Methods**: by comparing the quality of requirements generated through the LLM-driven STPA approach with those derived from a real-world aerospace system, the study demonstrated the efficiency and accuracy of the AI-assisted methodology. The

findings underscore the potential of LLMs to reduce manual effort while maintaining high standards of requirement elicitation.

4. **Assessment of MoC Assignment Accuracy**: the performance of the fine-tuned LLM model was benchmarked against that of domain experts. The significant gap in accuracy—80.18% for the LLM versus 6.07% for experts—highlights the transformative potential of AI in automating compliance tasks. This result validates the hypothesis that fine-tuned LLMs can meet or exceed expert-level accuracy in MoC assignments.
5. **Analysis of Limitations and Improvement Areas**: while the study highlights the efficacy of LLMs in STPA and MoC assignments, it also identifies limitations, such as challenges in hierarchical control structure modeling and the need for contextual information to enhance MoC assignments. Recommendations for integrating complementary tools like MBSE software and iterative prompt refinement address these limitations, ensuring continuous improvement of the methodology.
6. **Development of a Framework for Generating Comprehensive Safety Assessment Reports:** The study successfully developed and tested a framework for generating safety reports, such as Preliminary Hazard Analysis (PHA), using LLM-driven Prompt Engineering techniques. By applying GPT-4o to the BEF-1502 electronic fuze case study and leveraging MIL-STD-882E guidelines, the framework demonstrated its ability to produce structured, standards-compliant safety assessments. Expert evaluations validated the initial PHA as high quality, with subsequent iterations addressing all identified gaps to achieve full compliance. This accomplishment underscores the potential of LLMs to enhance traditional safety analysis workflows, delivering accurate and comprehensive outputs that align with aerospace industry standards and expert expectations. The findings fulfill the objective of creating a practical, AI-driven approach to safety report generation, highlighting its value as a complementary tool for system safety processes.

6.1.4 Significance of Results

The findings of this study have profound implications for aerospace defense systems engineering. By demonstrating that LLMs can automate STPA-based requirement elicitation

and MoC assignments with high accuracy, the research addresses a critical bottleneck in the field: the labor-intensive nature of traditional requirements engineering and compliance processes. The ability of LLMs to generate high-quality outputs consistently offers several key benefits:

1. **Enhanced efficiency**: the automation of STPA and MoC assignments significantly reduces the time and effort required for these tasks, allowing engineering teams to focus on higher-order design and analysis activities.
2. **Improved quality and consistency**: the LLM-generated requirements consistently met high-quality standards, reducing the variability often introduced by human subjectivity.
3. **Scalability**: the approach is scalable across complex systems, enabling efficient handling of the exponential growth in requirements and compliance tasks associated with advanced aerospace projects.
4. **Expert Collaboration**: LLMs complement expert judgment by acting as a productivity aid, enabling collaborative workflows where AI-generated outputs are reviewed and refined by domain experts.

A noteworthy observation emerged from the performance comparison between LLM-generated requirements and those derived from a real-world system. The former exhibited superior performance across 8 out of the 9 attributes under scrutiny, namely: Correctness, Completeness, Consistency, Clarity, Non-ambiguity, Connectivity, Testability, and Modifiability. The sole attribute where the real-world system requirements outperformed those generated by ChatGPT was Singularity.

It is intriguing that the Singularity attribute was the only attribute in which requirements from the real system outperformed those generated through the STPA + LLM approach. Some factors could explain this outcome, particularly considering the specific nature of the Singularity attribute.

Firstly, LLMs like ChatGPT often aim to provide comprehensive outputs that account for multiple scenarios or address different perspectives in a single statement. While this characteristic is beneficial for attributes like Completeness and Clarity, it can lead to requirements that inadvertently combine multiple ideas, subjects, or actions, violating the principle of Singularity.

In addition, unless explicitly guided by strict prompts emphasizing Singularity, LLMs may naturally generate compound or multi-faceted requirements. This reflects the model's design to offer rich, nuanced information rather than isolated, atomic ideas. But it also indicates a direction for improving the approach proposed by this research, adding guidelines for obtaining singular requirements through appropriate Prompt Engineering techniques adptation.

Real-world aerospace systems often follow strict guidelines and standardized templates for requirements, which enforce the decomposition of ideas into singular, testable statements. Such rigor is not inherently present in AI-generated outputs unless explicitly modeled into the prompts, which also reinforces the necessity of adjustments in the proposed Prompt Engineering approach.

Another possible cause for the observed outcome is that STPA focuses on identifying losses, hazards, and safety constraints at a high level before deriving detailed requirements. This hierarchical structure might lead to requirements that implicitly address multiple elements or relationships, making them less singular but still aligned with the system’s safety objectives.

Furthermore, LLMs might struggle to differentiate where one requirement should end and another should begin, especially when concepts are interrelated. This is particularly relevant in aerospace systems, where safety, functionality, and operational constraints are tightly intertwined. AI-generated requirements might embed implicit relationships or dependencies between subjects, verbs, or objects, which could cause them to appear as composite statements.

6.1.5 Addressing Limitations

The study acknowledges some limitations, including:

1. **Hierarchical Control Structure Modeling**: LLMs struggled with the second phase of STPA, necessitating the use of complementary tools like Capella’s STPA Viewpoint to model complex system architectures.
2. **Contextual Understanding for MoC Assignments**: The lack of contextual information in the requirements’ text and collaboration within a multidisciplinary team severely limited the experts’ performance in the comparative survey. Future work should explore methods to enhance contextual inputs for both AI and human evaluators.

3. **Error in Identifier Assignment**: GPT-4o exhibited difficulty in assigning unique identifiers to newly identified hazards, inadvertently reusing existing identifiers. This limitation underscores the need for additional measures to ensure proper hazard numbering, especially in iterative safety analyses where unique identifiers play a critical role in tracking hazards throughout the system development lifecycle.
4. **Dependence on Prompt Quality**: The quality and completeness of the safety reports generated by GPT-4o were heavily influenced by the clarity and specificity of the prompts provided. This highlights a broader limitation of LLM-driven methodologies, where the success of the output relies significantly on the expertise of the user in crafting effective prompts. Future work should explore more robust prompt design frameworks and automated prompt refinement techniques to mitigate this dependency.
5. **Human Oversight Requirement**: While GPT-4o demonstrated strong capabilities in generating safety assessments, expert review and iterative improvements were essential to achieve compliance with MIL-STD-882E standards. This emphasizes the necessity of human involvement to address gaps, validate outputs, and ensure the reliability of AI-assisted safety analysis. Incorporating automated validation mechanisms and tighter integration with domain-specific tools could reduce the burden on human reviewers while maintaining high standards of safety and compliance.

By identifying these limitations and proposing solutions, the research establishes a foundation for the ongoing refinement and development of AI-assisted methodologies in aerospace systems engineering. It also reinforces the relevance of keeping humans in the supervising loop of AI tools, especially when dealing with safety-critical subjects.

6.1.6 Research Originality, Generality and Utility

Enclosing the aspects of what characteristics a doctoral research must present, explored in Section 1.2– Justification, originality, generality, and utility must be discussed.

Originality: A critical hallmark of doctoral research is the demonstration of innovative thinking and novel methods. In this work, originality emerges from the introduction of new strategies to elicit requirements for aerospace systems, assign their Means

of Compliance (MoCs), and automate safety-report generation. While traditional approaches to systems engineering often rely on manual processes and domain-specific tools, the proposed framework leverages advanced techniques—such as large language models and fine-tuning methods—to streamline these tasks. By doing so, it addresses longstanding inefficiencies in aerospace development workflows. Moreover, the automation of safety-report generation represents a significant departure from conventional practices, offering a more dynamic, adaptive, and ultimately more efficient way of handling complex safety requirements. The work thus contributes a fresh perspective on how artificial intelligence can be harnessed to revolutionize critical facets of systems engineering, from early-stage design through final certification.

Generality: Beyond its immediate application in aerospace, the proposed approach has the potential for extensive cross-domain utilization. Many industries—automotive, pharmaceutical, and others with stringent regulatory requirements—could adopt the same underlying methods to enhance safety, compliance, and quality assurance processes. The multi-class classification strategy used for assigning MoCs, for example, can be readily adapted to various types of risk assessment challenges. Notably, fine-tuning large language models to classify and assign treatments to project risks offers a powerful illustration of this generalizability. By training such models with examples of identified risks and their respective mitigation strategies (e.g., acceptance, avoidance, transfer), project teams in any complex engineering domain could significantly reduce the time and effort spent on risk management. Similarly, the same methods can be deployed to determine contributing factors in incident or accident narratives, showcasing how the research's foundational principles transcend a single field or narrowly defined problem space. In essence, this dissertation's core ideas and techniques can serve as a blueprint for any sector that must systematically manage large volumes of documentation, requirements, and safety considerations within a tightly regulated environment.

Utility: A measure of a doctoral project's impact lies in how practically beneficial it is to the field and its stakeholders. Here, the utility of AI-driven tools—particularly those automating development and certification processes—cannot be overstated. The aerospace sector stands to gain substantial advantages from automated requirement elicitation, MoC assignment, and rapid safety analysis. Beyond mere time savings, these contributions also enhance the consistency and rigor of safety documentation, enabling more reliable compliance with regulatory standards. By reducing manual tasks and the potential for human

error, this approach not only accelerates certification timelines but also fosters a culture of proactive risk management, as stakeholders can devote more resources to higher-level design considerations. Consequently, the industrial implications are profound, with similar efficiencies readily transferred to other highly regulated sectors. The research's emphasis on applicability and effectiveness ensures that its benefits extend well beyond academic settings, offering a tangible, real-world impact that resonates with industry practitioners and certification authorities alike.

6.1.7 Broader Implications and Future Directions

The success of this research extends beyond the immediate context of aerospace defense systems. The demonstrated capabilities of LLMs in requirements engineering and compliance processes have potential applications across industries where safety and reliability are critical, such as automotive, healthcare, and energy systems. Future research could focus on:

1. Enhancing LLM architectures to address complex system modeling challenges.
2. Expanding datasets to include diverse aerospace systems and MoCs, improving model generalization.
3. Investigating collaborative AI-human workflows to maximize the benefits of automation while retaining expert oversight.
4. Explicitly Emphasize Singularity in Prompts: Ensure that prompts explicitly instruct the LLM to generate requirements addressing only one subject, verb, or object per statement.
5. Post-Processing of AI Outputs: Develop a refinement step to decompose composite requirements into singular, atomic statements before presenting them for evaluation.
6. Incorporate Feedback Loops: Use expert feedback to iteratively fine-tune the LLM, emphasizing the decomposition of multi-faceted statements.

6.2 Contributions

This doctoral research presents a dual impact, advancing both the academic understanding of LLMs and their integration with STPA and delivering practical technical

benefits tailored to the FAB. Below, the contributions are categorized into academic advancements and defense technical applications.

6.2.1 Academic Contributions

1. **Pioneering the Use of LLMs in STPA**: This study represents one of the first comprehensive integrations of LLMs, specifically ChatGPT, into the STPA framework. By automating Phase 1 of STPA through Prompt Engineering, the research demonstrates how LLMs can efficiently derive actionable safety requirements while maintaining rigor and traceability.
2. **Advancement of Prompt Engineering Methodology**: A refined methodology for Prompt Engineering was developed, emphasizing techniques like the "Tree-of-Thought" approach to improving the reasoning and outputs of LLMs. This contributes to the emerging academic field of Prompt Engineering by showcasing its application in safety-critical domains.
3. **Validation of LLM-Generated Requirements Against Real Systems**: This research compares AI-generated requirements to those of an operational aerospace system within the FAB, establishing a new benchmark for evaluating the relevance, accuracy, and quality of LLM-driven requirements elicitation.
4. **Quantitative Analysis of Requirement Quality Attributes**: The study evaluates requirements using nine critical quality attributes, highlighting LLMs' strengths and limitations in generating requirements. This detailed analysis provides a roadmap for improving AI models in engineering contexts.
5. **Development of a Fine-Tuned Model for MoC Assignment**: A fine-tuned version of ‘gpt-3.5-turbo’ was trained to assign Means of Compliance (MoCs) with an accuracy of 80.18%. This contribution introduces a novel use case for fine-tuning LLMs in aerospace certification processes.
6. **Comparison of AI and Human Performance in Safety-Critical Domains**: The research directly compared LLM-driven processes and expert-led approaches for both requirement elicitation and MoC assignment. These findings illuminate the complementary roles of AI and human expertise in engineering.
7. **Addressing Challenges in Hierarchical Control Structure (HCS) Modeling**: While acknowledging LLMs' limitations in modeling complex system

architectures, the research highlights how tools like the "STPA Viewpoint for Capella" can complement AI, suggesting pathways for hybrid methodologies in future studies.

8. **Framework for Automating Requirements Development**: By integrating STPA and LLMs, the research establishes a replicable framework for automating the elicitation and refinement of high-quality requirements, which can be adapted to other safety-critical domains.
9. **Application of LLMs in Safety Report Generation:** This study introduces an innovative framework for leveraging LLMs, specifically GPT-4o, to automate the generation of safety assessment reports, such as Preliminary Hazard Analyses (PHA). By integrating Prompt Engineering techniques with established safety standards like MIL-STD-882E, the research demonstrates how LLMs can produce structured, standards-compliant safety analyses. Validation by aerospace safety experts confirmed the high quality of the generated reports, with iterative improvements addressing identified gaps. This contribution showcases the potential of LLMs to complement traditional safety workflows, reducing manual effort while maintaining rigor and compliance in safety-critical domains.

6.2.2 Contributions to the FAB

1. **Streamlining Requirements Engineering Processes**: The research introduces a scalable and efficient methodology for generating system requirements directly from safety analyses. This capability reduces the manual workload on FAB engineering teams, freeing resources for other critical tasks.
2. **Enhanced Quality of Requirements for Aerospace Systems**: Requirements generated through the STPA + LLM approach demonstrated superior quality in eight out of nine evaluated attributes compared to real-world requirements. This improvement directly benefits the FAB by increasing the robustness and reliability of system designs.
3. **Automation of MoCs Assignments**: The fine-tuned LLM model for MoC assignment automates a traditionally labor-intensive process, ensuring consistency and reducing the risk of errors. This tool can accelerate the certification process for aerospace defense systems.

4. **Support for Ground Control Station (GCS) Development**: The study specifically focused on the GCS for a UCAV as a case study, producing AI-generated safety requirements that align with operational realities. These outputs can directly inform the design and safety assurance of GCSs in FAB's future UCAVs.
5. **Reduction of Expert Dependency in Certification Processes**: The methodology reduces dependence on a limited pool of certification experts by automating requirements elicitation and MoC assignment. This capability enhances the FAB's flexibility, especially in complex projects with tight timelines.
6. **Scalability Across Diverse Aerospace Projects**: The research methodology is adaptable to various aerospace projects, enabling its application to new systems or upgrades to existing platforms. This versatility supports the FAB's ongoing efforts to modernize its fleet and infrastructure.
7. **Enhancing Data Security Through Local LLM Deployment:** The study provides a practical framework for securely integrating LLMs into the FAB's workflows by emphasizing the deployment of open-source models, such as Llama and DeepSeek, on local intranets. This approach mitigates the risks associated with transmitting sensitive or classified information to third-party servers, ensuring data sovereignty while harnessing advanced AI capabilities. By leveraging open-source models that rival proprietary alternatives, the FAB can maintain strict control over its information while benefiting from state-of-the-art performance. Additionally, tools like Open WebUI offer intuitive interfaces for local LLM interaction, enabling customization, fine-tuning, and Retrieval-Augmented Generation (RAG) to meet the specific needs of aerospace systems engineering. This contribution equips the FAB with a secure, scalable, and adaptable solution for integrating generative AI technologies without compromising operational confidentiality.
8. **Data-Driven Decision Support**: The structured approach to eliciting and validating requirements provides a robust framework for data-driven decision-making in developing and certifying aerospace systems.
9. **Capacity Building in AI-Driven Engineering**: This work equips the FAB with a concrete example of how AI can be harnessed to enhance engineering processes. The methodology serves as a model for future applications of generative AI within

the organization.

10. **Knowledge Transfer Through Expert Engagement**: The research fosters knowledge transfer by involving FAB experts in the validation and evaluation phases, ensuring that the benefits of AI integration are well understood and effectively implemented within the organization.
11. **Strengthened Collaboration with Academic and Industrial Partners**: The project highlights the potential for academic research to address real-world operational challenges, strengthening the FAB's collaboration with research institutions like the ITA and industrial partners in aerospace development.

This research advances the state of the art in aerospace systems engineering and positions the FAB at the forefront of innovation in integrating AI into safety-critical domains. By bridging academic theory with practical application, the study lays the groundwork for more efficient, reliable, and scalable engineering processes within the FAB and the broader aerospace industry. The methodology and findings are poised to influence future developments in AI-driven systems engineering, ensuring the FAB remains a leader in adopting cutting-edge technologies for national defense.

6.3 Publications

This Section lists the articles developed during this doctoral program. The publications are divided into three categories: strongly related, weakly related, and remotely related.

6.3.1 Strongly Related Publications

1. MOREIRA, G.; PLEFFKEN, D.; SANTOS, W.; CERQUEIRA, C.; GOTELIP, M. (2024), Using LLMs to Automate Means of Compliance Assignment in Aerospace Defense Systems. Accepted at the Journal of Aerospace Information Systems (AIAA). Pre-print available at < https://www.techrxiv.org/users/853226/articles/1238870 >.
2. MOREIRA, G.; PEREIRA, A.; NABARRETE, A.; SANTOS, W. (2024), Addressing Gearbox Health Monitoring Challenges for Helicopters: A Machine Learning Approach. Accepted at the Anais da Academia Brasileira de Ciências.
3. MOREIRA, G.; PLEFFKEN, D.; SANTOS, W.; CERQUEIRA, C. (2024), Enhancing Brazilian Aerospace Systems Lifecycle Directive, XXVI SIGE – Simpósio de Aplicações Operacionais em Áreas de Defesa, São José dos Campos, Brazil.

4. MOREIRA, G.; PLEFFKEN, D.; SANTOS, W.; CERQUEIRA, C. (2022), STPA Analysis over the Earlier Phases of Brazilian Aerospace Products Life Cycle Using OPM, 13th ICMAE – International Conference on Mechanical and Aerospace Engineering, Bratislava, Slovakia. DOI: https://doi.org/10.1109/ ICMAE56000.2022.9852838. Available at <https://ieeexplore.ieee.org/ document/9852838 >.
5. MOREIRA, G.; LIMONGE, W.; LAHOZ, C.; SANTOS, W.; CERQUEIRA, C. (2021), STPA Analysis Over the Earlier Phases of Military Products Life Cycle, XXIII SIGE – Simpósio de Aplicações Operacionais em Áreas de Defesa, São José dos Campos, Brazil.
6. PLEFFKEN, D.; MOREIRA, G.; BOURGUIGNON, V.; CERQUEIRA, C. (2024), Automating eVTOL Airworthiness Certification Using MBSE and Large Language Models: A Framework for Regulatory Alignment. Pre-print available at <https://www.techrxiv.org/ users/ 859370/articles/1242623 >.
7. PLEFFKEN, D.; MOREIRA, G.; CERQUEIRA, C. (2023), Enhancing Spaceworthiness Process Based on Certification Procedures Applied in KC-390 and Gripen F-39. XXV SIGE – Simpósio de Aplicações Operacionais em Áreas de Defesa, São José dos Campos, Brazil.
8. PLEFFKEN, D.; MOREIRA, G.; CERQUEIRA, C. (2022), Aplicando Engenharia de Sistema Baseada em Modelos para suportar Projetos Aeroespaciais Militares no Brasil. XXIV SIGE – Simpósio de Aplicações Operacionais em Áreas de Defesa, São José dos Campos, Brazil.

6.3.2 Weakly Related Publications

1. MOREIRA, G.; SANTOS, W.; CERQUEIRA, C. (2024), Spaceworthiness: the Future of Space Products Safety. Revista Observatorio de la Economía Latinoamericana. DOI: https://doi.org/10.55905/oelv22n7-108. Available at <https://ojs. observatoriolatinoamericano. com/ojs/index.php/olel/article/view/5722>.
2. SILVA, C.; SOUZA, M.; MOREIRA, G. (2024), A proposal for a minimum viable mission assurance (MVMA) process for small and medium satellites. International Journal of Agile Systems and Management. DOI: http://dx.doi.org/10.1504/ IJASM.2024.142132. Available at < https://www.inderscience.com/offers.php ?id=142132 >.

3. SILVA, C.; MOREIRA, G.; SOUZA, M. (2023), Aplicação da uma Estratégia de Harmonização de Means of Compliance (Métodos de Verificação) a um Estudo de Caso da Área Espacial. Anais do Simpósio Acadêmico de Engenharia de Produção (SAEPRO) da EEL-USP. DOI: http://dx.doi.org/10.29327/1289625.4-1. Available at <https://www.even3.com.br/anais/saepro2023/636783-aplicacao-da-uma-estrategia-de-harmonizacao-de-means-of-compliance-(metodos-de-verificacao)-a-um-estudo-de-caso-d/>.
4. SILVA, C.; MOREIRA, G.; SOUZA, M. (2023), Um Método para a Avaliação de uma Proposta de Aperfeiçoamento de Processos da Garantia do Produto Espacial. Anais do Simpósio Acadêmico de Engenharia de Produção (SAEPRO) da EEL-USP. DOI: http://dx.doi.org/10.29327/1289625.4-2. Available at <https://www.even3.com.br/ anais/saepro2023/636803-um-metodo-para-a-avaliacao-de-uma-proposta-de-aperfeicoamento -de-processos-da-garantia-do-produto-espacial/>.
5. PLEFFKEN, D.; BOURGUIGNON, V.; MOREIRA, G.; CERQUEIRA, C. (2023), Acceptance of Residual Risk in a Brazilian Military Aeronautical Project Through the Application of a Method. 33rd European Safety and Reliability Conference. Available at <https://easychair.org/publications/preprint/7g91>.
6. SILVA, C.; MOREIRA, G.; SOUZA, M. (2022), Study of Practices and Criteria Used in the Military Aviation Certification to Improve the Satellite Product Assurance. International Journal of Advanced Engineering Research and Science. DOI: http://dx.doi.org/10.22161/ijaers.910.31. Available at <https://ijaers.com/detail/study-of-practices-and-criteria-used-in-the-military-aviation-certification-to-improve-the-satellite-product-assurance/>.

6.3.3 Remotely Related Publications

1. MOREIRA, G.; VENTURINI, M.; SANTOS, W.; CERQUEIRA, C. (2022), A Problemática das Queimadas na Região Amazônica Brasileira sob a Luz de uma Abordagem Multimetodológica de Estruturação de Problemas, LIV SBPO – Simpósio Brasileiro de Pesquisa Operacional, Juiz de Fora, Brazil. Available at <https://proceedings.science/sbpo/sbpo-2022/trabalhos/a-problematica-das-queimadas-na-regiao-amazonica-brasileira-sob-a-luz-de-uma-abo?lang=pt-br>

References

ABDOLLAHI, M. *et al*. **Hardware design and verification with large language models**: a literature survey, challenges, and open issues. 04 November 2024. Available at: https://www.preprints.org/manuscript/202411.0156/v1. Access on: 10 nov. 2024.

ABDULKHALEQ, A. *et al*. **Using STPA in compliance with ISO 26262 for developing a safe architecture for fully automated vehicles**. Ithaca: ArXiv Operational Status, 2017. https://doi.org/10.48550/arXiv.1703.03657.

ABDULKHALEQ, A.; WAGNER, S. **XSTAMPP**: an eXtensible STAMP platform as tool support for safety engineering. Stuttgart. [S.l.]. 2015. Available at: https://core.ac.uk/download/pdf/147543131.pdf. Access on: 12 nov. 2024.

ACEMOGLU, D.; RESTREPO, P. Automation and new tasks: how technology displaces and reinstates labor. **Journal of Economic Perspectives**, v. 33, n. 2, p. 3–30, 2019. https://doi.org/10.1257/JEP.33.2.3.

ADALOGLOU, N. **Why multi-head self attention works**: math, intuitions and 10+1 hidden insights. 25 march 2021. Available at: https://theaisummer.com/self-attention/. Access on: 14 nov. 2024.

AJITH, S.; RITHANI, M.; SYAMDEV, R. S. Identifying and mitigating gender bias in language models: a fair machine learning approach. *In*: 2023 SEVENTH INTERNATIONAL CONFERENCE ON IMAGE INFORMATION PROCESSING (ICIIP). 7., 2023. Solan, India. **Proceedings** […]. Solan, India: IEEE, 22-24 November, 2023. Available at: https://ieeexplore.ieee.org/document/10537623?utm_source=chatgpt.com. Access on: 19 nov. 2024.

ALDIN, N. B.; ALDIN, S. S. A. B. Accuracy comparison of different batch size for a supervised machine learning task with image classification. *In*: 2022 9th INTERNATIONAL CONFERENCE ON ELECTRICAL AND ELECTRONICS ENGINEERING (ICEEE). 9., 2022. Alanya, Turkey. **Proceedings** […]. Alanya, Turkey: IEEE, 29-31 March 2022. Available at: https://ieeexplore.ieee.org/document/9772551. Access on: 20 nov. 2024.

ALEMZADEH, H. **Data-driven resiliency assessment of medical cyber-physical systems**. Urbana: University of Illinois, 2016.

ALEXANDER, I. F.; STEVENS, R. **Writing better requirements**. London: Pearson Education, 2002.

ALMEIDA, J.; FONSECA, A. Improving railway safety by properly modeling “complex socio-technical systems. *In*: INTERNATIONAL RAILWAY SAFETY COUNCIL. 2014. Berlin. **Proceedings** […]. Berlin 12-17 October 2014. Available at: https://international-railway-safety-council.com/wp-content/uploads/2017/09/almeida-fonseca-improving-railway-safety-by-properly-modeling-complex-socio-technical-systems.pdf. Acess on: 12 nov. 2024.

ALMEIDA, L. **Integridade de poços offshore**: análise de segurança da fase de produção usando system theoretic process analysis (STPA). 2024. 92 f. Dissertação (Mestrado em Engenharia Naval) – Escola Politécnica da Universidade de São Paulo, São Paulo, 2024.

ALOISIO, D. C. **Lessons from systems engineering failures**: determining why systems fail, the state of systems engineering education, and building an evidence-based network to help systems engineers identify and fix problems on complex projects. 2019. 281 f. Thesis (Doctor of Philosophy) - Faculty of Purdue University, West Lafayette, Indiana, 2019.

ARNOLD, M. *et al*. FactSheets: increasing trust in AI services through supplier's declarations of conformity. **IBM Journal of Research and Development**, v. 63, n. 4/5, July-Sept., 2018.

ATHERTON, K. Mass-market military drones have changed the way wars are fought. MIT Technology Review, 30 January 2023. Available at: https://www.technologyreview.com/2023/01/30/1067348/mass-market-military-drones-have-changed-the-way-wars-are-fought/. Access on: 17 nov. 2024.

AUTOR, D. H.; LEVY, F.; MURNANE, R. J. The skill content of recent technological change: an empirical exploration. **The Quarterly Journal of Economics**, v. 118, n. 4, p. 1279–1333, 2003. https://doi.org/10.1162/003355303322552801.

BAEK, T. J. **Open WebUI Github**. Github. 2025. Available at: https://github.com/open-webui/open-webui. Access on: 17 jan. 2025.

BAU, D. *et al*. Understanding the role of individual units in a deep neural network. **Proceedings of the National Academy of Sciences**, v. 117, n. 48, p. 30071-30078, dez. 2020.

BENDER, E. M.; GEBRU, T.; MCMILLAN-MAJOR, A.; SHMITCHELL, S. On the dangers of stochastic parrots: can language models be too big? *In*: FAccT 2021 - PROCEEDINGS OF THE 2021 ACM CONFERENCE ON FAIRNESS, ACCOUNTABILITY, AND TRANSPARENCY. 2021. Canada. **Proceedings** […]. Canada: Association for Computing Machinery, Inc, 3 March. 2021. Available at: https://dl.acm.org/doi/pdf/10.1145/3442188.3445922. Access on: 13 nov. 2024.

BERTALANFFY, L. Von. General systems theory as integrating factor in contemporary science: akten des xiv. **Internationalen Kongresses für Philosophie**, v. 2, p. 335-340, 1968.

BEVARA, R. V. K. *et al*. Bias analysis in language models using an association test and prompt engineering. *In*:2023 IEEE 23rd INTERNATIONAL CONFERENCE ON SOFTWARE QUALITY, RELIABILITY, AND SECURITY COMPANION (QRS-C). 23., 2023. Chiang Mai, Thailand. **Proceedings** […]. Chiang Mai, Thailand: IEEE, 22-26 October 2023. Available at: https://ieeexplore.ieee.org/document/10430006. Access on: 19 nov. 2024.

BEYOND MBSE. **Creating use case roles**. 2024. Available at: https://www.beyondmbse.com/use-case-roles/. Access on: 10 may 2024.

BISHOP, C. **Pattern recognition and machine learning**. New York: Springer, 2006.

BLANCHARD, B. S.; FABRYCKY, W. J. **Systems engineering and analysis**. London: Pearson, 2014.

BONGIRWAR, R. Leveraging systems theoretic process analysis (STPA) for efficient ISO 26262 Compliance. **SAE Technical Papers**, n. 2021, 6 abr. 2021.

BOYER, R. R. *et al*. Materials considerations for aerospace applications. **Mrs Bulletin**, v. 40,

n. 12, p. 1055-1066, 2015.

BRASIL. Comando da Aeronáutica. Estado-Maior da Aeronáutica. Portaria EMAER n° 129/GC4, de 5 de março de 2007. Aprova a Diretriz que dispõe sobre Ciclo de Vida de Sistemas e Materiais da Aeronáutica: DCA 400-6. **Boletim do Comando da Aeronáutica**, Rio de Janeiro, n. 47, 9 mar. 2007.

BRASIL. Ministério da Defesa. Estado-Maior Conjunto das Forças Armadas. **MD40-M-01**: Manual de boas práticas para a gestão do ciclo de vida de sistemas de defesa. Brasília, DF, 2020. Available at: https://www.gov.br/caslode/pt-br/arquivos/gestao-do-ciclo-de-vida-de-sistemas-de-defesa/manual_md_40_m_01_13jan2020.pdf. Access on: 17 nov. 2024.

BROWN, T. B. *et al*. **Language models are few-shot learners**. Ithaca: ArXiv Operational Status, 28 maio 2020. https://doi.org/10.48550/arXiv.2005.14165.

BRUNDAGE, M. Taking superintelligence seriously. **Futures**, v. 72, p. 32–35, 2015. https://doi.org/10.1016/j.futures.2015.07.009.

BRUNTON, S. L. *et al*. Data-driven aerospace engineering: reframing the industry with machine learning. **AIAA Journal**, v. 59, n. 8, p. 2820-2847, 2021.

CAMBRIA, E. *et al*. **XAI meets LLMs**: a survey of the relation between explainable AI and large language models. Ithaca: ArXiv Operational Status, 21 jul. 2024. https://doi.org/10.48550/arXiv.2407.15248.

CAPES. **Parecer Sucupira**. 1965.

CHANG, G. H. *et al*. Assessment of knee pain from MR imaging using a convolutional Siamese network. **Eur Radiol**, v. 30, n. 6, p. 3538-48, 1 jun. 2020.

CHARALAMPIDOU, S.; ZELESKIDIS, A.; DOKAS, I. M. Hazard analysis in the era of AI: Assessing the usefulness of ChatGPT4 in STPA hazard analysis. **Safety Science**, v. 178, p. 106608, out. 2024.

CHATILA, R.; FIRTH-BUTTERFLIED, K.; HAVENS, J. C.; KARACHALIOS, K. The IEEE global initiative for ethical considerations in artificial intelligence and autonomous systems [Standards]. **IEEE Robotics & Automation Magazine**, v. 24, n. 1, p. 110–110, 2017. https://doi.org/10.1109/MRA.2017.2670225.

CHECKLAND, P. **Systems thinking, systems practice**. London: John Wiley, 1999.

CHEN, Y.; CHEN, H.; SU, S. Fine-tuning large language models in education. *In*: 2023 13th INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY IN MEDICINE AND EDUCATION (ITME). 13., 2023. Wuyishan, China. **Proceedings** [...]. Wuyishan, China: IEEE, 24-26 November 2024. Available at: https://ieeexplore.ieee.org/document/10505547. Access on: 20 nov. 2024.

CHENG, M.; DURMUS, E.; JURAFSKY, D. **Marked personas**: using natural language prompts to measure stereotypes in language models. Ithaca: ArXiv Operational Status, 29 maio 2023. https://doi.org/10.48550/arXiv.2305.18189.

CREVIER, D. **AI**: the tumultuous history of the search for artificial intelligence. New York:

BasicBooks, 1993.

CROSS, L. *et al*. **Hypothetical minds**: scaffolding theory of mind for multi-agent tasks with large language models. Ithaca: ArXiv Operational Status, 9 jul. 2024. https://doi.org/10.48550/arXiv.2407.07086.

DE KLEER, J. The fifth generation: artificial intelligence and Japan's computer challenge to the world. **Artificial Intelligence**, v. 22, n. 2, p. 222–226, 1 mar. 1984.

DEEPSEEK-AI; LIU, A.; FENG, B.; XUE, B.; WANG, B.; WU, B.; LU, C.; ZHAO, C.; DENG, C.; ZHANG, C.; RUAN, C.; DAI, D.; GUO, D.; YANG, D.; CHEN, D.; JI, D.; LI, E.; LIN, F.; DAI, F.; PAN, Z. (2024). **DeepSeek-V3 technical report**. Ithaca: ArXiv Operational Status, 2024. https://arxiv.org/html/2412.19437v1.

DEEPSEEK-AI; GUO, D.; YANG, D.; ZHANG, H.; SONG, J.; ZHANG, R.; XU, R.; ZHU, Q.; MA, S.; WANG, P.; BI, X.; ZHANG, X.; YU, X.; WU, Y.; WU, Z. F.; GOU, Z.; SHAO, Z.; LI, Z.; GAO, Z.; ZHANG, Z. **DeepSeek-R1**: incentivizing reasoning capability in LLMs via reinforcement learning. Ithaca: ArXiv Operational Status, 2025. Arxiv. https://arxiv.org/html/2501.12948v1.

DE FLORIO, F. **Airworthiness**. Third ed. [S.l.]: Elsevier, 2016.

DENG, J. *et al*. ImageNet: a large-scale hierarchical image database. *In*: 2009 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. 2009. Miami, FL,USA. **Proceedings** […]. Miami, FL,USA: IEEE, 20-25 June 2009, p. 248-255.

DEPAUW, T. C. *et al*. Development of a commercial airplane certification AI digital assistant. *In*: AIAA SCITECH 2024 FORUM. 2024. Orlando, FL. **Proceedings** […]. Orlando, FL, 8-12 January 2024.

DEVLIN, J. *et al*. **BERT**: pre-training of deep bidirectional transformers for language understanding. Ithaca: ArXiv Operational Status, 10 out. 2018. https://doi.org/10.48550/arXiv.1810.04805.

DIAMANDIS, P. H.; KOTLER, S. **The future is faster than you think**: How converging technologies are transforming business, industries, and our lives. New York: Simon & Schuster, 2020.

DODGE, J. *et al*. **Fine-tuning pretrained language models**: weight initializations, data orders, and early stopping. Ithaca: ArXiv Operational Status, 14 fev. 2020. https://doi.org/10.48550/arXiv.2002.06305.

DOGANIS, R. **Flying off course**: the economics of international airlines. Boca Raton: Routledge, 2013.

DONG, Y. Deep learning-based opponent aircraft attitude detection in autonomous air combat. **Journal of Aerospace Information Systems**, v. 16, n. 4, p. 162–167, 2019.

DONG, Y. *et al*. Knowledge-driven accurate opponent trajectory prediction for gun-dominated autonomous air combat. **Journal of Aerospace Information Systems**, v. 20, n. 5, p. 251-263, 2023.

DORI, D. **Object-process methodology**. [S.l.]: Springer, 2002.

ELKINS, J. G.; SOOD, R.; RUMPF, C. Bridging reinforcement learning and online learning for spacecraft attitude control. **Journal of Aerospace Information Systems**, v. 19, n. 1, p. 62-69, jan. 2022.

ELM, W. C. *et al*. Integrating cognitive systems engineering throughout the systems engineering process. **Journal of Cognitive Engineering and Decision Making**, v. 2, n. 3, p. 249-273, 1 set. 2008.

ERICSON, C. **Hazard analysis techniques for system safety**. 2nd Edition ed. [S.l: s.n.].

ESTEFAN, J. A. **Survey of model-based systems engineering (MBSE) methodologies**. [S.l: s.n.], 2008.

ESTEVA, A. *et al*. Dermatologist-level classification of skin cancer with deep neural networks. **Nature**, v. 542, n. 7639, p. 115-118, 25 jan. 2017.

FANG, X. *et al*. **Bias of AI-generated content**: an examination of news produced by large language models. Ithaca: ArXiv Operational Status, 18 set. 2023. https://doi.org/10.48550/arXiv.2309.09825.

FENG, S. *et al*. From pretraining data to language models to downstream tasks: tracking the trails of political biases leading to unfair NLP models. *In*: PROCEEDINGS OF THE 61st ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS (Volume 1: Long Papers). 61., 2023. Stroudsburg, PA, USA. **Proceedings** [...]. Stroudsburg, PA, USA: Association for Computational Linguistics, 2023. Available at: https://aclanthology.org/2023.acl-long.656/. Access on: 19 nov. 2024.

FERRARA, E. Fairness and bias in artificial intelligence: a brief survey of sources, impacts, and mitigation strategies. **Sci**, v. 6, n. 1, p. 3, 26 dez. 2023.

FINANCIAL TIMES. **Generative AI exists because of the transformer**. 2023. Available at: https://ig.ft.com/generative-ai/. Access on: 13 nov. 2024.

FLEMING, C. H. *et al*. Safety assurance in NextGen and complex transportation systems. **Safety Science**, v. 55, p. 173-187, 1 jun. 2013.

FLORIDI, L. **The logic of information** : a theory of philosophy as conceptual design. Oxford: Oxford University Press, 2019. Available at: https://books.google.com/books/about/The_Logic_of_Information.html?hl=pt-BR&id=nYZUEAAAQBAJ. Access on: 19 nov. 2024.

FORD, M. **Rise of the robots**: basic books. [S.l], 2015.

FORRESTER, J. W. **Urban dynamics**. [S.l.]: MIT Press, 1976, p. 285.

FORSBERG, K.; MOOZ, H. The relationship of systems engineering to the project cycle. **EMJ - Engineering Management Journal**, v. 4, n. 3, p. 36-43, 1992.

FREY, C. B.; OSBORNE´S, M. A. **The future of employment**: how susceptible are jobs to computerisation? Oxford: University of Oxford, 2013. Available at: https://oms-

www.files.svdcdn.com/production/downloads/academic/The_Future_of_Employment.pdf. Access on: 17 dez. 2024.

FRIEDENTHAL, S.; MOORE, A.; STEINER, R. **A practical guide to sysML**: the systems modeling language, 2rd Edition. [S.l.], 2011, p. 1-615.

GALLINA, B.; ANDREWS, A. Deriving verification-related means of compliance for a model-based testing process. *In*: 2016 IEEE/AIAA 35th DIGITAL AVIONICS SYSTEMS CONFERENCE (DASC). 35., 2016. Sacramento, CA, USA. **Proceedings** […]. Sacramento, CA, USA: IEEE, 25-29 September, 2016.

GAO, J. *et al*. Multiple moving vehicles tracking algorithm with attention mechanism and motion model. **Electronics**, v. 13, n. 1, p. 242, jan. 2024.

GARCIA, A. B.; BABICEANU, R. F.; SEKER, R. Artificial intelligence and machine learning approaches for aviation cybersecurity: an overview. *In*: 2021 INTEGRATED COMMUNICATIONS NAVIGATION AND SURVEILLANCE CONFERENCE (ICNS). 2021. Dulles, VA, USA. **Proceedings** […]. Dulles, VA, USA: IEEE, 19-23 April 2021. Available at: https://ieeexplore.ieee.org/document/9441594. Access on: 14 nov. 2024.

GEORGIOU, I. The idea of emergent property. **Journal of the Operational Research Society**, v. 54, n. 3, p. 239-247, 2003.

GICHOYA, J. W. *et al*. AI recognition of patient race in medical imaging: a modelling study. **The Lancet Digital Health**, v. 4, n. 6, p. e406-e414, 1 jun. 2022.

GOERTZEL, B. Artificial general intelligence: concept, state of the art, and future prospects. **Journal of Artificial General Intelligence**, v. 5, n. 1, p. 1-48, 1 dez. 2014.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. [S.l.]: MIT Press, 2016.

GREENE, N. **Technical bias in neural networks**. Ann Arbor: Utica University, 2022.

HAGOS, D. H.; BATTLE, R.; RAWAT, D. B. Recent advances in generative AI and large language models: current status, challenges, and perspectives. **IEEE Transactions on Artificial Intelligence**, v. 12, p. 1-21, 2024.

HALLIGAN, R. J.; CPENG, F. A. Requirements quality metrics: the basis of informed requirements engineering management. *In*: PRESENTED AT THE 1993 COMPLEX SYSTEMS ENGINEERING SYNTHESIS AND ASSESSMENT TECHNOLOGY WORKSHOP. 1993. Calvados, MD, USA. **Proceedings** […]. Calvados, MD, USA 1993. Available at: https://www.ppi-int.com/wp-content/uploads/2019/05/Requirements-Quality-Metrics-Paper-with-Addendum-PPA-005330-9-140710.pdf. Access on: 21 nov. 2024.

HE, Y. *et al*. **HI-TOM**: a benchmark for evaluating higher-order theory of mind reasoning in large language models. Ithaca: ArXiv Operational Status, 25 out. 2023. https://doi.org/10.48550/arXiv.2310.16755.

HEAVEN, W. **The open-source AI boom is built on big tech's handouts**: how long will it last? MIT Technology Review. 12 May 2023. Available at: https://www.technologyreview.com/2023/05/12/1072950/open-source-ai-google-openai-eleuther-meta/. Access on: 19 nov. 2024.

HOHMA, E. **A practical organizational framework for AI accountability**. Futurium. 2 October 2023. Available at: https://futurium.ec.europa.eu/en/european-ai-alliance/best-practices/practical-organizational-framework-ai-accountability. Access on: 19 nov. 2024.

HUGGINGFACE. **Training and fine-tuning**. 2024. Available at: https://huggingface.co/docs/transformers/training. Access on: 19 nov. 2024.

INSAURRALDE, C. C. Artificial intelligence engineering for aerospace applications. 2020 AIAA/IEEE 39th DIGITAL AVIONICS SYSTEMS CONFERENCE (DASC). 39., 2020. San Antonio, TX, USA. **Proceedings** […]. San Antonio, TX, USA: IEEE, 11 out. 2020. Available at: https://ieeexplore.ieee.org/document/9256770. Access on: 14 nov. 2024.

INTERNATIONAL ENERGY AGENCY (IEA). **Data centres and data transmission networks**. July 11 2023. Available at: https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks. Access on: 14 sep. 2024.

ISHIMATSU, T. *et al*. Modeling and hazard analysis using stpa. *In*: PROCEEDINGS OF THE 4th IAASS CONFERENCE, MAKING SAFETY MATTER. 4., 2010. Huntsville, Alabama. **Proceedings** […]. Huntsville, Alabama, 19-21 May 2010.

ISHIMATSU, T. *et al*. Hazard analysis of complex spacecraft using systems-theoretic process analysis. **JSR Spacecraft Rockets**, v. 51, n. 2, p. 509-522, mar. 2014.

ISO. **ISO 19450:2024**: Automation systems and integration — object-process methodology. 2024. Available at: https://www.iso.org/standard/84612.html. Access on: 20 nov. 2024.

ISO. **ISO/IEC/IEEE 15288:2023**: Systems and software engineering — system life cycle processes. 2023. Available at: https://www.iso.org/standard/81702.html. Access on: 11 nov. 2024.

JAERYANG BAEK. **Open WebUI Github**. Available at: https://github.com/open-webui/open-webui. Access on: 23 jan. 2025.

JAIN, A. K. Data clustering: 50 years beyond K-means. **Pattern Recognition Letters**, v. 31, n. 8, p. 651–666, 1 jun. 2010.

JORDAN, J. The future of unmanned combat aerial vehicles: an analysis using the three horizons framework. **Futures**, v. 134, p. 102848, 1 dez. 2021.

JURAFSKY, D.; MARTIN, J. H. **Speech and language processing**. 3rd Edition. [S.l: Springer, 2024.

KAMRUZZAMAN, M.; SHOVON, MD. M. I.; KIM, G. L. **Investigating subtler biases in LLMs**: ageism, beauty, institutional, and nationality bias in generative models. 16 set. 2023.

KARRAS, T. *et al*. **Analyzing and improving the image quality of StyleGAN**. Ithaca: ArXiv Operational Status, 2020. https://doi.org/10.48550/arXiv.1912.04958.

KATZ, D.; KHAN, R. LOUIS. **The social psychology of organizations**. Canada: John Wiley & Sonsp. 1978, p. 838.

KOHEN, H.; DORI, D. Improving conceptual modeling with object-process methodology

stereotypes. **Applied Sciences**, v. 11, n. 5, p. 2301, 5 mar. 2021.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature** v. 521, n. 7553, p. 436–444, 27 maio 2015.

LEVESON, N. G. **Engineering a safer world**: systems thinking applied to safety. New York: The MIT Press, 2016.

LEVESON, N. G. **System safety engineering**: back to the future. Massachusetts: Massachusetts Institute of Technology, 2002. Available at: http://sunnyday.mit.edu/book2.pdf. Access on: 12 nov. 2024.

LEVESON, N.; THOMAS, J. **STPA Handbook**. [S.l.], mar. 2018.

LI, L. *et al*. Digital twin in aerospace industry: a gentle introduction. **IEEE Access**, v. 10, p. 9543–9562, 2021.

LIAO, B.; VARGAS, D. V. **Extending token computation for LLM reasoning**. Ithaca: ArXiv Operational Status, 21 mar. 2024. https://doi.org/10.48550/arXiv.2403.14932.

LIAO, Q. V.; VAUGHAN, J. W. **AI transparency in the age of LLMs**: a human-centered research roadmap. Ithaca: ArXiv Operational Status, 2 jun. 2023. https://doi.org/10.48550/arXiv.2306.01941.

LU, S. *et al*. Are emergent abilities in large language models just in-context learning? *In*: PROCEEDINGS OF THE 62nd ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS (Volume 1: Long Papers). 62., 2024. Bangkok, Thailand. **Proceedings** [...]. Bangkok, Thailand: Association for Computational Linguistics, 2024, p. 5098–5139. Available at: https://aclanthology.org/2024.acl-long.279/. Access on: 13 nov. 2024.

LUO, Q.; PUETT, M. J.; SMITH, M. D. A **Perspectival mirror of the elephant**: investigating language bias on Google, ChatGPT, YouTube, and Wikipedia. 28 mar. 2023.

MACIASZEK, L. **Requirements analysis and system design**: developing information systems with UML. New York: Addison-Wesley, 2001.

MAHAJAN, H. S.; BRADLEY, T.; PASRICHA, S. Application of systems theoretic process analysis to a lane keeping assist system. **Reliability Engineering & System Safety**, v. 167, p. 177-183, 1 nov. 2017.

MARKOVIC, M. Rise of the robot lawyers. **Arizona Law Review**, v. 61, 2019.

MARTINS, L. E. G.; GORSCHEK, T. Requirements engineering for safety-critical systems: overview and challenges. **IEEE Software**, v. 34, n. 4, p. 49–57, 2017.

MAVROGIORGOS, K. *et al*. Bias in machine learning: a literature review. **Applied Sciences**, v. 14, n. 19, p. 8860, 2 out. 2024.

MCCARTHY, J. *et al*. A Proposal for the dartmouth summer research project on artificial intelligence, August 31, 1955. **AI Magazine**, v. 27, n. 4, p. 12–12, 15 dez. 2006.

MCDONALD, R. A. *et al*. Future aircraft concepts and design methods. **The Aeronautical Journal**, v. 126, n. 1295, p. 92–124, 2022.

MEGHA, C. R.; MADHURA, A.; SNEHA, Y. S. Cognitive computing and its applications. *In*: 2017 INTERNATIONAL CONFERENCE ON ENERGY, COMMUNICATION, DATA ANALYTICS AND SOFT COMPUTING, ICECDS. 2017. Chennai, India. **Proceedings** […]. Chennai, India: IEEE, 01-02 August 2017, p. 1168-1172.

MEHRABI, N. *et al*. A survey on bias and fairness in machine learning. **ACM Computing Surveys**, v. 54, n. 6, p. 1–35, 31 jul. 2022.

MIKOLOV, T. *et al*. Distributed representations of words and phrases and their compositionality. *In*: BURGES, C. J. *et al*. (eds.), **Advances in neural information processing systems**. [S.l.], 16 out. 2013.

MILAN, F. F.; BASSIRI TABRIZI, A. Armed, unmanned, and in high demand: the drivers behind combat drones proliferation in the Middle East. *In*: ROSSITER, A. (ed.), **Robotics, autonomous systems and contemporary international security**. [S.l.]: Routledge, 2020. p. 40-60.

MITCHELL, T. **Machine learning**. [S.l.]: McGraw-Hill Pub. Co., 1997.

MOIR, I.; SEABRIDGE, A. **Design and development of aircraft systems**. [S.l.]: John Wiley & Sons, 2012. v. 67.

MORALES, S.; CLARISÓ, R.; CABOT, J. Automating bias testing of LLMs. *In*: 2023 38th IEEE/ACM INTERNATIONAL CONFERENCE ON AUTOMATED SOFTWARE ENGINEERING (ASE). 38., 2023. Luxembourg. **Proceedings** […]. Luxembourg: IEEE, 11-15 Septembr 2023. Available at: https://ieeexplore.ieee.org/document/10298519. Access on: 19 nov. 2024.

NASA. **Probabilistic risk assessment procedures guide for NASA managers and practitioners**. 2011. Washington: NASA. Available at: https://ntrs.nasa.gov/api/citations/20120001369/downloads/20120001369.pdf. Access on: 12 nov. 2024.

NATH, B. *et al*. A comparative study of model variations: english-nepali language pair. *In*: 2024 OPJU INTERNATIONAL TECHNOLOGY CONFERENCE (OTCON) ON SMART COMPUTING FOR INNOVATION AND ADVANCEMENT IN INDUSTRY 4.0. 2024. Raigarh, India. **Proceedings** […]. Raigarh, India: IEEE, 5-7 June 2024. Available at: https://ieeexplore.ieee.org/document/10687932. Access on: 20 nov. 2024.

NATURE MACHINE INTELLIGENCE. Achieving net zero emissions with machine learning: the challenge ahead. **Nature Machine Intelligence**, v. 4, n. 8, p. 661–662, 2022. https://doi.org/10.1038/s42256-022-00529-w.

NECULA, S.-C.; DUMITRIU, F.; GREAVU-ȘERBAN, V. A systematic literature review on using natural language processing in software requirements engineering. **Electronics**, v. 13, n. 11, p. 2055, 24 maio 2024.

NGAI, E. W. T. *et al*. The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. **Decision Support Systems**,

v. 50, n. 3, p. 559–569, 1 fev. 2011.

NIKOLOPOULOU, K. **Easy introduction to reinforcement learning**. 2023. Available at: https://www.scribbr.com/ai-tools/reinforcement-learning/. Access on: 13 nov. 2024.

NUSEIBEH, B.; EASTERBROOK, S. Requirements engineering: a roadmap. *In*: PROCEEDINGS OF THE CONFERENCE ON THE FUTURE OF SOFTWARE ENGINEERING, ICSE. 2000. New York. **Proceedings** […]. New York: Association for Computing Machinery, 1 maio 2000, p. 35-46.

OECD. **Accountability (OECD AI Principle) - OECD.AI**. 2024. Available at: https://oecd.ai/en/dashboards/ai-principles/P9. Access on: 19 nov. 2024.

OPCLOUD. **OPCloud**. 2022. Available at: https://opcloud.systems/. Access on: 20 nov. 2024.

OPENAI. **Fine-tuning - OpenAI API**. 2024a. Available at: https://platform.openai.com/docs/guides/fine-tuning. Access on: 20 nov. 2024.

OPENAI. **Privacy policy**. 2024b. Available at: https://openai.com/en-GB/policies/row-privacy-policy/. Access on: 19 nov. 2024.

OPENAI. **Hello GPT-4o**. May 13 2024c. Available at: https://openai.com/index/hello-gpt-4o/. Access on: 20 dez. 2024.

PATTERSON, D.; GONZALEZ, J.; LE, Q.; LIANG, C.; MUNGUIA, L.-M.; ROTHCHILD, D.; SO, D.; TEXIER, M.; DEAN, J. **Carbon emissions and large neural network training**. Ithaca: ArXiv Operational Status, 5 fev. 2021. https://arxiv.org/abs/2104.10350v3.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. GloVe: global vectors for word representation. *In*: PROCEEDINGS OF THE 2014 CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE. 2024. Doha, Qatar. **Proceedings** […]. Doha, Qatar: Association for Computational Linguistics, 2014, p. 1532-1543.

PERIAUX, J. *et al.* Multidisciplinary design optimisation and robust design in Aerospace systems. *In*: **Evolutionary optimization and game strategies for advanced multi-disciplinary design**: applications to aeronautics and UAV design. 2015, p. 53-68.

PICANÇO, P.; FARIA, M.; SILVA, C. **PDQSAT university space mission**: hazard analysis and risk assessment of the operation phase. São José dos Campos, Brazil, Sept. 2024. Available at: https://www1.univap.br/la-stamp-workshop/assets/presentations/LASW-Pedro_Pican%C3%A7o.pdf. Access on: 19 nov. 2024.

PRESSMAN, R. **Software engineering**: a practitioner's approach. 7th Edition. New York: McGraw Hill, 2010.

PRESSMAN, S. M. *et al.* AI and ethics: a systematic review of the ethical considerations of large language model use in surgery research. **Healthcare**, v. 12, n. 8, p. 825, 13 abr. 2024.

PRINCE, S. J. D. **Understanding deep learning**. [S.l.]: MIT Press, 2024.

QIU, Z.; ZHAO, H.; WANG, S. Applications and challenges of artificial intelligence in

aerospace engineering. *In*: 2023 6th INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND BIG DATA (ICAIBD). 6., 2023. Chengdu, China. **Proceedings** […]. Chengdu, China: IEEE 26-29 May 2023. Available at: https://ieeexplore.ieee.org/document/10206205. Access on: 14 nov. 2024.

RADFORD, A. *et al*. **Language models are unsupervised multitask learners**. [S.l: s.n.]. 2019. Available at: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf. Access on: 13 nov. 2024.

RAY, A. T.; FISCHER, O. P.; COLE, B. F.; WHITE, R. Aerobert-classifier: classification of aerospace requirements using bert. **Aerospace**, v. 10, n. 3, p. 279, 2023.

RISING, J. M.; LEVESON, N. G. Systems-theoretic process analysis of space launch vehicles. **Journal of Space Safety Engineering**, v. 5, n. 3-4, p. 153-183, 1 Sept. 2018.

RTCA. **DO-178C**: Software considerations in airborne systems and equipment certification. Washington, DC, USA, 2011.

RUIZ, A. *et al*. Challenges for an open and evolutionary approach to safety assurance and certification of safety-critical systems. *In*: 2011 FIRST INTERNATIONAL WORKSHOP ON SOFTWARE CERTIFICATION. 2011. Hiroshima, Japan. **Proceedings** […]. Hiroshima, Japan: IEEE, 29 November 2011.

RUSSELL, S.; NORVIG, P. **Artificial intelligence a modern approach**. 4th Edition. [S.l.]: Pearson, 2020.

SAAKSVUORI, A.; IMMONEN, A. **Product lifecycle management systems**. [S.l.]: Springer, 2008.

SADRAEY, M. H. **Ground control stations**: unmanned aircraft design. [S.l.]: Springer, 2024. p. 171-189.

SAE INTERNATIONAL. **ARP 4754A**: Guidelines for development of civil aircraft and systems. Warrendale, PA, USA: SAE, 2010.

SAHOO, P. *et al*. **A systematic survey of prompt engineering in large language models**: Techniques and Applications. 5 fev. 2024.

SCHMITT, B. **From cooperation to integration**: defence and aerospace industries in Europe. Paris: Institute for Security Studies, Western European Union Paris, 2000.

SCHWARTZ, R.; DODGE, J.; SMITH, N. A.; ETZIONI, O. Green AI. **Communications of the ACM**, v. 63, n. 12, p. 54–63, 2020. https://doi.org/10.1145/3381831.

SENGE, P. M. The Fifth Discipline [Electronic version]. **Measuring Business Excellence**, v. 1, n. 3, p. 46–51, 1997.

SILVA, C. M. Z.; DE SOUZA, G. M.; DE OLIVEIRA; SOUZA, M. L. Proposals for a space product assurance process improvement based on an aeronautical process. **IEEE Aerospace Conference Proceedings**, v. 2018- March, p. 1–9, 25 jun. 2018.

SILVA, R. **System and requirements definition processes for the life cycle of space**

systems at Brazilian Air Force. São José dos Campos: [s.n.]. Available at:http://www.bdita.bibl.ita.br/tesesdigitais/78215.pdf. Access on: 24 fev. 2025.

SILVER, D. *et al*. Mastering the game of go with deep neural networks and tree search. **Nature**, v. 529, n. 7587, p. 484-489, 27 jan. 2016.

SOMMERVILLE, I. **Software engineering**. 10th Edition. New York: Pearson, 2015.

STERMAN, J. **Business dynamics-systems thinking and modeling for a complex world**. London: McGraw-Hill, 2011. v. 53.

STOCK, A. Achieving net zero emissions with machine learning: the challenge ahead. **Nature Machine Intelligence**, v. 4, n. 8, p. 661–662, 2022. https://doi.org/10.1038/s42256-022-00529-w.

STOLLENWERK, F. **Adaptive fine-tuning of transformer-based language models for named entity recognition**. Ithaca: ArXiv Operational Status, 5 fev. 2022. https://doi.org/10.48550/arXiv.2202.02617.

STRUBELL, E.; GANESH, A.; MCCALLUM, A. Energy and policy considerations for deep learning in NLP. *In*: PROCEEDINGS OF THE 57th ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, 2019. 57., Florence, Italy. **Proceedings** […]. Florence, Italy, July 2019, p. 3645–3650. https://doi.org/10.18653/V1/P19-1355.

SUTTHITHATIP, S. *et al*. Explainable AI in Aerospace for enhanced system performance. 2021 IEEE/AIAA 40th DIGITAL AVIONICS SYSTEMS CONFERENCE (DASC). 40., 2021. San Antonio, TX, USA. **Proceedings** […]. San Antonio, TX, USA: IEEE, 03-07 October 2021. Available at: https://ieeexplore.ieee.org/document/9594488. Access on: 14 nov. 2024.

SUTTON, R. S.; BARTO, A. G. **Reinforcement learning**: an introduction. 2nd Edition. [S.l.]: MIT Press, 2015.

SUZUKI, K.; MATSUZAWA, T. Model soups for various training and validation data. **AI**, v. 3, n. 4, p. 796-808, 28 Sept. 2022.

TABIBU, S.; VINOD, P. K.; JAWAHAR, C. V. Pan-renal cell carcinoma classification and survival prediction from histopathology images using deep learning. **Sci Rep**, v. 9, n. 1, p. 10509, 1 dez. 2019.

TARAMSARI, H. B. *et al*. Identification of variables impacting cascading failures in aerospace systems: a natural language processing approach. *In*: SALADO, A., VALERDI, R., STEINER, R., HEAD, L. (eds), **The Proceedings of the 2024 Conference on Systems Engineering Research. CSER 2024**. Conference on Systems Engineering Research Series. Springer, Cham. https://doi.org/10.1007/978-3-031-62554-1_26.

TEGMARK, M. **Life 3.0**. 2017. Available at: https://books.google.com/books/about/Life_3_0.html?hl=pt-BR&id=3_otDwAAQBAJ. Access on: 17 oct. 2024.

THOMAS, J. P. **Extending and automating a systems-theoretic hazard analysis for**

requirements generation and analysis. Massachusetts: Massachusetts Institute of Technology, 2013.

THOMAS, J.; SUO, D. STPA-based method to identify and control feature interactions in large complex systems. **Procedia Engineering**, v. 128, p. 12–14, 1 jan. 2015.

THOMPSON, K. D. **How the drone war in Ukraine is transforming conflict**. Council on Foreign Relations. 16 January 2024. Available at: https://www.cfr.org/article/how-drone-war-ukraine-transforming-conflict. Access on: 17 nov. 2024.

TONDJI, Y.; GHAZI, G.; MIHAELA BOTEZ, R. Neural networks and support vector regression for the CRJ-700 longitudinal dynamics modeling. **Journal of Aerospace Information Systems**, p. 1-16, 2024.

TRAD, F.; CHEHAB, A. Prompt engineering or fine-tuning? A case study on phishing detection with large language models. **Machine Learning and Knowledge Extraction**, v. 6, n. 1, p. 367-384, 6 Febr. 2024.

TRAN, P. N. *et al*. An interactive conflict solver for learning air traffic conflict resolutions. **Journal of Aerospace Information Systems**, v. 17, n. 6, p. 271-277, June 2020.

TURING, A. M. **Computing machinery and intelligence**: parsing the turing test. Dordrecht: Springer Netherlands, 1950. p. 23-65.

UMAR, M. A.; LANO, K. Advances in automated support for requirements engineering: a systematic literature review. **Requirements Engineering**, p. 1-31, 2024.

UNITED STATES. Department of Defense. **MIL-HDBK-516C**: AIRWORTHINEss certification criteria. [s.l: s.n.]. Available at: http://www.everyspec.com. Access on: 12 jun. 2024.

UNITED STATES. Department of Defense. **MIL-STD-882E**: System safety. [s.l: s.n.]. 11 May 2012. Available at: https://www.nde-ed.org/NDEEngineering/SafeDesign/MIL-STD-882E.pdf. Access on: 12 jun. 2024.

UNITED STATES. Department of Transportation. Federal Aviation Administration. **AC 23.2010-1:** FAA accepted means of compliance process for 14 CFR Part 23. Washington, DC: FAA, 2017. Available at:https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_23_2010-1.pdf. Access on: 12 nov. 2024.

UNITED STATES. Department of Transportation. Federal Aviation Administration. **AC 25.1309-1B**: System design and analysis. Washington, DC: FAA, 2024. Available at: https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_25.1309-1B.pdf. Access on: 12 nov. 2024.

UNITED STATES. Department of Transportation. Federal Aviation Administration. **Order 8110.4C Chg 7**: Type certification. Washington, DC: FAA, 2023. Available at: https://www.faa.gov/documentLibrary/media/Order/Order_8110.4C_CHG_7.pdf. Access on: 12 nov. 2024.

VASWANI, A. *et al*. Attention is all you need. *In*: CONFERENCE ON NEURAL

INFORMATION PROCESSING SYSTEMS (NeurIPS). 2017. Long Beach. **Proceedings** [...]. Long Beach, Dez. 2017. Available at: https://user.phil.hhu.de/~cwurm/wp-content/uploads/2020/01/7181-attention-is-all-you-need.pdf. Access on: 13 nov. 2024.

WALDEN, D. *et al*. **Systems engineering handbook**. New York: John Wiley & Sons, 2015.

WANG, H.; QIU, F. AI adoption and labor cost stickiness: based on natural language and machine learning. **Information Technology and Management**, p. 1-22, 10 ago. 2023.

WANG, X.; LIU, X.-Q. Potential and limitations of ChatGPT and generative artificial intelligence in medical safety education. **World Journal of Clinical Cases**, v. 11, n. 32, p. 7935–7939, 16 nov. 2023.

WEBB, A. **The big nine**: how the tech titans and their thinking machines could warp humanity (1st ed.). PublicAffairs, 2019.

WILLEMINK, M. J. *et al.* Preparing medical imaging data for machine learning. **Radiology**, v. 295, n. 1, p. 4-15, 2020.

WORLD ECONOMIC FORUM. **The future of jobs report 2023**. 2023. Available at: www.weforum.org. Access on: 19 nov. 2024.

YAO, Y.; KOLLER, A. **Predicting generalization performance with correctness discriminators**. Ithaca: ArXiv Operational Status, 15 nov. 2023. https://doi.org/10.48550/arXiv.2311.09422.

ZDRAVKOVIĆ, M.; PANETTO, H.; WEICHHART, G. AI-enabled enterprise information systems for manufacturing. **Enterprise Information Systems**, v. 16, n. 4, p. 668-720, 3 abr. 2022.

ZHANG, Y. *et al*. Leveraging biases in large language models: “bias-kNN” for effective few-shot learning. *In*: ICASSP 2024 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING (ICASSP). 2024. Seoul, Korea, Republic. **Proceedings** […]. Seoul, Korea, Republic: IEEE, 14-19 April 2024. Available at: https://ieeexplore.ieee.org/document/10447730. Access on: 19 nov. 2024.

ZHAO, L.; GAO, W.; FANG, J. Optimizing large language models on multi-core CPUs: a case study of the BERT model. **Applied Sciences**, v. 14, n. 6, p. 2364, 11 March 2024.

ZOU, T.; SUN, K. Application and prospect of artificial intelligence in aircraft design. *In*: 2021 INTERNATIONAL CONFERENCE ON NETWORKING SYSTEMS OF AI (INSAI). 2021. Shanghai, China. **Proceedings** […]. Shanghai, China: IEEE, 19-20 November 2021. Available at: https://ieeexplore.ieee.org/document/9757942. Access on: 14 nov. 2024.

Appendix A – Prompts Used in This Research

A.1 Prompt to Perform Phase 1 of STPA by ToT Approach

You are an expert in safety-critical systems analysis and System-Theoretic Process Analysis (STPA). Your task is to assist in conducting Phase 1 of STPA for a UCAV (Unmanned Combat Air Vehicle) for the Brazilian Air Force (FAB), capable of operating in cooperation with other aircraft from FAB's fleet.

To accomplish this task you shall identify:

1. Losses: Undesired outcomes that the stakeholders wish to prevent.

2. Hazards: System-level states or conditions that, in combination with specific environmental conditions, can lead to a loss.

3. Safety Constraints: Actions, conditions, or states that must be maintained to prevent hazards and mitigate losses.

To ensure systematic and comprehensive outputs, use a Tree-of-Thought approach to explore the relationships between these components. Break down your reasoning into structured steps, considering various scenarios, environmental factors, and interactions.

Step 1: System Context and Boundaries

Describe the system, its operational purpose, key components, and boundaries, as well as its interactions with the external environment. Provide information about stakeholders (e.g., users, operators, customers) and the system's goals. If you do not specify, I will assume a general safety-critical system.

Output format for Step 1:

- System description
- Stakeholders and their objectives
- System boundaries

Step 2: Identifying Losses

Based on the system description, brainstorm potential six to nine losses. Losses are outcomes that stakeholders consider unacceptable, such as harm to people, damage to property, mission failure, or loss of sensitive information. For each loss:

- Specify its nature and its relevance to stakeholders.

- Consider different stakeholder perspectives and prioritize the losses accordingly.

Output format for Step 2:

- Loss ID (e.g., L-1, L-2)

- Description of the loss

- Stakeholders impacted

Step 3: Identifying Hazards

Brainstorm at least two potential hazards for each identified loss. A hazard is a system state or condition that, under worst-case environmental conditions, could lead to a loss. Use the following guiding questions:

- What system states or behaviors could lead to the identified loss?

- What environmental factors or operational conditions could exacerbate the risk?

- Are there interactions between components that could create unsafe conditions?

Provide traceability by linking each hazard to its associated losses.

Output format for Step 3:

- Hazard ID (e.g., H-1, H-2)

- Description of the hazard

- Link to losses (e.g., L-1, L-3)

Step 4: Identifying Safety Constraints

For each identified hazard, propose safety constraints. Safety constraints define the conditions or system behaviors required to prevent the hazard and mitigate the associated losses. Use the following structure:

- Invert the hazard statement to formulate the constraint.

- Ensure constraints address specific system states, operational conditions, or behaviors.

- Consider preventive and mitigative actions.

Output format for Step 4:

- Constraint ID (e.g., SC-1, SC-2)

- Description of the safety constraint

- Link to hazards (e.g., H-1, H-3)

Step 5: Iterative Refinement

Revisit and refine your results to ensure completeness.

- Check for additional losses, hazards, or constraints overlooked during the analysis.

- Consolidate overlapping or redundant items.

- Ensure traceability between losses, hazards, and constraints.

Final Deliverable: Provide the complete list of losses, hazards, and safety constraints in a tabular format for clarity and traceability.

A.2 Prompt to generate the Fuze BEF-1502 PHA

You are an aerospace systems engineering expert in systems safety assessment. I want you to perform a Preliminary Hazard Analysis (PHA) for the BEF-1502 Electronic Fuze system, adhering strictly to the methodology outlined in MIL-STD-882E and structured like the provided example PHA.

Your analysis should identify hazards, assess their initial risks, and propose potential mitigation measures. You can find attached to this prompt the following files that will help you in your task:

1) BEF-1502 Technical Specification.

2) A presentation about the Fuze System.

3) Schematic view of electronic Fuze system.

4) What is expected to a PHA to be.

5) Example output format Instructions:

a) System Overview: Begin by considering the system components (e.g., FZU-1502 and FMU-1502), energy sources, interfaces, controls, and operational environments as described in the BEF-1502 Technical Specification.

b) Hazard Identification: Use the MIL-STD-882E framework to identify hazards stemming from: System component; Ordnance and hazardous materials; Interfaces and controls, including software and hardware; Environmental factors and operational constraints; and Human factors engineering and potential operator errors.

c) Hazard Analysis: For each hazard, document: Function or system element causing the hazard; Specific failure mode leading to the hazard; Associated mission phase; Potential severity and likelihood of occurrence.

d) Risk Assessment: Evaluate the severity and probability of each hazard using MIL-STD-882E's Risk Assessment Codes (RACs).

e) Risk Mitigation: Propose mitigation measures in line with the system safety design order of precedence: Eliminate the hazard; Incorporate safety features to reduce risk; Implement warning devices; Develop special procedures or training to manage residual risks.

f) Additional Guidelines: Use a step-by-step reasoning approach (Chain-of-Thought) for thorough and logical hazard identification and risk assessment. Refer to similar systems or legacy data for context (Few-Shot Prompting). Ensure alignment with safety-critical standards using references from MIL-STD-882E. Present the PHA in tabular format, modeled after the provided example, with columns for Function ID, Function, Failure Mode, Hazard Description, Mission Phase, and Severity.

Generate your analysis in English with academic precision.

Appendix B – Relevant Documentation Used in this Work

B.1 IFI Authorization to the Data Used to Train and Validate the Research Hypothesis

MINISTÉRIO DA DEFESA
COMANDO DA AERONÁUTICA
INSTITUTO DE FOMENTO E COORDENAÇÃO INDUSTRIAL

Ofício nº 439/CDR-AT/983 São José dos Campos, 23 de abril de 2024.
Protocolo COMAER nº 67770.001107/2024-19

Do Assessor Técnico
À Pró-Reitora de Administração do Instituto Tecnológico de Aeronáutica

Assunto: Apoio no fornecimento de dados para projeto de pesquisa no ITA/PPGAO.

Referência: 1. Of nº 468/IP/1816, do(a) ITA ao(à) IFI.

1. Trata o presente expediente de resposta ao Ofício da referência que trata sobre pesquisa relacionada à automação de procedimentos de verificação de sistemas aeroespaciais, conduzida pelo Oficial-Aluno Ten Cel Av GUILHERME MICHELI BEDINI MOREIRA.

2. Sobre o assunto, incumbiu-me o Diretor do IFI de informar que estão autorizados os acessos solicitados, resguardado o compromisso de salvaguarda de informações sensíveis que eventualmente possam constar dos dados acessados.

3. Por fim, para coordenações, coloco-me à disposição no telefone (12) 3947-7114 e e-mail cdrat.ifi@fab.mil.br.

GUSTAVO BORGES BASILIO Ten Cel Av
Assessor Técnico

MARECHAL DO AR CASIMIRO MONTENEGRO 120 ANOS

Assinado digitalmente por GUSTAVO BORGES BASILIO
ESTE DOCUMENTO DEVE SER AUTENTICADO NO PORTAL https://adoc.fab.mil.br/adoc,
informando o código: RVHBQZH3.MZVZ3X3S.EXC3MOKY.MC6EGURZ

B.2 Mac Jee Authorization to Use Their Data

+55 12 2012-4103

MACJEE TECNOLOGIA

MACJEE

EQUIPAER INDÚSTRIA AERONÁUTICA

www.macjee.com.br

Paraibuna, SP, 21 de novembro de 2024 Carta DINOV 026/24

Ao Senhor
Ten Cel Guilherme Micheli Bedini MOREIRA
Aluno de Doutorado do Instituto Tecnológico de Aeronáutica - ITA
Praça Marechal Eduardo Gomes, 50
Vila das Acácias – São José dos Campos/SP
CEP: 12.228-900

Assunto: Autorização da Mac Jee para fazer menção ao documento PHA da espoleta eletrônica BEF-1502

Prezado Ten Cel Moreira,

Ao cordial e respeitosamente cumprimentar V.Sra, passo a tratar sobre a declaração abaixo exposta.

Em nome da Mac Jee, declaro que autorizamos o envio do documento *Preliminary Hazard Analysis* (PHA) da Espoleta Eletrônica BEF-1502 feito pelo senhor, com informações enviadas pela Mac Jee, como parte do seu doutorado em engenharia pelo ITA, aos especialistas de Safety Assessment do IFI.

A empresa entende ser um trabalho de cunho acadêmico e que faz uso de técnicas avançadas de Inteligência Artificial, e que é benéfico para o progresso das técnicas ligadas à certificação aeroespacial.

O envio de documentos ou informações para membros outros que não aqueles mencionados acima deve ser objeto de nova consulta à empresa.

Por fim, me coloco ao inteiro dispor do senhor para esclarecimentos gerais que se façam necessários, através do telefone: +55 12 99706-3924 e e-mail: danilo.miranda@macjee.com.br

Sem mais, tomamos da oportunidade de agradecer e reiterar nossos melhores cumprimentos.

Cordialmente,

danilo.f.miranda

Danilo Miranda
Diretor de Inovação
Grupo Mac Jee

B.3 Survey with 26 Requirements Evaluated by Experts

Evaluation of Aerospace Defense Systems Requirements

First and foremost, thank you very much for accepting the invitation to participate in this research. Your assistance is invaluable!

This questionnaire is part of an ongoing doctoral research at the Aeronautics Institute of Technology. In this study, ChatGPT was used to conduct an STPA (System Theoretic Process Analysis) to derive requirements for a fictitious UCAV (Unmanned Combat Air Vehicle) system.

The aim of this evaluation is to establish a comparison benchmark between the requirements of a real UAV (Unmanned Air Vehicle) operating within the Brazilian Air Force and those of a fictitious UCAV generated with the support of Artificial Intelligence.

The requirements used in this experiment are System-Level Requirements related to the safety of a GCS (Ground Control System), a system present and necessary in UAVs and UCAVs.

Your task is to evaluate the quality metrics of these requirements. The quality characteristics to observe include: **Correctness**, **Completeness**, **Consistency**, **Clarity**, **Non-ambiguity**, **Connection**, **Singularity**, **Testability**, **Modifiability**, and **Feasibility**.

Below, you can find a brief explanation about the meaning of each quality characteristic that should be present in a well-formed requirement. Your role, therefore, is to assess the presence or absence of these characteristics in each requirement.

Correctness refers to an absence of errors of fact in the statement of requirement.

Completeness requires that the requirement contain all of the information necessary, including constraints and conditions, to enable the requirement to be implemented such that the need will be satisfied.

Consistency requires that a requirement not be in conflict with any other requirement, nor with any element of its own structure.

Clarity requires that the requirement be readily understandable without semantic analysis.

Non-Ambiguity requires that there be only one semantic interpretation of the requirement.

Singularity refers to the attribute whereby a requirement cannot sensibly be expressed as two or more requirements having different subjects, verbs and/or objects.

Testability refers to the existence of a finite and objective process with which to verify that the requirement has been satisfied.

Modifiability requires that necessary changes to a requirement can be made completely and consistently.

Feasibility requires that a requirement be able to be satisfied

- within natural physical constraints;
- within the state-of-the-art as it applies to the project; and
- within all other absolute constraints applying to the project.

List of requirements on the survey:

R01: The GCS (Ground Control System) design shall aim for full compliance with the most severe safety requirements within specified environmental conditions.

R02: The GCS design shall consider engineering safety issues.

R03: The GCS shall make efforts to eliminate or control all identified hazards to an acceptable level of risk.

R04: The GCS design shall consider operational safety issues.

R05: The GCS operation shall request comprehensive training programs and implement standard operating procedures to minimize the potential for human error.

R06: The GCS design shall implement reliable power supply solutions, including backup systems, to prevent operational interruptions due to power loss.

R07: The GCS shall undergo verification through safety engineering analysis.

R08: The GCS design shall implement robust authentication and access control mechanisms to ensure that access to UAV control systems is restricted to authorized personnel only.

R09: The criteria in accepting the residual risks and establishing necessary corrective actions will be defined, in conjunction with normative safety documents.

R10: The GCS shall eliminate or mitigate System catastrophic events.

R11: The GCS design shall contribute to the safety integration effort by controlling and mitigating hardware and software hazards that may affect UAV System integrity.

R12: The GCS design shall ensure software reliability through rigorous testing, regular updates, and prompt resolution of identified issues.

R13: The GCS safety analysis shall establish and maintain a robust risk management framework that ensures all risks are assessed, managed, and documented appropriately.

R14: The GCS safety analysis shall incorporate comprehensive hazard identification and mitigation processes, including regular reviews and updates, to address all identified safety risks.

R15: The risk assessment model is intended to be used along with the GCS and UAV System Hazard analyses.

R16: The GCS design shall incorporate the accumulated experience of verified safety engineering criteria in complex UAV/UCAV integrated systems.

R17: The GCS design shall implement measures to protect against adverse environmental conditions and ensure safe operational continuity.

R18: The GCS shall ensure the highest degree of safety, consistent with its operational requirements, throughout its life cycle.

R19: The resolutions will be reflected in the Safety Assessments Reports in terms of severity and probability of occurrence.

R20: The GCS design shall adhere to all relevant safety standards and undergo regular audits and updates to ensure compliance.

R21: The resolution of identified land and air hazards in the GCS constituents will be defined and documented.

R22: The GCS design shall implement robust cybersecurity measures to protect against cyber-attacks and vulnerabilities.

R23: The GCS safety policy shall conduct thorough and accurate safety assessments and maintain comprehensive reporting practices to ensure all safety-related information is documented and reviewed.

R24: The GCS shall maintain robust communication channels and implement backup systems to ensure continuous and reliable communication with the UAV/UCAV.

R25: The GCS operation shall request the development and enforcement standard operating procedures and the provision of ongoing training to ensure personnel are adequately prepared for their roles.

R26: The GCS design shall ensure the reliability of all system components through rigorous testing, regular maintenance, and prompt resolution of technical issues.

B.4 Survey of Experts' Performance in Assigning MoCs to Aerospace Systems Requirements

First and foremost, thank you very much for accepting the invitation to participate in this research, which will integrate into my doctoral thesis and serve as scientific support to validate the approach proposed here. I have endeavored to recruit the support of professionals who are among the best experts in Brazil in the field of development and/or certification of aerospace defense products.

This survey aims to estimate the accuracy in defining Means of Compliance (MoC) for aerospace system requirements by experts in the field. Support material (available at https://drive.google.com/file/d/1awwM5C6ZHzYJEmlwshAiPMWTnvpnjMIL/view) has been prepared, describing the definition of each MoC, and should be your sole reference source for MoCs assignments to the requirements. I encourage the reading of this material even if you feel confident about the full meaning of each MoC. The possible MoCs are: **Compliance Statement**, **Design Review**, **Calculations/Analysis**, **Safety Assessment**, **Laboratory Test**, **Ground Test**, **Flight Test**, **Simulation**, **Inspection**/**Audit**, and **Equipment Qualification**.

R1. Need for the Launch System operator interface to be able to allow the operator to check the current mode and status of the Launch System.
R2. Location of the Operator Interface in such a way as to maintain visibility for an operator of whether the Launch System is in the loaded state or unloaded state.
R3. Fuze Functional and Mechanical Performance Characteristics.
R4. Entry into Reset mode: from any operating mode, including Non-Operational mode, and at the moment the subsystem is energized.

R5. If the acceleration mechanism is kept activated during the delay time of the 1st stage Timing Device, it must lock the acceleration mechanism in this position and activate the 2nd stage Timing Device.

R6. Mechanical interface with the acceleration engine, the cruise engine exhaust pipe and the elevators, with the appropriate safety factors.

R7. Umbilical break detection.

R8. Clearly identified connectors with references of the cables.

R9. Operate in Rain and Blowing Rain: Procedure I – Method 506.4 of MIL-STD-810F

R10. Multiple reboot requests before declaring a C/I-BIT failure from the missile to the Aircraft.

R11. Enable fuse on entry to terminal phase.

R12. BIT errors are stored in MFS and can be download.

R13. Missile Fuse Shock and Vibration.

R14. Standby missile power consumption < 120W.

R15. Default Missile Fail state from reported C-BIT FAIL status.

R16. Maximum dimensions compatibility.

R17. Bomb Points for fixing mechanical reinforcement.

R18. Bomb Operation / human-machine interface.

R19. Tests only through its interface connector and the interface with the elevators, without the need for a subsystem disassembly operation.

R20. Protection against unintentional release of the elevator lock up to 1.0 s ± 0.1 s after the umbilical rupture.

Appendix C – OPM Model Description

C.1 Proposed Approach OPM Model Description

This subsection provides a formal language description (OPL) of the OPM model prepared to describe the proposed approach.

SD

1. Defense Aerospace Project Organizations is a physical object.
2. Requirements of Defense Aerospace Project Organizations can be MoCs assigned or MoCs not assigned.
3. Dataset is stateful.
4. AI Staff is a physical object.
5. LLM is stateful.
6. Defense Aerospace Project Organizations exhibits Requirements.
7. Moc Generator exhibits Assigning Mocs.
8. Assigning Mocs of Moc Generator changes Requirements of Defense Aerospace Project Organizations from MoCs not assigned to MoCs assigned.
9. AI Staff handles Assigning Mocs of Moc Generator.
10. Assigning Mocs of Moc Generator requires Dataset, LLM, and Moc Generator.

Assigning Mocs in-zoomed

1. Assigning Mocs of Moc Generator from SD zooms in SD1 into Pre Processing Dataset, Uploading Processed Dataset, Fine-tuning Model, Validating Trained Model, Testing Trained Model, and Assigning Mocs With Trained Model, which occur in that time sequence.
2. Requirements of Defense Aerospace Project Organizations can be MoCs assigned or MoCs not assigned.
3. Defense Aerospace Project Organizations is a physical object.
4. Dataset can be processed, unprocessed or uploaded.
5. AI Staff is a physical object.
6. LLM can be fine-tuned or pre-trained.
7. Defense Aerospace Project Organizations exhibits Requirements.
8. Moc Generator exhibits Assigning Mocs.

9. AI Staff handles Assigning Mocs of Moc Generator.
10. Assigning Mocs of Moc Generator requires Dataset and Moc Generator.
11. Pre Processing Dataset changes Dataset from unprocessed to processed.
12. Uploading Processed Dataset changes Dataset from processed to uploaded.
13. Fine-tuning Model changes LLM from pre-trained to fine-tuned.
14. Validating Trained Model requires LLM.
15. Testing Trained Model requires LLM.
16. Assigning Mocs With Trained Model changes Requirements of Defense Aerospace Project Organizations from MoCs not assigned to MoCs assigned.
17. Assigning Mocs With Trained Model requires LLM.

OPL spec.

1. Defense Aerospace Project Organizations is a physical object.
2. Requirements of Defense Aerospace Project Organizations can be MoCs assigned or MoCs not assigned.
3. Dataset is stateful.
4. AI Staff is a physical object.
5. LLM is stateful.
6. Defense Aerospace Project Organizations exhibits Requirements.
7. Moc Generator exhibits Assigning Mocs.
8. Assigning Mocs of Moc Generator changes Requirements of Defense Aerospace Project Organizations from MoCs not assigned to MoCs assigned.
9. AI Staff handles Assigning Mocs of Moc Generator.
10. Assigning Mocs of Moc Generator requires Dataset, LLM, and Moc Generator.
11. Assigning Mocs of Moc Generator from SD zooms in SD1 into Pre Processing Dataset, Uploading Processed Dataset, Fine-tuning Model, Validating Trained Model, Testing Trained Model, and Assigning Mocs With Trained Model, which occur in that time sequence.
12. Requirements of Defense Aerospace Project Organizations can be MoCs assigned or MoCs not assigned.
13. Defense Aerospace Project Organizations is a physical object.
14. Dataset can be processed, unprocessed or uploaded.
15. AI Staff is a physical object.
16. LLM can be fine-tuned or pre-trained.

17. Defense Aerospace Project Organizations exhibits Requirements.
18. Moc Generator exhibits Assigning Mocs.
19. AI Staff handles Assigning Mocs of Moc Generator.
20. Assigning Mocs of Moc Generator requires Dataset and Moc Generator.
21. Pre Processing Dataset changes Dataset from unprocessed to processed.
22. Uploading Processed Dataset changes Dataset from processed to uploaded.
23. Fine-tuning Model changes LLM from pre-trained to fine-tuned.
24. Validating Trained Model requires LLM.
25. Testing Trained Model requires LLM.
26. Assigning Mocs With Trained Model changes Requirements of Defense Aerospace Project Organizations from MoCs not assigned to MoCs assigned.
27. Assigning Mocs With Trained Model requires LLM.

Appendix D – Pseudocodes

This subsection presents the scripts pseudocodes used to perform Dataset processing; its division into Training, Validation and Test Data; upload it to the OpenAI training platform via API; create the ‘gpt-3.5-turbo’ model training job to perform Fine-Tuning; and measure the accuracy of the model when subjected to validation and test data. For this endeavor it was used the Anaconda Navigator version 2.5.3, Spyder IDE version 5.4.3, with Python 3.11.7 64-bit, in an 11th Gen Intel(R) Core(TM) i5-1135G7 2.40GHz 8,00 GB RAM Computer, with Windows 10 Pro version 22H2 64-bit OS. Needless to say, it is very important to keep all involved software up to date before trying to reproduce our experiment. It is also important to check if there are any update at the ’openai’ library, which can be checked at <https://platform.openai.com/docs/guides/fine-tuning>.

D.1 Script for Transforming XLSX to JSONL Format

1. Define Required Libraries :
    - Library for reading Excel files
    - Library for handling JSON data
2. Define a Function " xlsx_to_jsonl " with Parameters :
    - xlsx_path : Path to the Excel file
    - jsonl_path : Path to the output JSONL file
3. Begin Function :
    - Load the Excel workbook from the specified path
    - Select the active worksheet from the workbook
4. Open the JSONL file at the specified path in write mode
5. Process each row in the Excel worksheet , starting from the second row:
    - For each row , extract the row data
6. Construct a JSON object for the current row:
    - Create a list named " messages " with three parts:
    - A fixed system message
    - A user message containing the first column of the current row
    - An assistant message containing the second column of the current row
7. Convert the JSON object to a string and write it to the JSONL file, followed by a newline

8. Repeat steps 5-7 for each row in the worksheet
9. Close the JSONL file after finishing writing all rows
10. End Function
11. Execute the Function " xlsx_to_jsonl ":
    - Provide the actual path to the Excel file (‘AerospaceDS .xlsx’)
    - Provide path where the JSONL file should be saved (‘AerospaceDS.jsonl ’)

D.2 Script for Shuffling and Splitting the Dataset into Training, Validation, and Test Dataset

1. Define Required Library :
    - Library for generating random numbers
2. Define a Function " split_jsonl_file " with Parameters :
    - original_file : Path to the source JSONL file
    - train_file : Path to the output training file
    - validation_file : Path to the output validation file
    - test_file : Path to the output test file
    - train_ratio : Fraction of data to be used for training (default is 0.7)
    - validation_ratio : Fraction of data to be used for validation (default is 0.15)
3. Begin Function :
    - Open the original JSONL file and read all lines into a list
4. Shuffle the list of lines randomly to ensure unbiased data splits
5. Calculate the number of lines for each data split based on the provided ratios:
    - Total number of lines in the original file
    - Calculate the index where the training data ends
    - Calculate the index where the validation data ends
6. Split the shuffled lines into three separate lists:
    - Training data: from the start to the training end index
    - Validation data: from the training end index to the validation end index
    - Test data: from the validation end index to the end of the list
7. Write each data split to its respective new file:
    - Write the training data lines to the training file
    - Write the validation data lines to the validation file
    - Write the test data lines to the test file

8. Close all files
9. End Function
10. Execute the Function " split_jsonl_file ":
    - Provide paths for the original JSONL file and the destination files for training, validation , and testing

D.3 Script for Uploading the Training/Validation/Test Dataset

1. Define Required Library:
    - Library for interacting with OpenAI’s API
2. Set the API Key:
    - Store the API key "my_key" for connecting with the OpenAI API
3. Define a Function "fine_tune_model" with Parameter:
    - file_path: Path to the file intended for upload
4. Begin Function:
    - Open the file located at the specified file_path in read-binary mode
5. Upload the file to OpenAI:
    - Call the file upload function from the OpenAI library
    - Set the purpose of the upload to 'fine-tune'
6. Retrieve and store the file ID from the upload response:
    - Extract the 'id' attribute from the upload response
7. Display a message indicating successful upload and show the file ID:
    - Print the message "File uploaded successfully. ID: [file_id]"
8. End Function
9. Execute the Function "fine_tune_model":
    - Call the function with the path to the training dataset ("AerospaceDS-train.jsonl") as an argument

D.4 Script for creating the Fine-Tuning job at OpenAI Platform

1. Define Required Library:
    - Library for interacting with OpenAI’s API
2. Set the API Key:
    - Store the API key "my_key" for connecting with OpenAI’s API
3. Define a Function "start_finetuning_job" with Parameters:

- file_id: ID of the file uploaded to OpenAI for fine-tuning
- model: The model type to use for fine-tuning, default is "gpt-3.5-turbo"

4. Begin Function:
    - Try to:
        a. Create a fine-tuning job using the provided file ID and model:
            - Specify the training file by file_id
            - Specify the model type
            - Set hyperparameters, such as "n_epochs" to 20
        b. Display a message indicating the successful creation of the fine-tuning job along with job details
        c. Return the job object
    - If an error occurs:
        a. Display an error message with the error details
        b. Return None to indicate the failure of job creation
5. End Function
6. Execute the Function " start_finetuning_job ":
    - Call the function with a specific file ID to start the fine-tuning process

D.5 Script for Validating/Testing the Trained Model

1. Define Required Libraries :
    - Library for making HTTP requests
    - Library for handling JSON data
2. Define a Function " query_chat_model " with Parameters :
    - prompt: The input prompt to send to the model
    - model_name : Identifier for the specific trained model (default specified)
    - api_key : Authorization key for API access (actual key must be provided in real cases)
3. Begin Function " query_chat_model ":
    - Define headers for the HTTP request :
        a. Authorization header with the bearer token using the api_key
        b. Content -Type set to " application /json"
    - Define the request data:
        a. Set the model to use

b. Set the messages for system and user roles with the prompt

- Make a POST request to the model API endpoint with the specified headers and data
- Check the HTTP response status:

a. If 200 OK , extract the message content from the response , process , and return it

b. If not 200 OK , return an error message with the status code and text from the response

4. Define a Function " run_test_suite " with Parameters :

- model_name : Identifier for the specific trained model
- api_key : Authorization key for API access
- test_file : Path to the test dataset (default " test_dataset .jsonl")

5. Begin Function " run_test_suite ":

- Initialize counters for correct and total responses
- Open and read the test file line by line:

a. For each line , parse the JSON to get the user prompt and expected responses

b. Print the prompt being tested

c. Call model with the prompt to get the model's response

d. Print the model's response

e. Check if the model's response contains any of the expected responses:

- If yes, increment the count of correct responses

f. Increment the count of total responses

- Calculate the accuracy percentage:

a. Compute the ratio of correct responses to total responses and multiply by 100

b. Print the accuracy along with the counts of correct and total responses

6. Set specific values for model_name and test_file

7. Execute the Function "run_test_suite" to measure model accuracy using the specified model and test file

Appendix E – Former IFI employees FATs

E.1 FAT from Nelshio Haraguchi

	EXAMPLE TEMPLATE Technical Analysis Datasheet		
FOLHA DE ANÁLISE TÉCNICA – FAT			
CONTRATO: NOT APPLICABLE	**PROCESSO:** NOT APPLICABLE	**FAT Nº:** NOT APPLICABLE	
TÍTULO DO DOCUMENTO: PHA FUZE BEF-1502			
Nº DE REF.: NOT APPLICABLE			
REQUISITOS: MIL-STD-882E – Task 202			

VERIFICATION OF COMPLIANCE WITH APPLICABLE REQUIREMENTS AND COMMENTS:

1. OBJETIVE

To evaluate the document PHA FUZE BEF-1502 (Brazilian Electronic Fuze model 1502), as per MIL STD 882E Task 202 - Preliminary Hazard Analysis (PHA) to identify hazards, assess the initial risks, and identify potential mitigation measures. For this evaluation the following documents have been received:

- PHA FUZE BEF-1502 (Brazilian Electronic Fuze model 1502)
- MAC JEE Electronic Fuze Presentation
- BEF-1502 - TECHNICAL SPECIFICATIONS - Rev 000
- PHA - MIL-STD-882E – Task 202

2. RELATED DOCUMENTATION:

2.1 MIL-STD-882E System Safety.

3. ANALYSIS RESULTS:

As explained on TASK 202 – Preliminary Hazard Analysis (PHA) this analysis in one of the first step of an interactive process where the PHA is being periodically reviewed during the project development.

Document PHA FUZE BEF-1502 is one of the main components of PSSA - Preliminary System Safety Assessment. As such an additional PHA objective is the completeness of hazard events list to avoid a late discovery, a careful severity failure conditions effects classification as it will be used as safety requirement for the developing engineers and as such will be subjected to further substantiation before the program certification closure.

Fuse internal architecture may be affected as severity 'I' as a safety requirement 'no single event leading to this outcome' does apply.

In some cases an event severity level may be reduced from the original evaluation considering implement mitigation at design level. However this is not a common procedure for mitigation related to operation tasks and training requirements.

The following are the PHA FUSE BEF-1502 analysis:

F1. 'Power supply from FZU to FMU' – fuze does not receive power due to battery failure and fails to arm during a pre-release phase with a severity 'I'. Justifiy the severity level 'I' as the only immediate effect seems to be the inability to arm.

F2. 'Arming time selection' – selector malfunction causes incorrect arming leading to premature detonation at pre release phase with severity 'I'. Justify severity 'I'. Incorrect delay time should not lead to a catastrophic event during pre

VERIFICATION OF COMPLIANCE WITH APPLICABLE REQUIREMENTS AND COMMENTS: (continued)	PROCESSO:	NOT APPLICABLE	FAT Nº:	NOT APPLICABLE

release phase. Addition of operational requirement review should be considered.

F3. 'Delay to function' due to timer failure leading to wrong delay and unwanted detonation timing at post release phase with severity 'II'. Besides proposed mitigation it would be appropriate to stablish a set of qualification requirements that result in a reliable component performance.

F4. 'Safe distance parameter' – incorrect parameter value leading to detonation within unsafe distance during arming phase with severity 'I'. Please justify severity and besides proposed mitigation, consider introduction of additional safety requirement to avoid detonation before arming. Single failure path is not acceptable for severity 'I'.

F5. 'Connection between FZU and FMU', damaged cable leads to loss of comm or power at pre release phase with severity 'III'. Effect would be an inactive fuze. Besides proposed mitigation, connection cable should be properly qualified.

F6. 'DSU-33 Proximity Sensor Integration' fails signal to detonation before impact at airburst phase with severity 'III'. Proposed mitigation seems to address failure due to installation. Fuse testing requirement should address this concern.

F7. 'Voltage Booster (40V to 1200V)' circuit overload leading to inability to detonate at arming phase with severity 'II'. Environment qualification requirement and testing should be stablished to obtain desired reliability detonation performance.

F8. 'Super capacitor charge' inability to detonation due to capacitor charge failure at arming phase with severity 'III'. Besides proposed mitigation a component qualification requirement should be stablished similar to F7.

F9. "Environmental exposure' to extreme temperature leads fuze to fail at storage/operation phase with severity 'III'. Requirements for environmental qualifications during storage and operation should be stablished. MIL HDBK 310 should be used as guidance for qualification requirement.

F10. 'Software driven arming logic' – software error leading to incorrect sequence execution during arming phase with severity 'II'. Besides proposed mitigation, software development requirement should be implemented as per NATO-AOP-52 – guidance on software safety design and assessment or equivalent.

F11. 'FZU safety activation pin', inadvertent removal with fuze being armed during pre release phase with severity 'I'. Besides proposed mitigation, consider implementation of requirement "fuze shall not initiate detonation when safety pin is activated". Severity 'I' classification requires more than one path leading to this event.

F12. 'Proximity sensor input' signal interference, false or no signal during airburst phase with severity 'III'. 'False signal' and 'no signal' leads to a different scenario resulting in no detonation before impact, or an untimely detonation. Each scenario may need a specific requirement for the component qualification.

F13. 'Detonator initiation' inability to initiate detonation due to LEEFI failure during impact phase with severity "II". Besides proposed mitigation an appropriate reliability requirement should be stablished.

F14. 'Mechanical integrity' loose closure ring leading to loss of ring in flight during pre release phase with severity 'III'. Besides proposed mitigation implementation a requirement for a qualified personal to be in charge of fuze installation should be implemented.

F15. 'Operator error in setting delay parameters' resulting in an incorrect delay setting with mission failure due to premature detonation during pre release phase with severity 'II'. Justify severity as detonation occurs during pre release phase. Besides proposed mitigation a requirement regarding personal in charge of this activity should be previously qualified for this task.

F16. 'High Voltage Charge Bank' with component damage or fire during arming phase with severity 'I'. Justify severity 'I'. Besides proposed mitigation implementation considers as applicable, a fuze architecture review as a single failure is not allowed to lead to a catastrophic severity.

F17. 'MK70 cable connector' loose resulting in loss of power or signal during pre release phase with severity 'III'. Besides proposed mitigation being adopted consider reviewing F1 and F6 for harmonization regarding severity classification.

F18. "Human factors' error setup due to inadequate training during pre release phase with severity 'III'. This hazard is

VERIFICATION OF COMPLIANCE WITH APPLICABLE REQUIREMENTS AND COMMENTS: (continued)	PROCESSO:	NOT APPLICABLE	FAT Nº:	NOT APPLICABLE

related to F15. Consider reviewing severity classification.

F19. "Impact sensor 'malfunction fails to trigger detonation upon impact phase with severity 'III'. Besides proposed mitigation using redundant sensors, a reliability requirement compatible with fuse performance should be specified.

F20. 'Built in self tests' to detect faults before entering mission at pre release phase with severity 'II'. Besides proposed mitigation being implemented, a built in self-test should be able to indicate when a fault/failure are detected.

4. NOTE:

The following should be considered as an additional hazard events to be included in the PHA:

1. Free fall sensor malfunction fuze input to the BFE-1502 ESAD.
2. Fuze unable to initiate detonation due to electro magnetic interference.
3. Delay setting devices – should be analyzed to be free of error prone characteristics.

CONCLUSION - RESULTS OF DOCUMENT ANALYSIS:

SUBSTANTIATION / VERIFICATION OF APPLICABLE REQUIREMENTS:

X the document evidences the substantiation and/or compliance with applicable requirements

the document does NOT demonstrate substantiation and/or compliance with applicable requirements

ANALYST'S OPINION **(see comments, if applicable):**

document approved/accepted

X document approved/accepted with restrictions – DOES NOT REQUIRE REVIEW

document approved/accepted with restrictions – REQUIRES REVIEW

document NOT approved/accepted

RESPONSIBLE	DATA
Haraguchi	Jan 11, 2025

E.2 FAT from Raphael Cortes

	EXAMPLE TEMPLATE Technical Analysis Datasheet

FOLHA DE ANÁLISE TÉCNICA – FAT

CONTRATO:	NOT APPLICABLE	**PROCESSO:**	NOT APPLICABLE	**FAT Nº:**	NOT APPLICABLE
TÍTULO DO DOCUMENTO:	PHA				
Nº DE REF.:	NOT APPLICABLE				
REQUISITOS:	MIL-STD-882E – Task 202				

VERIFICATION OF COMPLIANCE WITH APPLICABLE REQUIREMENTS AND COMMENTS:

1. The PHA has identified in the function ID F9, the function environmental exposure in the mission phase storage and operation. This is the only environmental function identified in the PHA.
   The MIL-STD-882E–Task 202 states that PHA should consider the contribution mishaps from operating environment and constraints (202.2.2.k) and environmental impacts (202.2.2.o).
   In the other hand, the technical specification mentions MIL-STD-331D as a reference for the system. MIL-STD-331D includes temperature tests, vibration tests and shock tests for example.
   Indeed, the PHA problems related to other environmental aspects should be included.
2. The PHA has identified the function ID F11 related to FZU safety activation pin at pre-release phase. The PHA does not include a possible malfunction due to activation pin of FZU after realease, which may be linked to the operation environment (202.2.2.k) or production malfunctions (202.2.2.t). In this case the battery supply will not be provided to the system.
3. As the Fuze will be operated in an environment with high electrical electromagnetic fields (intended or not intended), this kind of energy could interfere with the system or software's behavior. The PHA does not include an analysis regarding possible external interference due to external energy sources (202.2.2.b).
4. The PHA has identified an incorrect safe distance parameter value in the function ID F4. However, no analysis was conducted of the possible failure of the 3-axis free fall sensor that could lead to arming within the safety distance (202.2.2.t).
5. The severity category 3 (Marginal) should be better explained in the case of the arming phase for the functions F4 and F16 (202.2.2.q). In these functions, there is a possibility to damage the aircraft though the severity may be greater.

CONCLUSION - RESULTS OF DOCUMENT ANALYSIS:

SUBSTANTIATION / VERIFICATION OF APPLICABLE REQUIREMENTS:

X **the document evidences the substantiation and/or compliance with applicable requirements**

the document does NOT demonstrate substantiation and/or compliance with applicable requirements

***ANALYST'S OPINION* (see comments, if applicable):**

document approved/accepted

document approved/accepted with restrictions – DOES NOT REQUIRE REVIEW

X **document approved/accepted with restrictions – REQUIRES REVIEW**

document NOT approved/accepted

RESPONSIBLE	DATA
Raphael Gomes Cortes - Maj	**13/12/2024**

E.3 FAT from William Limonge

	EXAMPLE TEMPLATE Technical Analysis Datasheet

FOLHA DE ANÁLISE TÉCNICA – FAT

CONTRATO:	NOT APPLICABLE	**PROCESSO:**	NOT APPLICABLE	**FAT N°:**	NOT APPLICABLE
TITULO DO DOCUMENTO:	PHA				
N° DE REF.:	NOT APPLICABLE				
REQUISITOS:	MIL-STD-882E – Task 202.2.2				

VERIFICATION OF COMPLIANCE WITH APPLICABLE REQUIREMENTS AND COMMENTS:

The PHA was evaluated against the item 202.2.2 of MIL-STD-882E. The issues pointed on 3rd column must be addressed to assure the PHA is in compliance with Task 202.2.2:

202.2.2 The PHA shall identify hazards by considering the potential contribution to subsystem or system mishaps from:		
a. System components.	F1 / F2 / F3 / F5 / F7 / F8 / F13 / F16 / F17 / F19	
b. Energy sources.	F1 / F16 / F17	
c. Ordnance.	F10 / F20	
d. Hazardous Materials (HAZMAT).		There shold be an engineering statement about the "absence of Hazardous Materials".
e. Interfaces and controls.	F15 / F18	
f. Interface considerations to other systems when in a network or System-of-Systems (SoS) architecture.	F1 / F2 / F3 / F4 / F6 / F7 / F8 / F9 / F10 / F11 / F12 / F13 / F14 / F15 / F16 / F17 /	
g. Material compatibilities.		There shold be an engineering statement about the "absence of problems related to material compatibilities".
h. Inadvertent activation.	F2 / F3 / F4 / F10 / F11 / F15	
i. Commercial-Off-the-Shelf (COTS), Government-Off-the-Shelf (GOTS), Non-Developmental Items (NDIs), and Government-Furnished Equipment (GFE).		There should be statements about the use or not of this kind of components or materials.
j. Software, including software developed by other contractors or sources. Design criteria to control safety-significant software commands and responses (e.g., inadvertent command, failure to command, untimely command or responses, and inappropriate magnitude) shall be identified, and appropriate action shall be taken to incorporate these into the software (and related hardware) specifications.	F/10	
k. Operating environment and constraints.	F/9	
l. Procedures for operating, test, maintenance, built-in-test, diagnostics, emergencies, explosive ordnance render-safe and emergency disposal.	F10 / F20	
m. Modes.		
n. Health hazards.		There shold be an engineering statement about the "absence of problems related to health hazards".
o. Environmental impacts.		There shold be an engineering statement about the "absence of problems related to environmental impacts".
p. Human factors engineering and human error analysis of operator functions, tasks, and requirements.	F15 / F18	
q. Life support requirements and safety implications in manned systems, including crash safety, egress, rescue, survival, and salvage.		There shold be a reference to the SHA.
r. Event-unique hazards.		There shold be engineering statement about the compliance to this requirement.
s. Built infrastructure, real property installed equipment, and support equipment.		There shold be engineering statement about the compliance to this requirement.
t. Malfunctions of the SoS, system, subsystems, components, or software.	F2 / F10 / F12 / F19	

VERIFICATION OF COMPLIANCE WITH APPLICABLE REQUIREMENTS AND COMMENTS: (continued)	PROCESSO:	NOT APPLICABLE	FAT Nº:	NOT APPLICABLE

CONCLUSION - RESULTS OF DOCUMENT ANALYSIS:

SUBSTANTIATION / VERIFICATION OF APPLICABLE REQUIREMENTS:

X **the document evidences the substantiation and/or compliance with applicable requirements**

the document does NOT demonstrate substantiation and/or compliance with applicable requirements

ANALYST'S OPINION **(see comments, if applicable):**

document approved/accepted

document approved/accepted with restrictions – DOES NOT REQUIRE REVIEW

X **document approved/accepted with restrictions – REQUIRES REVIEW**

document NOT approved/accepted

RESPONSIBLE	DATA
Engenheiro Industrial Mecânico **Willian Limonge**	**19/01/2025**

E.4 FAT from Vitor Bourguignon

	EXAMPLE TEMPLATE Technical Analysis Datasheet

FOLHA DE ANÁLISE TÉCNICA – FAT

CONTRATO:	NOT APPLICABLE	**PROCESSO:**	NOT APPLICABLE	**FAT Nº:**	NOT APPLICABLE
TITULO DO DOCUMENTO:	PHA				
Nº DE REF.:	NOT APPLICABLE				
REQUISITOS:	MIL-STD-882E – Task 202				

VERIFICATION OF COMPLIANCE WITH APPLICABLE REQUIREMENTS AND COMMENTS:

After analyzing the PHA, the following comments were made:

1) The definition (legend) for the severity categories, for the Risk Assessment Code (RAC) categories, and for the Mission Phase was missing.

2) The analysis should contain a value for probability, even if qualitative.

   See MIL-STD-882E: "202.2 Task description. The contractor shall perform and document a PHA to determine initial risk assessments of identified hazards. Hazards associated with the proposed design or function shall be evaluated for severity and probability based on the best available data."

3) The analysis should include a hazard related to electromagnetic interference/compatibility.
   See MIL-STD-882E: "202.2.2 The PHA shall identify hazards by considering the potential contribution to subsystem or system mishaps from:
   o. Environmental impacts."

4) A hazard related to human error was missing during the Pre-Installation Inspection, for example, "If the component has been dropped from a height exceeding three feet when packaged or from more than one foot when unpacked, it must be rejected if visual damage is observed." A hazard would be if the component has been dropped from a height exceeding 3 feet, and no visual inspection is performed, or if there is visual damage and the operator does not identify it during inspection. This item is related to item "l. Procedures for operating, testing, maintenance, built-in-test, diagnostics, emergencies, explosive ordnance render-safe and emergency disposal" of MIL-STD-882E.

5) There was no mention of hazards related to storage. Since there are no comments on the environmental qualification of Fuze, storage precautions should be observed, such as: rain, snow, humidity, etc.
   This item is related to item "o. Environmental impacts" of the MIL-STD-882E.

CONCLUSION - RESULTS OF DOCUMENT ANALYSIS:

SUBSTANTIATION / VERIFICATION OF APPLICABLE REQUIREMENTS:

(X) **the document evidences the substantiation and/or compliance with applicable requirements**

the document does NOT demonstrate substantiation and/or compliance with applicable requirements

***ANALYST'S OPINION* (see comments, if applicable):**

document approved/accepted

document approved/accepted with restrictions – DOES NOT REQUIRE REVIEW

(X) **document approved/accepted with restrictions – REQUIRES REVIEW**

document NOT approved/accepted

RESPONSIBLE	DATA
Vitor Bourguignon	**20/12/2024**

Appendix F – MoCs JSONL Dataset sample

{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": "Tracking capabilities"}, {"role": "assistant", "content": "Laboratory Test, Demonstration"}]}
{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": "Aluminium alloy components"}, {"role": "assistant", "content": "Analysis"}]}
{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": "Missile Active mode: Cooling gas ON shall be received"}, {"role": "assistant", "content": "Laboratory Test"}]}
{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": "On dynamic I-BIT reboot when the reboot masking level have been set on I-BIT, the MFS shall: request a Reboot from the Missile Fail state"}, {"role": "assistant", "content": "Laboratory Test"}]}
{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": "Store F-BIT failures on the MFS, to download to support equipment\nduring maintenance"}, {"role": "assistant", "content": "Laboratory Test"}]}
{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": "Jettison state"}, {"role": "assistant", "content": "Laboratory Test"}]}
{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": "Composition"}, {"role": "assistant", "content": "Inspection"}]}
{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": "MCP BIT failure handling"}, {"role": "assistant", "content": "Laboratory Test"}]}
{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": "Contamination"}, {"role": "assistant", "content": "Laboratory Test"}]}
{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": "Maintenance state requirements"}, {"role": "assistant", "content": "Qualification test"}]}
{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": "Navigation state capabilities"}, {"role": "assistant", "content": "Laboratory Test, Flight Test, Simulation"}]}
{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": "The UA must have handling qualities that ensures the demands made on the DUO are not excessive taking into account the phase of flight and its duration."}, {"role": "assistant", "content": "Flight Test"}]}
{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": "Actuator moments of inertia"}, {"role": "assistant", "content": "Inspection"}]}
{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": "GNSS interface

as a source of navigation data for other equipment:"}, {"role": "assistant", "content": "Compliance Statement"}]}
{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": "Markings in Airspeed Indicator"}, {"role": "assistant", "content": "Inspection"}]}
{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": "If launch command is received: The MFS shall transition to Missile Slaved vector [0,0] state and continue to Separation state if the\nA/C AIM message was received"}, {"role": "assistant", "content": "Laboratory Test"}]}
{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": "Mode Maintenance"}, {"role": "assistant", "content": "Inspection, Demonstration"}]}
{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": "The position of the instruments used to guide and control the aircraft must allow for clear viewing with minimal deviation from the normal flight line of sight (pilot looking forward \u00ef\u00bf\u00bd center of the basic T);"}, {"role": "assistant", "content": "Design Review, Ground Test"}]}
{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": "Cabin Assessment - Airplane instruments"}, {"role": "assistant", "content": "Ground test"}]}
{"messages": [{"role": "system", "content": "You are a System Engineer and need to assign proper verification methods for a requirement."}, {"role": "user", "content": " It applies to materials and parts that support or reinforce the structure, in relation to their resistance, aiming to choose design values (item origin reference) that minimize the probability of structural failure."}, {"role": "assistant", "content": "Analysis"}]}

FOLHA DE REGISTRO DO DOCUMENTO

1. CLASSIFICAÇÃO/TIPO	2. DATA	3. REGISTRO N°	4. N° DE PÁGINAS
TD	27 de fevereiro de 2025	DCTA/ITA/TD-004/2025	207

5. TÍTULO E SUBTÍTULO:

Use of large language models to support aerospace defense systems engineering.

6. AUTOR(ES):

Guilherme Micheli Bedini Moreira

7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES):

Instituto Tecnológico de Aeronáutica – ITA

8. PALAVRAS-CHAVE SUGERIDAS PELO AUTOR:

Large Language Model; System Theoretic Process Analysis; Aerospace Systems Engineering.

9.PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO:

Sistemas aeroespaciais; Elicitação de requisitos; Inteligência artificial; Engenharia de sistemas; Processos; Análise de sistemas; Força Aérea Brasileira; Engenharia aeroespacial.

10. APRESENTAÇÃO: **X Nacional** **Internacional**

ITA, São José dos Campos. Curso de Doutorado. Programa de Pós-Graduação em Engenharia Aeronáutica e Mecânica. Área de Projeto Aeronáutico, Estruturas e Sistemas Aeroespaciais. Orientador: Prof. Dr. Willer Gomes dos Santos; coorientador: Prof. Dr. Christopher Shneider Cerqueira. Defesa em 14/02/2025. Publicada em 2025.

11. RESUMO:

This research investigates the integration of Large Language Models (LLMs) into aerospace defense systems engineering to automate two critical processes: eliciting requirements through System Theoretic Process Analysis (STPA) and assigning Means of Compliance (MoCs) to aerospace defense systems' requirements. The motivation lies in addressing the labor-intensive and error-prone nature of traditional methods, which heavily rely on human expertise. The study specifically evaluates the feasibility and performance of LLMs, such as GPT-3.5 and GPT-4, when guided by advanced Prompt Engineering techniques and fine-tuning methodologies. These approaches aim to maintain or surpass the accuracy and quality typically achieved by experts in the field. The problem under investigation is the inefficiency and variability of manual requirements engineering and compliance processes, which are critical in defense aerospace systems due to stringent safety and reliability demands. Using a hypothetical Unmanned Combat Air Vehicle (UCAV) as a case study, the study situates the research in the context of the Brazilian Air Force (FAB), where these challenges are particularly acute. The methodology involves automating Phase 1 of STPA through tailored prompts to generate system requirements and training a fine-tuned model to assign MoCs accurately. Performance was benchmarked against real-world system data and domain experts' outputs. The findings highlight that LLMs guided by Prompt Engineering can generate requirements that meet or exceed eight of nine evaluated quality attributes, including testability, completeness, clarity, and modifiability. The fine-tuned 'gpt-3.5-turbo' model achieved an 80.18% accuracy in MoC assignments. Finally, with appropriate techniques, it was possible to generate safety assessment reports such as PHAs (Preliminary Hazard Analysis) from the technical documentation of real products. The implications of this research are profound. By streamlining requirements elicitation, MoC assignment, and the generation of engineering reports, LLMs reduce the time, effort, and cost associated with engineering processes while maintaining high standards of rigor and reliability. This work advances academic understanding of LLM applications in safety-critical systems, introduces a scalable and replicable framework for integrating LLMs into engineering workflows, and offers practical tools to the aerospace defense industry.

12. GRAU DE SIGILO:

(X) OSTENSIVO **( ) RESERVADO** **( ) SECRETO**