CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of transportation sciences
Department of air transport

Hodnocení informační bezpečnosti letadlového softwaru v letecké údržbě

Evaluation of Aircraft Software Security during Aircraft Maintenance

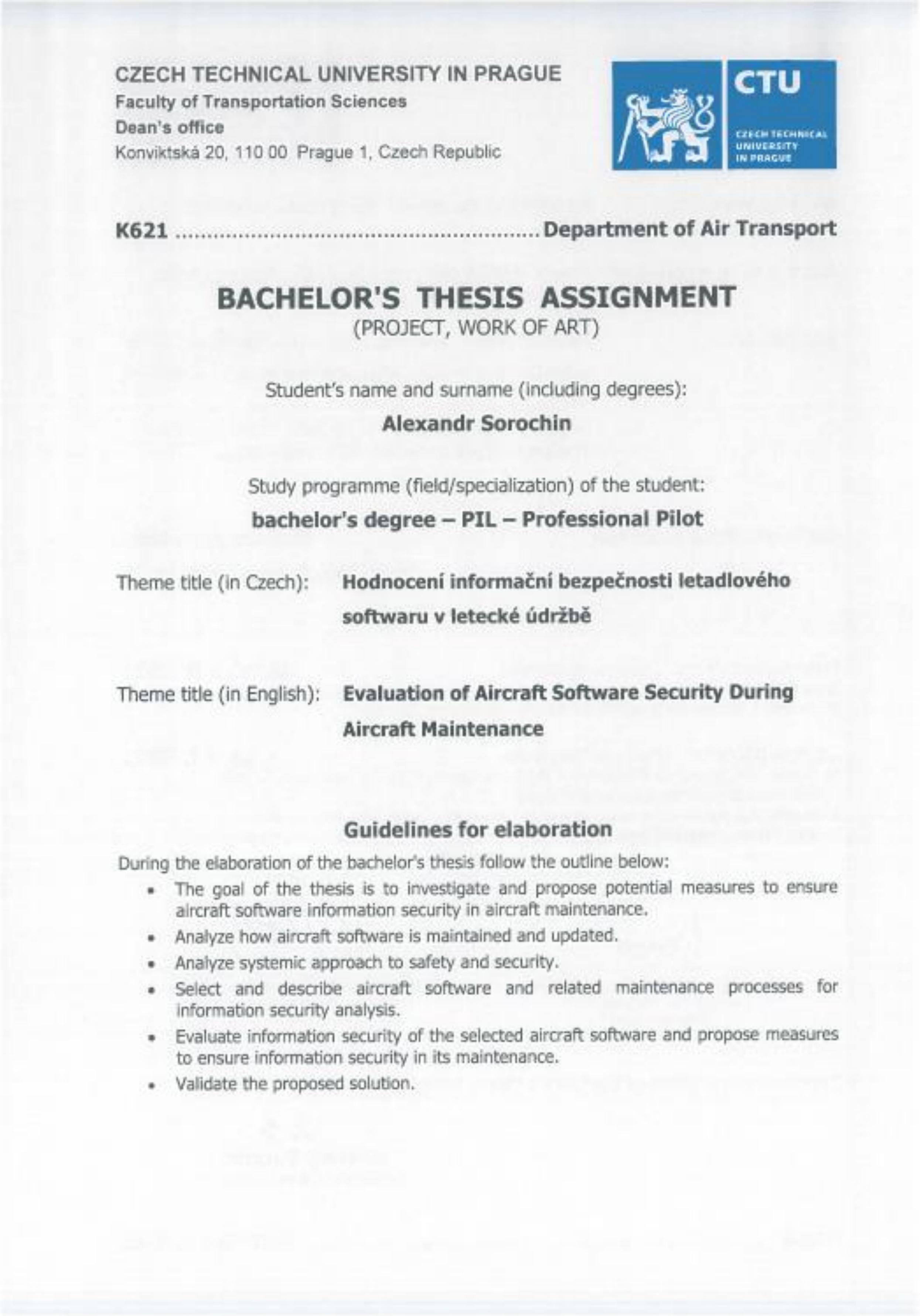
Bachelor's Thesis

Study Programme: Technology in Transportation and Telecommunications
Study Field: Professional Pilot

Thesis Supervisors: doc. Ing. Andrej Lališ, Ph.D.,
Max Chopart, MSc.

Alexandr Sorochin

Praha 2023



CZECH TECHNICAL UNIVERSITY IN PRAGUE
Faculty of Transportation Sciences
Dean's office
Konviktská 20, 110 00 Prague 1, Czech Republic

CTU
CZECH TECHNICAL UNIVERSITY IN PRAGUE

K621 .... Department of Air Transport

BACHELOR'S THESIS ASSIGNMENT

(PROJECT, WORK OF ART)

Student's name and surname (including degrees):

Alexandr Sorochin

Study programme (field/specialization) of the student:

bachelor's degree – PIL – Professional Pilot

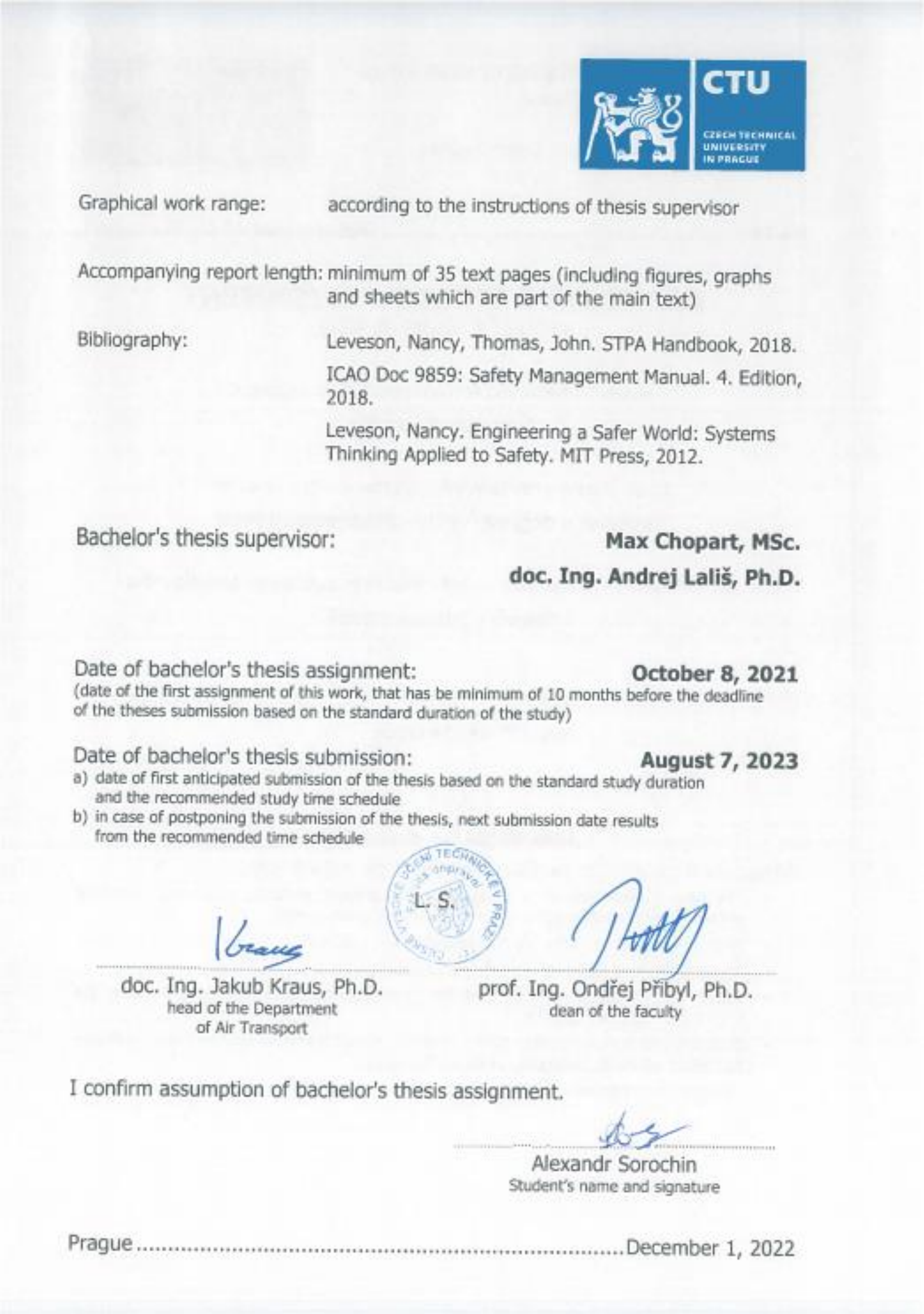
Theme title (in Czech): **Hodnocení informační bezpečnosti letadlového softwaru v letecké údržbě**

Theme title (in English): **Evaluation of Aircraft Software Security During Aircraft Maintenance**

Guidelines for elaboration

During the elaboration of the bachelor's thesis follow the outline below:

- The goal of the thesis is to investigate and propose potential measures to ensure aircraft software information security in aircraft maintenance.
- Analyze how aircraft software is maintained and updated.
- Analyze systemic approach to safety and security.
- Select and describe aircraft software and related maintenance processes for information security analysis.
- Evaluate information security of the selected aircraft software and propose measures to ensure information security in its maintenance.
- Validate the proposed solution.



CTU
CZECH TECHNICAL UNIVERSITY IN PRAGUE

Graphical work range: according to the instructions of thesis supervisor

Accompanying report length: minimum of 35 text pages (including figures, graphs and sheets which are part of the main text)

Bibliography: Leveson, Nancy, Thomas, John. STPA Handbook, 2018.

ICAO Doc 9859: Safety Management Manual. 4. Edition, 2018.

Leveson, Nancy. Engineering a Safer World: Systems Thinking Applied to Safety. MIT Press, 2012.

Bachelor's thesis supervisor: **Max Chopart, MSc.**
doc. Ing. Andrej Lališ, Ph.D.

Date of bachelor's thesis assignment: **October 8, 2021**
(date of the first assignment of this work, that has be minimum of 10 months before the deadline of the theses submission based on the standard duration of the study)

Date of bachelor's thesis submission: **August 7, 2023**
a) date of first anticipated submission of the thesis based on the standard study duration and the recommended study time schedule
b) in case of postponing the submission of the thesis, next submission date results from the recommended time schedule

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta dopravní
L. S.

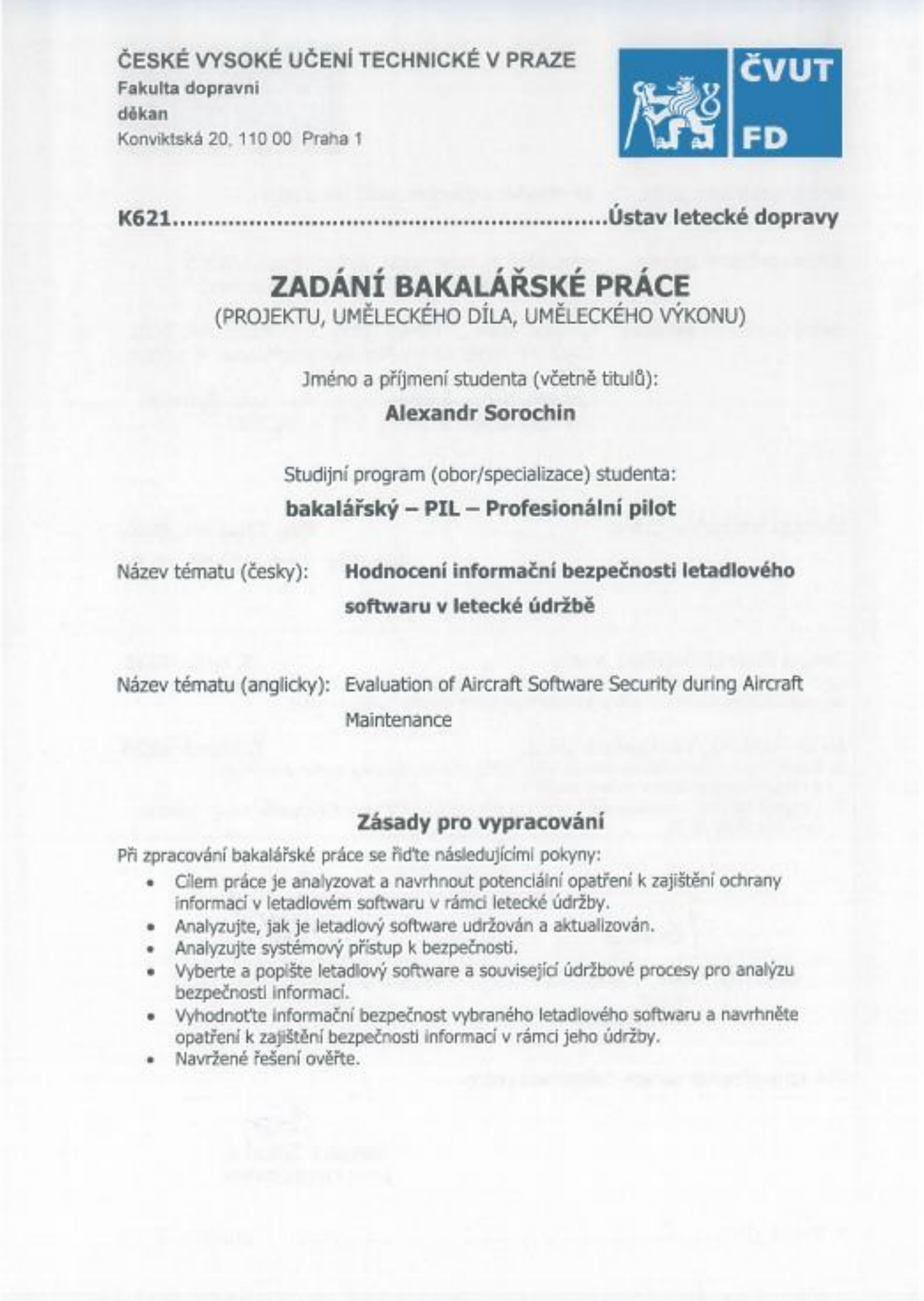
Kraus
doc. Ing. Jakub Kraus, Ph.D.
head of the Department of Air Transport

prof. Ing. Ondřej Přibyl, Ph.D.
dean of the faculty

I confirm assumption of bachelor's thesis assignment.

Alexandr Sorochin
Student's name and signature

Prague .... December 1, 2022



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta dopravní
děkan
Konviktská 20, 110 00 Praha 1

ČVUT
FD

K621...Ústav letecké dopravy

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Alexandr Sorochin

Studijní program (obor/specializace) studenta:

bakalářský – PIL – Profesionální pilot

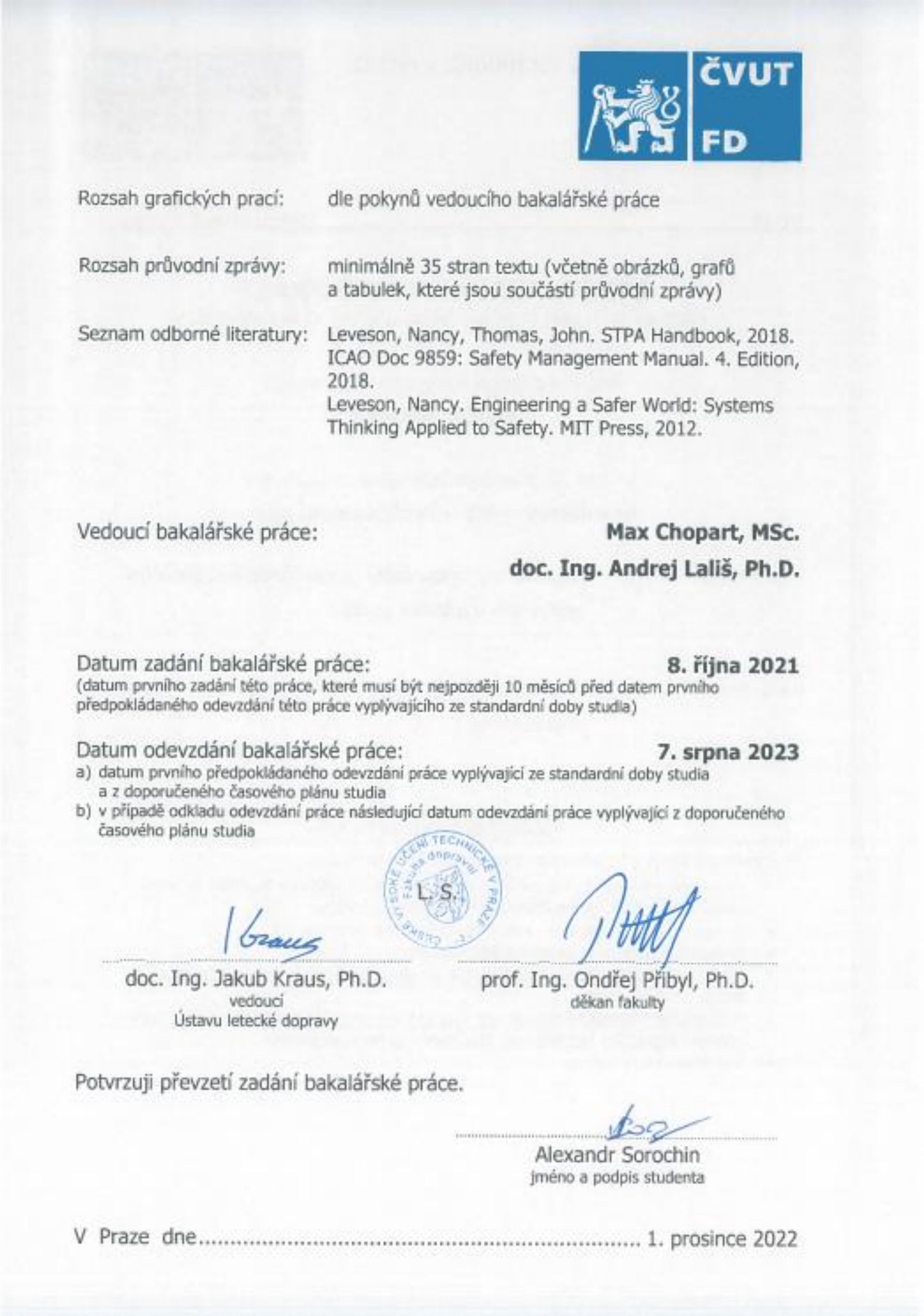
Název tématu (česky): **Hodnocení informační bezpečnosti letadlového softwaru v letecké údržbě**

Název tématu (anglicky): Evaluation of Aircraft Software Security during Aircraft Maintenance

Zásady pro vypracování

Při zpracování bakalářské práce se řiďte následujícími pokyny:

- Cílem práce je analyzovat a navrhnout potenciální opatření k zajištění ochrany informací v letadlovém softwaru v rámci letecké údržby.
- Analyzujte, jak je letadlový software udržován a aktualizován.
- Analyzujte systémový přístup k bezpečnosti.
- Vyberte a popište letadlový software a související údržbové procesy pro analýzu bezpečnosti informací.
- Vyhodnoťte informační bezpečnost vybraného letadlového softwaru a navrhněte opatření k zajištění bezpečnosti informací v rámci jeho údržby.
- Navržené řešení ověřte.



ČVUT FD

Rozsah grafických prací: dle pokynů vedoucího bakalářské práce

Rozsah průvodní zprávy: minimálně 35 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)

Seznam odborné literatury: Leveson, Nancy, Thomas, John. STPA Handbook, 2018.
ICAO Doc 9859: Safety Management Manual. 4. Edition, 2018.
Leveson, Nancy. Engineering a Safer World: Systems Thinking Applied to Safety. MIT Press, 2012.

Vedoucí bakalářské práce: **Max Chopart, MSc.**
doc. Ing. Andrej Lališ, Ph.D.

Datum zadání bakalářské práce: **8. října 2021**
(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání bakalářské práce: **7. srpna 2023**
a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta dopravní
L. S.

doc. Ing. Jakub Kraus, Ph.D.
vedoucí
Ústavu letecké dopravy

prof. Ing. Ondřej Přibyl, Ph.D.
děkan fakulty

Potvrzuji převzetí zadání bakalářské práce.

Alexandr Sorochin
jméno a podpis studenta

V Praze dne .... 1. prosince 2022

Abstrakt

V dnešní době se letadla pro svůj provoz silně spoléhají na odlišný software. Takový software je však extrémně obtížné testovat a zajistit, aby byl bezpečný, protože je stále složitější, zvláště pokud se do rovnice přidají vnější hrozby. Když je potřeba aktualizace nebo údržba, je k letadlu připojeno externí zařízení. V tuto chvíli by potenciální hrozby mohly integrovat software a ohrozit jeho integritu a bezpečnost. Práce je proto zaměřena na analýzu údržby konkrétního softwaru, jeho vyhodnocení a návrh opatření k zajištění informační bezpečnosti při jeho údržbě na základě STPA analýzy.

Klíčová slova: software, analýza bezpečnost, údržba, rizika, STPA

Abstract

Nowadays, aircraft heavily rely on the different software for its operation. However, such software is extremely difficult to test and to ensure that they are safe as they are more and more complex, especially if external threats are added into the equation. When an update or maintenance is needed, an external device is connected to the aircraft. At this moment, potential threats could integrate the software and compromise its integrity and safety. Therefore, the work is focused on the analysis of the maintenance of a specific software, its evaluation and proposal of measures to ensure information security in its maintenance, based on STPA analysis.

Keywords: software, security analysis, maintenance, hazards, STPA

Acknowledgment

My most profound appreciation goes to doc. Ing. Andrej Lališ Ph.D. and Max Chopart, MSc., my Bachelor thesis advisors and mentors, for their time, effort, and understanding in helping me succeed in my studies. Their vast wisdom and wealth of experience have inspired me throughout my studies. In addition, I'd like to thank Mr. Jan Zizka for his technical assistance throughout my research.

I would like to thank the Faculty of Transportation at Czech Technical University for providing me with the resources to pursue graduate study in the Air Trasport Department.

To conclude, I'd like to thank God, my parents, and my girlfriend. It would have been impossible to finish my studies without their unwavering support over the past few years.

Declaration

I hereby submit for assessment and defense a bachelor's thesis, prepared at the end of my studies at the Czech technical university in Prague, the faculty of transportation sciences.

I do not have a compelling reason against the use of this schoolwork within the intention of s.

60 of the Act No. 121/2000 Coll., on Copyright and Rights Related to Copyright and on Amendment to Certain Acts (Copyright Act).

I declare that I have elaborated this thesis independently using information sources listed in the bibliography in accordance with ethical guidelines for writing diploma thesis, which are listed on the document Methodological Instruction No. 1/2009.

In Prague on 07.08.2023 ....

Alexandr Sorochin
Student's name and signature

Table of content

List of figures

List of tables

List of abbreviations

AFDX – Avionics Full-Duplex Switched Ethernet, also ARINC 664;

DDoS – Distributed Denial of Service;

VHF – Very High Frequency;

HSDB – High Speed Data Bus;

BYOD – Bring your own device;

FAA – U.S. Federal Aviation Administration;

IRM – Integrated Risk Management;

ICAO – International Civil Aviation Organization;

LRU – Line-replaceable unit;

SCADA – Supervisory Control and Data Acquisition

STPA – System-Theoretic Process Analysis;

SD – SanDisk card

STAMP – System Theoretic Accident Model and Process

Glossary

AFDX Avionics Full-Duplex Switched Ethernet: is a data network, patented by international aircraft manufacturer Airbus, for safety-critical applications that utilizes dedicated bandwidth while providing deterministic quality of service;

ARINC 429: is a data transfer standard for aircraft avionics;

Accident: an accident is an unplanned and undesired loss event (involve human death and injury, and other major losses, including mission, equipment, financial, and information losses); Consequence: outcome of an event affecting objectives or the damage(s) done;

Hazard: a system state or set of conditions that, together with a particular set of worst-case environmental conditions, will lead to an accident;

Risk: a situation in which safety or security is lost and which requires immediate reaction to avoid or mitigate potential consequences;

Safety: freedom from accidents (loss events);

SCADA: is a control system architecture comprising computers, networked data communications and graphical user interfaces for high-level supervision of machines and processes. It also covers sensors and other devices, such as programmable logic controllers, which interface with process plant or machinery.

Introduction

The aircraft software is super reliable, but still not perfect. This was known long before the latest problems with the Boeing 737 MAX software. Unfortunately, when a more than 14 million lines of code are responsible for controlling an aircraft, no one could absolutely guarantee that everything will always be executed 100% as intended [1].

At the same time, the programs that control aircraft today, through years of improvement and "paranoid programming" mode that seeks to eliminate any possible errors, have reached a level where they are statistically more stable and reliable than the pilots that control them. The Federal Aviation Administration data shows that the majority of accidents and disasters within the past 10 years happened due to human factors, and not due to software problems [1].

A modern aircraft Flight control system, like programming languages, is not a novelty. One of the main programing languages in which code for civil aircraft is written is known to any programmer C/C++. However, the security of the software development process, due to its extreme reliability, is considered in this work as an assumption [1].

Modern onboard information infrastructure is divided by a firewall into several logical domains with different degrees of security used by the pilots in the cockpit, and a common network for the cabin crew, passenger’s entertainment systems connected to the Internet. Theoretically, hackers who gain access to the public aircraft domain are able to connect to trusted domain systems as well. However, the most likely scenarios of unauthorized access into onboard information infrastructure might associate with the wireless connections utilization, vulnerabilities in the pre-flight preparation process, in the process of updating the onboard software, downloading the flight database, which occurs every 28 days, as well as during routine preventive maintenance [1].

In this paper, we will consider situations associated with the software used by aircraft maintenance services employees, and the other ground personnel. The one of the possible reason of Spanair Flight 5022 crash might be an airline's information infrastructure infection by a computer virus called a Trojan. As a result, the technicians during the aircraft routine maintenance could not detect in time indications of the presence of at least three technical malfunctions that caused the crash [2].

In this paper, an information security framework is proposed for its software maintenance procedure, as well as a software security requirements extraction method based on STPA is introduced and a practical demonstration how to put it into use. STPA analysis method involves the environment, personnel, organizational and other factors, but this paper focuses on system failure and software fault of the G950 NXi Integrated Flight Deck software as a basis of the Tecnam P2006T aircraft information infrastructure.

1. The general aircraft onboard information security approach

1.1. Threats and vulnerabilities for an aircraft information infrastructure

Due to increase in the level of automation, the modern aircraft turns into a real flying computer, or in more professional terms SCADA system, which heavily relays on different software for its operations. This situation is not still considered as a relevant during the pilot training program, because of the low probability of cyberattacks in comparison with the hijacking and the threat of biological and radiation contamination. However, in the context of the development of the availability of malicious tools, openness and dynamics of the knowledge dissemination about the methods of attacks in the Internet, it is possible to predict an increase in number of cyberattacks in the aviation segment [3].

Hackers are able not only to extract information processing at the aircraft information infrastructure, but also to distort the reliability of information about the air situation, air traffic parameters, commercial data, etc., which might negatively affect various air traffic management and organization processes [3].

The main sources of information security vulnerabilities of onboard information infrastructure could be (see Table 1):

- undeclared capabilities of the onboard software or ground services;

- vulnerabilities of airborne and ground communications, navigation, surveillance and guidance services;

- vulnerabilities of onboard network information infrastructure;

- Vulnerabilities of onboard wireless and sensors networks of the aircraft (3).

Date	Incident short description
April, 2015	American Airlines was forced to delay multiple flights after the iPad app used by pilots to plot a route, as well as to get information about the estimated time of the aircraft movement, crashed. The issue affected a few dozen flights across the airline [1].
May, 2015	Boeing‘s 787 airplane, nicknamed the Dreamliner, that has been powered continuously for 248 days can lose all alternating current electrical power due to the generator control

	units simultaneously going into failsafe mode, the FAA said in a statement warning of the flaw [1].
May, 2015	Chris Roberts, a computer security expert, told the FBI agency that he hacked a plane's in-flight entertainment system while on board and managed to move the plane sideways [13].
June, 2015	Polish airline LOT was unable to create flight plans for outbound flights from its Warsaw hub and as a result, outbound flights from Warsaw were not able to depart because of the Distributed Denial of Service (DDoS) attack. A LOT representative said other airlines use comparable software systems [3].
April, 2013	Hugo Teso demonstrate at the security conference in Amsterdam how to remotely attack and take full control of an aircraft using an Android application. Here are few important facts: Automated Dependent Surveillance-Broadcast (ADS-B) has no security (It is unencrypted and unauthenticated), because of that hacker was able to inject ghost planes into radar. The Aircraft Communications Addressing and Reporting System (ACARS) also has no security; it is used for exchanging text messages between aircraft and ground stations via radio (VHF) or satellite [13].
August, 2018	The virus infiltration into the computers of the ground technical service did not allow timely detection of indications of the presence of at least three technical malfunctions in the Spanair aircraft, which crashed lately [2].

Table 1. The known information security incidents

1.2. The safety and security concept in the aviation

The aviation system as a whole comprises many and different functional systems such as finance, environment, safety and security. The latter two are the primary operational domains of the greater aviation system. As concepts, they share important features, as they are all concerned with the risk of events with consequence of various magnitudes. Nevertheless, they differ in the important element of intent. Security is concerned with malicious, intentional acts to disrupt the performance of a system. Safety focuses on the negative impact to the concerned systems' performance caused by unintended consequences of a combination of factors [4].

In the operational context, all of the functional systems produce some sort of risk that needs to be appropriately managed to lessen any adverse consequence. Traditionally, each system has developed sector specific risk management frameworks and practices designed to address the distinct characteristics of each system. Most of those risk management practices include comprehensive analysis on intra-system consequences, often referred to as the management of unintended consequences [4]. Another aspect is inter-system consequences resulting from system specific risk management processes. This relates to the fact that an effective risk management strategy of one specific sector can have an adverse impact on another operational sector of aviation. In aviation, the most often emphasized inter-system dependence is the safety/security dilemma. Effective security measures may have negative impacts on safety, and vice versa. Safety and security domains may differ in the element of underlying intent, but they converge in their common goal to protect people and assets (e.g. addressing cyber threats and risks requires coordination across the aviation safety and security domains). In some cases, the management of the inherent risk of one may affect the other domain in unforeseen ways, such as in the following examples:

a) reinforced cockpit doors necessitated due to security risks may have safety implications on the operation of an aircraft;

b) restrictions on the carriage of personal electronic devices in the cabin may displace the security risk from the cabin to the cargo hold, leading to heightened safety risk; and

c) change of routes to avoid flying over conflict zones may result in congested air corridors that pose a safety issue [4].

1.3. An aircraft onboard information infrastructure architecture

Traditionally, an aircraft represented a relatively closed information infrastructure. All aircraft instruments and devices were autonomous, without the ability to connect to them and transfer information during the flight, due to which they had a high level of information security in terms of unauthorized access from the external environment [3].

Because of the transition to predominantly digital methods of data processing and provision, an increase significantly of the degree of intellectualization of the aircraft’s onboard equipment complex, as well as the complexity of the aircraft software, for instance, the modern aircraft navigation software function has about 850 thousand lines of code now [3].

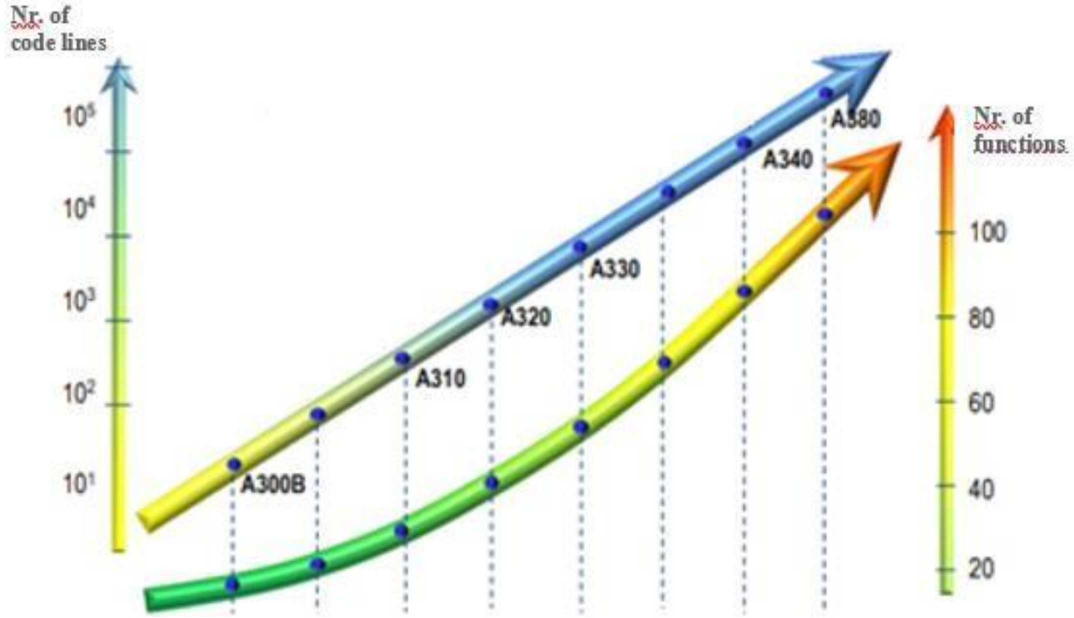
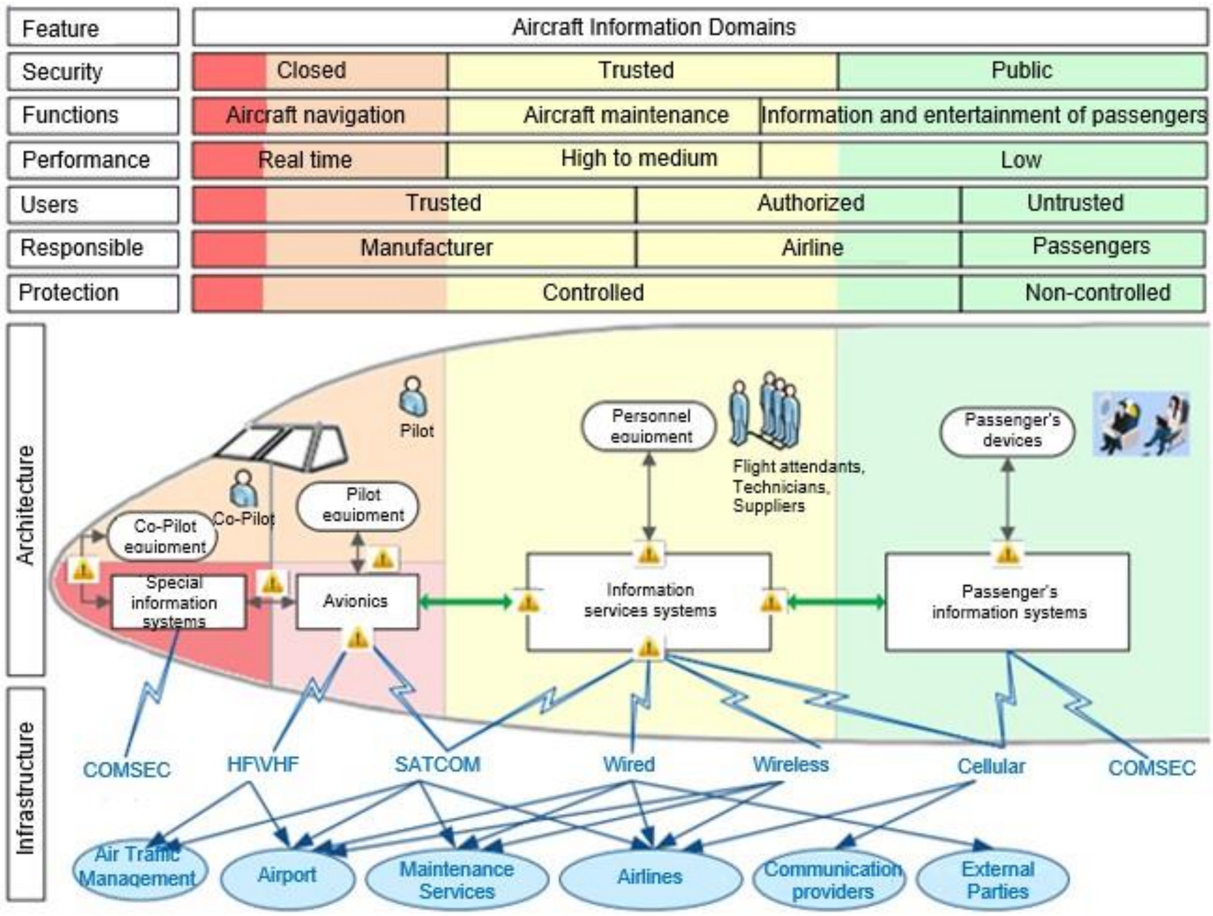


Figure 1. Aircraft software complexity trends [3]

Distributed and integrated principles of constructing on board information infrastructure based on an open network standards and a unified computing platform also increase the degree of both, internal and external informational data exchange of the aircraft. Internal and external connections are constantly increasing due to the increasing bandwidth of data networks, memory, storage, speed and performance of processors, while reducing the footprint, weight and cost of components. Because of such integration with external connections (public networks, supplier's maintenance information systems, etc.), aircraft onboard equipment should receive and send many different signals to the outside environment, significantly increasing the degree of impact on potential information security vulnerability [3].

Reducing weight cost of the IT equipment, as well as high operation and integration requirements to its effective data exchange on board and beyond, lead to the needs to the aircraft information infrastructure divide into several logical domains with different degrees of security (Figure 2):

- closed: aircraft control domain;
- trusted: domain of the aircraft information services;
- public: domain of the onboard entertainment and passenger’s information system [3].



Feature	Aircraft Information Domains		
Security	Closed	Trusted	Public
Functions	Aircraft navigation	Aircraft maintenance	Information and entertainment of passengers
Performance	Real time	High to medium	Low
Users	Trusted	Authorized	Untrusted
Responsible	Manufacturer	Airline	Passengers
Protection	Controlled	Non-controlled	

Figure 2. Aircraft information architecture and infrastructure [3]

The Closed aircraft control domain has a highest level of trust and includes flight control systems, navigation and radio systems, as well as other systems that operate in a highly reliable integrated modular avionics environment. Its architecture consists of two subdomains: an avionics domain and a pilot (co-pilot) domain [3].

The avionics domain the most important and the most secure domain that includes all critical systems for reliable aircraft control. It has the highest level of security requirements and consists of systems and networks, the main functions of which are to ensure the safe and efficient operation of the aircraft. All systems that are not part of the avionics domain can be unified into one information and computing space, conventionally called the external environment [3].

The pilot (co-pilot) domain includes cockpit information and control systems, which allow the crew to interact with the aircraft avionics. It also contains a passenger's compartment management system, which performs some operations functions, such as: monitoring the environment in the cabin, information requests to passengers, smoke detection, etc.) [3].

The aircraft information services domain provides information for maintenance and technical personnel in order to provide a secure communication between independent aircraft domains: avionics, passenger entertainment systems and any external networks. It includes an aircraft service domain, which provides operational and administrative information for the aircraft crew (service and technical), as well as a passenger support domain, which provides information to the passenger's information system [3].

The onboard entertainment and passenger's information system domain provides information and entertainment services to passengers. A domain might contain multiple systems from different vendors, sometimes interconnected with each other. Usually its perimeters do not correspond to the boundaries of physical devices. In addition to traditional entertainment systems, it might also include systems for connecting to passenger devices, flight information systems, broadband television, communication and messaging systems, and information server functions that provide services to passengers [3].

The passenger information system domain provides passengers with the necessary information and allows them to control the cabin through the flight attendant panel (lights, seat drives, personnel call system, etc.), use onboard wireless and cellular communications, connect mobile phones, tablets and laptops to the network. Only those devices that passengers could carry on board are included in the passenger's device domain. The devices could be connected to each other via the wireless network or by the other means [3].

1.4. Ensuring information security at the aircraft design and development stage

Information security at the aircraft design and development stage is carried out by improving the technological purity of design processes in accordance with regulatory framework. This process consists of three interrelated procedures:

- development of the information security requirements;
- software and hardware development;
- integration and testing [3].

At the design stage, information security should rely on end-to-end design technologies, including those using automated tools [3].

The development of detailed information security requirements is a top-down approach, since the distribution of requirements is made from the highest level (aircraft requirements) to the lowest detailed level (software and hardware requirements). In order to meet all the requirements, it is carried out a preliminary information security assessment of the aircraft and its information infrastructure, software and hardware potential information security threats and possible sources of their occurrence are identified and analyzed. This allow to link together all levels of information security requirements - aircraft, systems, software and hardware [3].

In order to automate and secure the aircraft software development process life cycle the following tools are mostly applying:

- a compiler from the C language (guarantee of using only safe optimizations, preserving the code structure for accurate analysis of test coverage);
- static and dynamic analysis tools;
- deductive verification tools C programs;
- formal inspection;
- unit and integration testing;
- analysis of the coverage and characteristics of the code, etc. [3].

As this thesis assumption suggested that the programming languages, methods and tools used at the design and development stage of the G950 NXi Integrated Flight Deck software meet all applicable security requirements and because of that does not include into the analysis scope.

1.5. Ensuring information security during the aircraft operation stage

The general goal of the onboard information security system is to confirm that the probability of the realization of information security threats through all possible scenarios operation stage are at an acceptable level of risks [3].

Ensuring secure and effective integration of onboard, air and ground networks is carried out by dividing the information and computing space of the aircraft by levels of trust into controlled secure domains with varying degrees of security and by disposing additional protections means among them (Figure 3): - onboard secure gateway;

- onboard secure servers.

By grouping onboard information systems into secure domains a clear boundary is established between highest security requirements areas and lower level of trust zones, which are able to interact with public networks. Such segregation allows to reduce the number of potential threats, which could harm the vital aircraft systems [3].

Secure controlled domains and additional means of protection between them continuously receive data packets and indicators of compromise from the network for traffic characteristics anomalies identification by an intelligent information security threat detection algorithm, which determines whether the analyzed data is secure [3].

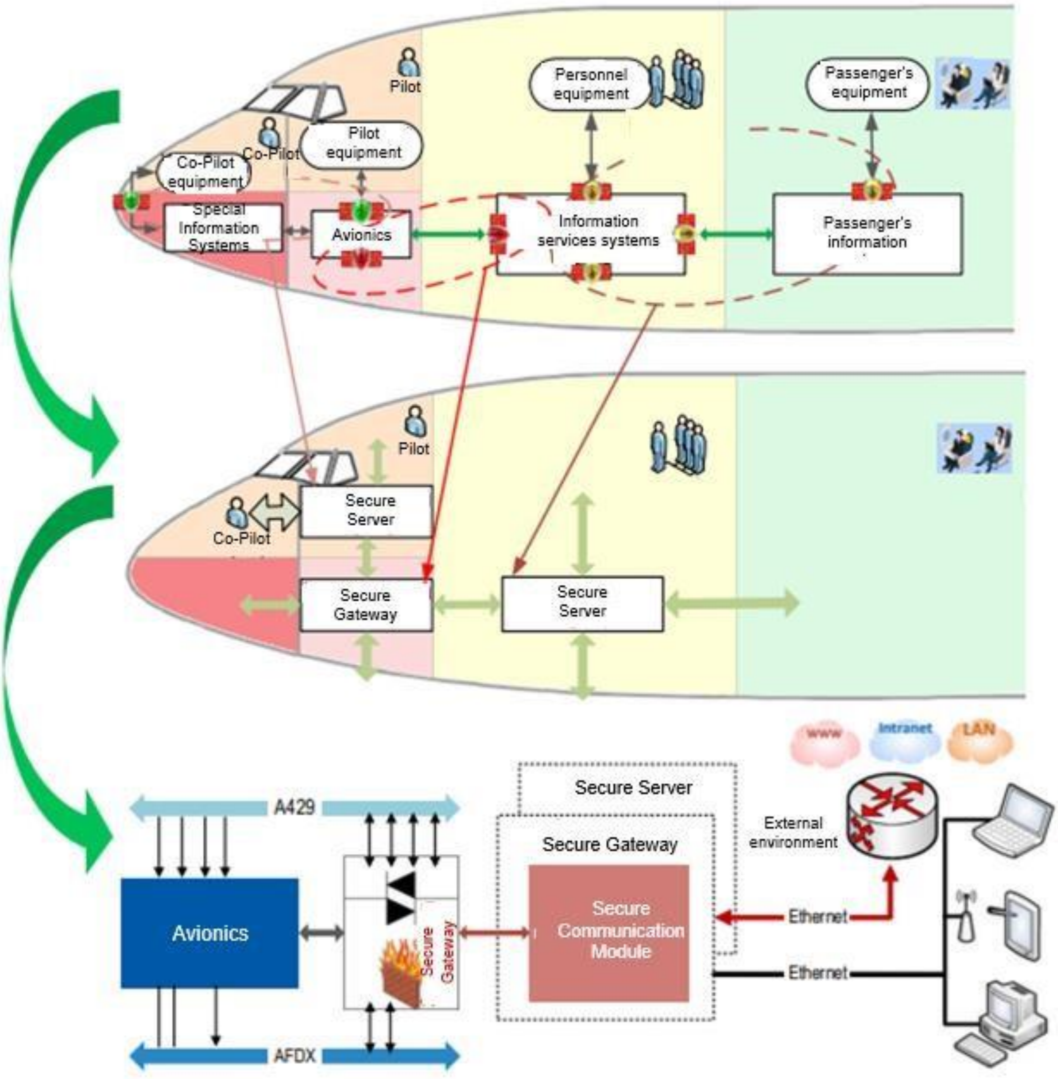


Figure 3. Aircraft information security architecture [3]

2. Aircraft onboard software and its maintenance

2.1. Aircraft software description and system interconnection

Aircraft software includes aircraft information management and command control systems, and platforms (such as embedded real-time operating system), on which these systems relied, etc. Modern avionics systems transit from electronic machinery-intensive to software-intensive, and software plays an increasingly important role in implementing safety critical function, controlling and eliminating the hazards. In recent years, the loss of life, property, and other major disasters caused by software fault in the field of aviation are presented upward trend [5].

The Tecnam P2006T aircraft Information services domain basis element is the G950 NXi Integrated Flight Deck system, which is an advanced technology avionics software suitably designed to integrate pilot/aircraft interaction into one central system. The system combines primary flight instrumentation, aircraft systems instrumentation, and navigational information, all displayed on two large color screens. The G950 NXi Integrated Flight Deck system is composed of several sub-units or Line Replaceable Units (LRUs). LRUs have a modular design and can be installed directly behind the instrument panel or in a separate avionics bay if desired. This design greatly eases troubleshooting and maintenance of the system. A failure or problem can be isolated to a particular LRU, which can be replaced quickly and easily. Each LRU has a particular function, or set of functions, that contributes to the system’s operation.

The G950 NXi Integrated Flight Deck system is distributed across the following Line Replaceable Units:

- GDU 1040 Primary Flight Display (PFD);
- GDU 1040 Multi-Function Display (MFD);
- GMA 1347 Audio Panel with Integrated Marker Beacon Receiver:
- GIA 63W Integrated Avionics Units (IAU);
- GDC 74A Air Data Computer (ADC);
- GTX 33 Mode S Transponder;
- GRS 77 Attitude and Heading Reference System (AHRS);

- GMU 44 Magnetometer [5].

The interaction between the Line Replaceable Units is shown on Figure 4.

Common features of the System are noted below:

- Hardware / Software Configuration: a standard set of supported configurations with common LRU part numbers. Although individual LRUs will utilize common TSO authorized software, separate system software loader images will be released for G950 NXi Integrated Flight Deck system;
- Engine Indicating System (EIS) Support: provide the installer the option of configuring the system with or without engine instrumentation display. In the event that G950 system is configured as a “No EIS” system (and separate external engine indications are used), then additional flight plan information is placed in the location normally occupied by the EIS strip;
- Crew Alerting System (CAS): a common set of engine and airframe integration alerts exist for G950 systems;
- Autopilot Support: system provide outputs to a separate 3rd party autopilot, although supported autopilots may vary [5].

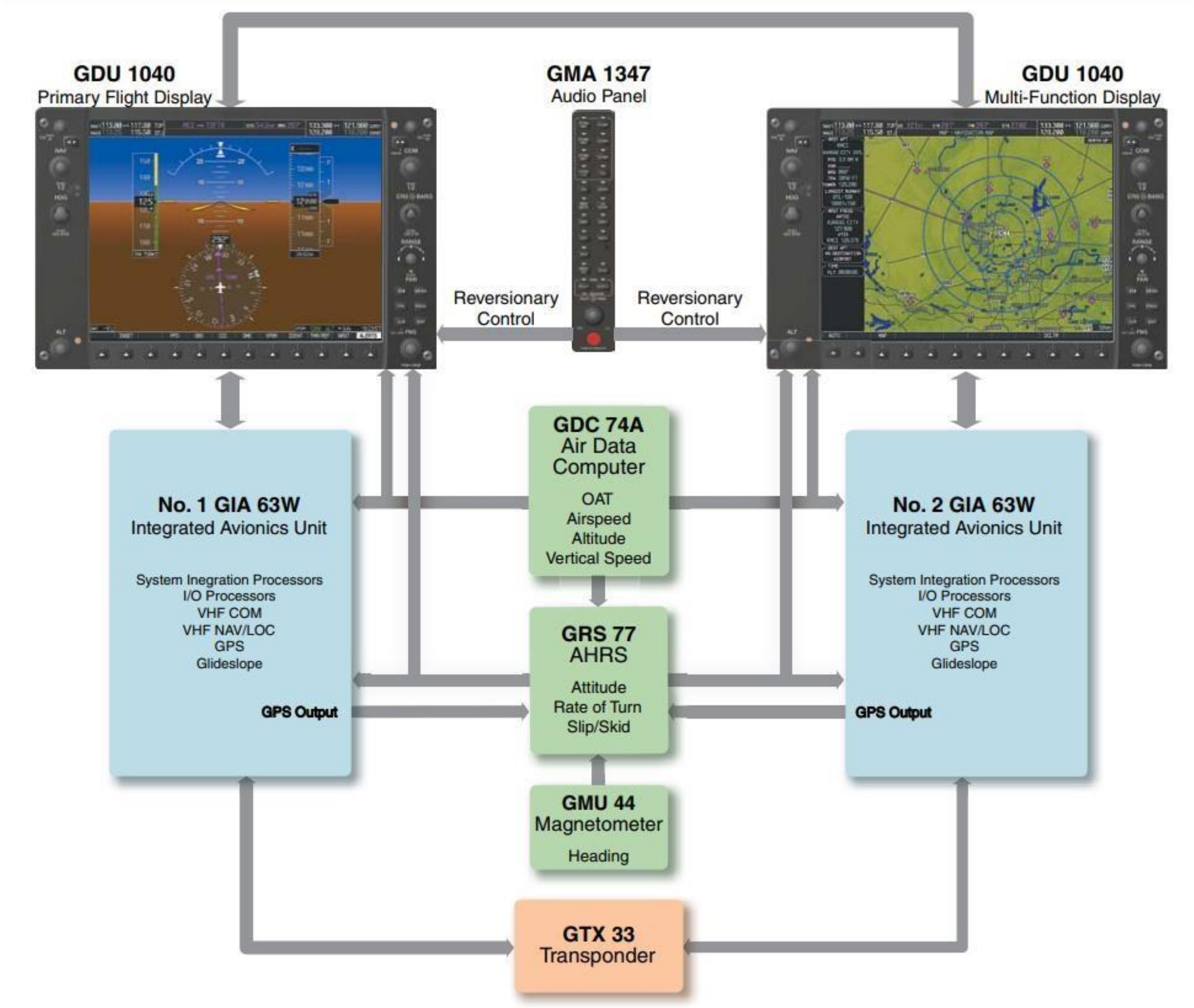


Figure 4. Interactions between the Line Replaceable Units (5)

The Figure 5 illustrate the redundant communication paths that are in place in a G950 NXi Integrated Flight Deck System installation

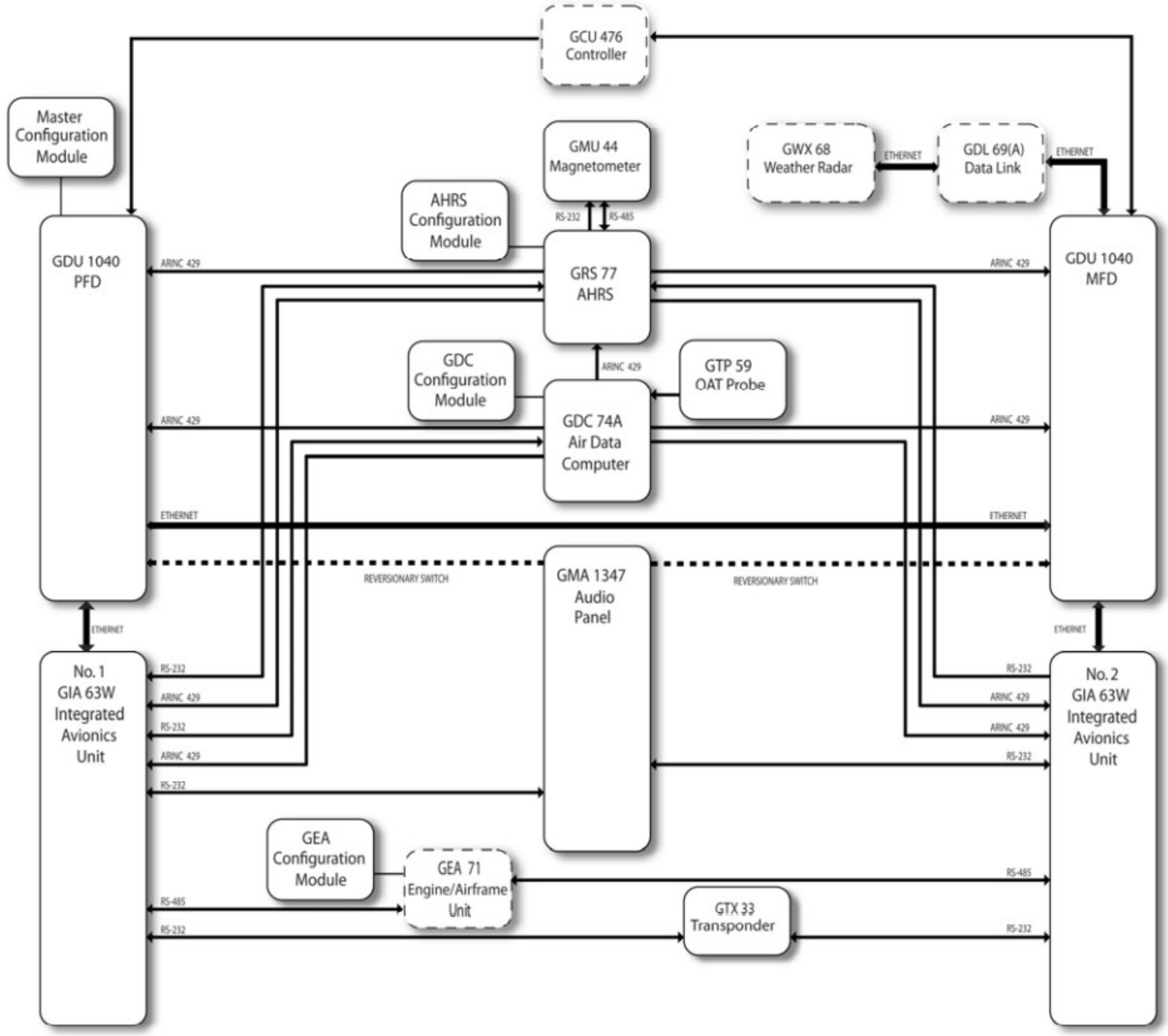


Figure 5. G950 NXi Integrated Flight Deck System Interconnection Diagram [6].

2.1.2 Primary Flight Display and Multi-Function Display

GDU 1040 – the left-hand GDU is configured as a Primary Flight Display (PFD) and the right-hand GDU as a Multi-Function Display (MFD). Both feature 10.4-inch LCD screens with 1024 x 768 resolution. The displays communicate with each other through a High-Speed Data Bus (HSDB) Ethernet connection. Each display is also paired with an Ethernet connection to an IAU [5].

In the event of a display failure, the G950 NXi System automatically switches to reversionary (backup) mode. In reversionary mode, all important flight information is presented on the remaining display(s) in the same format as in normal operating mode. PFD failure – MFD enters reversionary mode and vice verso. If a display fails, the appropriate IAU-display Ethernet interface is cut off. Thus, the IAU can no longer communicate with the remaining display, and the NAV and COM functions provided to the failed display by the IAU are flagged as invalid on the remaining display. The system reverts to backup paths for the AHRS, ADC, Engine/Airframe Unit, and Transponder, as required. The change to backup paths is completely automated for all Line Replaceable Units and no pilot action is required [5].

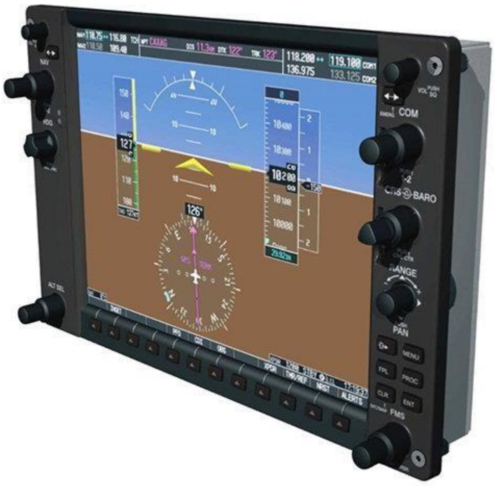


Figure 6. GDU 1040 [5].

2.1.2 The Audio Panel

GMA 1347 – The Audio Panel integrates navigation/communication radio (NAV/COM) digital audio, intercom, and marker beacon controls, and is installed between the displays. This unit also provides manual control of display reversionary mode and communicates with both IAUs using an RS-232 digital interface [5].

Figure 7. GMA1347 [5].

2.1.3 The Integrated Avionics Units

GIA 63W – The Integrated Avionics Units (IAU) function as the main communication hubs, linking all Line Replaceable Units with the on-side display. Each IAU contains a GPS WAAS receiver, VHF COM/NAV/GS receivers, and system integration microprocessors, and is paired with the on-side display via HSDB connection. The IAUs are not paired together and do not communicate with each other directly (5)

Figure 8. GIA 63W [5].

2.1.4 The Air Data Computer

GDC 74A – The Air Data Computer (ADC) processes data from the pitot/static system and outside air temperature (OAT) sensor. The ADC provides pressure altitude, airspeed, vertical speed, and OAT information to the G950 System, and it communicates with the primary IAU, displays, and AHRS using an ARINC 429 digital interface [5].

Figure 9. GDS 74A [5].

2.1.5 The solid-state Transponder

GTX 33 – The solid-state Transponder provides Modes A, C, and S capability and communicates with both IAUs through an RS-232 digital interface [5].

Figure 10. GTX 33 [5].

2.1.6 The Attitude and Heading Reference System

GRS 77 – The Attitude and Heading Reference System (AHRS) provides aircraft attitude and heading information via ARINC 429 to both PFDs and the primary IAU. The AHRS contains advanced sensors (including accelerometers and rate sensors) and interfaces with the Magnetometer to obtain magnetic field information, with the ADC to obtain air data, and with both IAUs to obtain GPS information [5].

Figure 11. GRS 77 [5].

2.1.7 The Magnetometer

GMU 44 – The Magnetometer measures local magnetic field and sends data to the AHRS for processing to determine aircraft magnetic heading. This unit receives power directly from the AHRS and communicates with it via an RS-485 digital interface [5].

Figure 12. GMU 44 [5].

2.1.8 SiriusXM satellite radio

GDL 69/69A (if installed) – A SiriusXM satellite radio receiver that provides real-time weather information to the MFD (and, indirectly, to the inset map of the PFD) as well as digital audio entertainment. The GDL 69A communicates with the MFD via HSDB connection. A subscription to the SiriusXM Satellite Radio service is required to enable the GDL 69A audio entertainment capability [5].

Figure 13. GDL 69/69A [5].

2.2. Aircraft software and related maintenance processes

In accordance with distributed and integrated principles of constructing onboard information infrastructure based on an open network standards and a unified computing platform described previously the aircraft information infrastructure is rational to divide into several logical domains with different degrees of security [3].

The most effective method of ensuring security of an aircraft information infrastructure is the division of onboard equipment into secure domains in order to clearly define boundaries where the information exchange must meet the highest security requirements, while other domains can have a lower level of trust and interact with public networks without impact of potential threats to critical aircraft systems [3].

Thus, the Tecnam P2006T aircraft Information services domain foundation is the G950 NXi Integrated Flight Deck software, which belongs to the trusted domain that address the most probable risks related with business and operational environment of the aircraft software maintenance. When this software is updated or service during the maintenance procedures by technical personnel who perform an aircraft maintenance services, external devices such as computers or SD memory cards is directly connected to the aircraft information infrastructure. At this moment, potential threats could penetrate the software and compromise its integrity and availability.

G950 NXi Integrated Flight Deck software could be also put at risk by its software developers with inadequate information security management.

G950 NXi Integrated Flight Deck software components belongs to Closed aircraft control domain, which is out of scope of this analysis because of extremely improbable existence of potential vulnerabilities implemented during development and testing stage of the software development life cycle, as well as Public domain information security of the onboard entertainment and passenger's information system – because it is not applicable for Tecnam P2006T aircraft.

As the results of the Tecnam P2006T aircraft G950 NXi Integrated Flight Deck software maintenance process overview were identified the following evidence that allow to identify some nonconformities in the process concept and to trace them back to system losses:

a) The appropriate Software for regular updates downloaded from my.garmin.com website by authorized personnel based on user ID and password, received in accordance with agreement with Jeppesen Company. The installation procedure is carried out in accordance with Capitol 15 Software, Configuration, and Calibration of the

G900X/G950 Installation and Maintenance Manual (revision D) procedure [6];

b) The G950 NXi Integrated Flight Deck software configuration provides the installer with a means of configuring, checking, and calibrating various G900X/G950 sub-systems under configuration mode. Troubleshooting/diagnostics information could also be derived from this mode;

c) The dedicated SanDisk SD cards is used for aviation database and system software updates as well as terrain database storage download and storage;

d) The SD cards is stored in a safe place, inaccessible to electromagnetic interference with very limited access for unauthorized personnel;

e) The G950 NXi Integrated Flight Deck software update and maintenance is done by dedicated personnel with appropriate level of IT knowledge and technical experience;

f) The computer, used for maintenance procedure, has supported by Microsoft operating system Windows 10 with actual security updates installed. The control for the new security updated download is scheduled weekly at the automate mode;

g) The endpoint and response system installed on computer, used for maintenance procedure, has actual antivirus database installed. The control for the new security update download is scheduled daily at the automate mode;

h) As an available option, the G950 NXi Integrated Flight Deck software update and maintenance could be done via Bluetooth short-range wireless technology standard.

i) The G950 NXi Integrated Flight Deck software update, reconfiguration and preventive maintenance and diagnostic is carried out by directly connected SanDisk SD cards with previously copied on it information into dedicated interface of the Integrated Avionics Units.

j) Prior to maintenance procedure is required to ensure that the G950 NXi System is powered off before inserting an SD cards.

3. The systemic approach to aircraft safety and security

3.1. Overview of the STPA method

STPA is a relatively new hazard analysis technique based on an extended model of accident causation [7].

STPA is a hazard analysis technique based on an extended model of accident causation. In addition to component failures, STPA assumes that accidents can also be caused by unsafe interactions of system components, none of which may have failed [7].

In comparison with the other hazard analysis methods, STPA is able to identify many more causal scenarios, often software-related and non-failure, scenarios that the traditional methods did not find [7].

The steps in basic STPA are shown in Figure 14 along with a graphical representation of these steps [7].

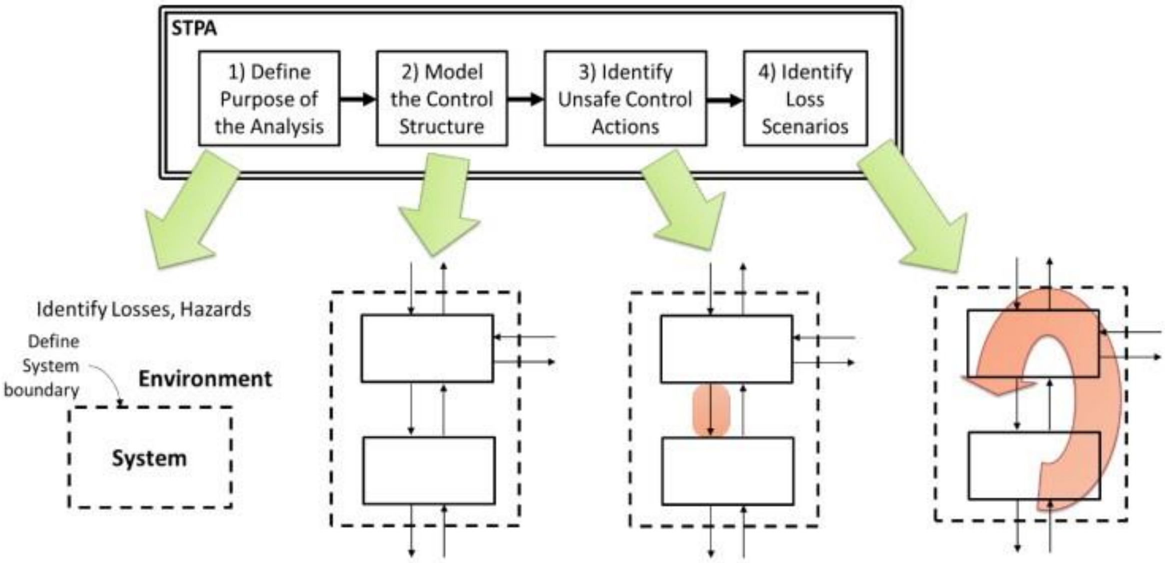

STPA
1) Define Purpose of the Analysis
2) Model the Control Structure
3) Identify Unsafe Control Actions
4) Identify Loss Scenarios
Identify Losses, Hazards
Define System boundary
Environment
System

Figure 14. Overview of the basic STPA Method (7)

Defining the purpose of the analysis is the first step with any analysis method. What kinds of losses will the analysis aim to prevent? Will STPA be applied only to traditional safety goals like preventing loss of human life or will it be applied more broadly to security, privacy, performance, and other system properties? What is the system to be analyzed and what is the system boundary? These and other fundamental questions are addressed during this step [7].

The second step is to build a model of the system called a control structure. A control structure captures functional relationships and interactions by modeling the system as a set of feedback control loops.

The control structure usually begins at a very abstract level and is iteratively refined to capture more detail about the system. This step does not change regardless of whether STPA is being applied to safety, security, privacy, or other properties [7].

The third step is to analyze control actions in the control structure to examine how they could lead to the losses defined in the first step. These unsafe control actions are used to create functional requirements and constraints for the system. This step also does not change regardless of whether STPA is being applied to safety, security, privacy, or other properties [7].

The fourth step identifies the reasons why unsafe control might occur in the system. Scenarios are created to explain:

a) How incorrect feedback, inadequate requirements, design errors, component failures, and other factors could cause unsafe control actions and ultimately lead to losses;

b) How safe control actions might be provided but not followed or executed properly, leading to a loss [7].

Once scenarios are identified, they can be used to create additional requirements, identify mitigations, drive the architecture, make design recommendations and new design decisions, evaluate/revisit existing design decisions and identify gaps, define test cases and create test plans, develop leading indicators of risk, and for other uses (7).

The process of safety requirements elicitation based on STPA are shown in Figure 15 along with a graphical representation of these steps.

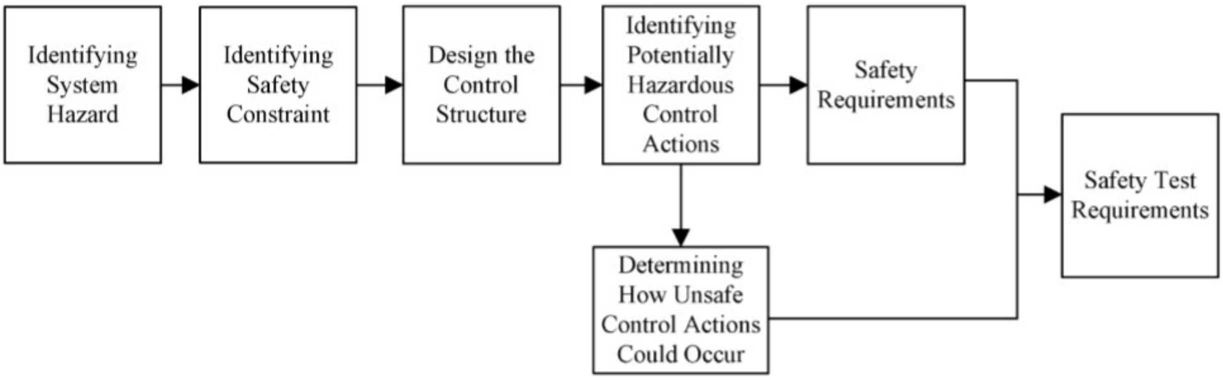

Identifying System Hazard
Identifying Safety Constraint
Design the Control Structure
Identifying Potentially Hazardous Control Actions
Safety Requirements
Safety Test Requirements
Determining How Unsafe Control Actions Could Occur

Figure 15. Process of safety requirements elicitation based on STPA [7]

4. Information security evaluation of the aircraft software

4.1 The purpose of the STPA analysis

The system purpose and goal is to ensure security of the onboard information infrastructure exchange by maintenance procedure in order to provide the pilot with correct flight instrumentation, position, navigation, communication, and identification information.

The system boundary includes G950 NXi Integrated Flight Deck System Software and its Line Replaceable Units, local and supplier level 2-3 maintenance personnel, as well as correspond hardware and software tools.

The following list of losses that required to avoid due to analysis:

L1: Loss of life or serious injury to people;

L2: Damage to the aircraft or objects outside the aircraft;

L3: Loss of or damage of equipment of the aircraft;

L4: Inability to complete the software maintenance mission.

L5: Loss of flight databases

The following list of hazards related to these losses:

H1: Compromised Maintenance Hardware

H2: Maintenance procedure security requirement violation

H3: Security requirements for communication interfaces protection violation

H4: Software integrity violation;

H5: Presence of errors in flight databases

A system condition that will lead to a loss in worst-case scenario conditions and that need to be satisfied to prevent hazards:

Hazard	Description	Worst case scenario	Associated Losses	Constraints
H1: Compromised Maintenance Hardware	Computer and SD cards might be used for other purposes than maintenance. That allows malware infiltrate to above mentioned hardware using the well-known technical vulnerabilities and compromised it.	Software damage by malware compromise flight databases integrity	L4, L5	Computer and SD cards used for maintenance must satisfy security standards \ best practices for patching and malware protection
H2: Maintenance procedure security requirement violation	Insufficient technicians experience, knowledge, lack of dual control, time shortage for maintenance, lack of electrostatic discharge safety could harm aircraft software removable media due to unintentional errors and omissions of the maintenance personnel	Entering inaccurate information to the aircraft software or removable media chip destroy	L2 - L4	Maintenance procedure must follow the detailed instructions from installation and maintenance guide

H3: Security requirements for communication interfaces protection violation	Bluetooth technology could be optionally used for aircraft software access and open by default even if never use. The malware from pilot smartphone or portable computer might infiltrate to the aircraft software and damage it.	Software damage by malware compromise flight databases integrity	L2 – L5	The Bluetooth communication interface must be deactivated all the time when not in use for maintenance
H4: Software integrity violation	The software integrity might be compromise from download from the fake site as a result of cross site forgery attack	Entering inaccurate information to the aircraft software	L1, L2, L4,L5	Updates must be loaded from the trust source by security awareness trained personnel including phishing countermeasures
H5: Presence of errors in flight databases	Out of date or inaccurate information about aviation, terrain, and system condition might be transfer to the aircraft software due to scam fraud	Piloting errors or difficulties due to lack of flight information	L1 – L5	Updates must be loaded from the trust source by security awareness trained personnel including social engineering countermeasures

Table 2. A system condition for loss, hazards, and system-level constraints

4.2 The control structure model

An effective control structure of the onboard software security in aircraft maintenance process is designed to enforce constraints on the behavior of the overall system. A control structure composition is shown in Figure 18 below:

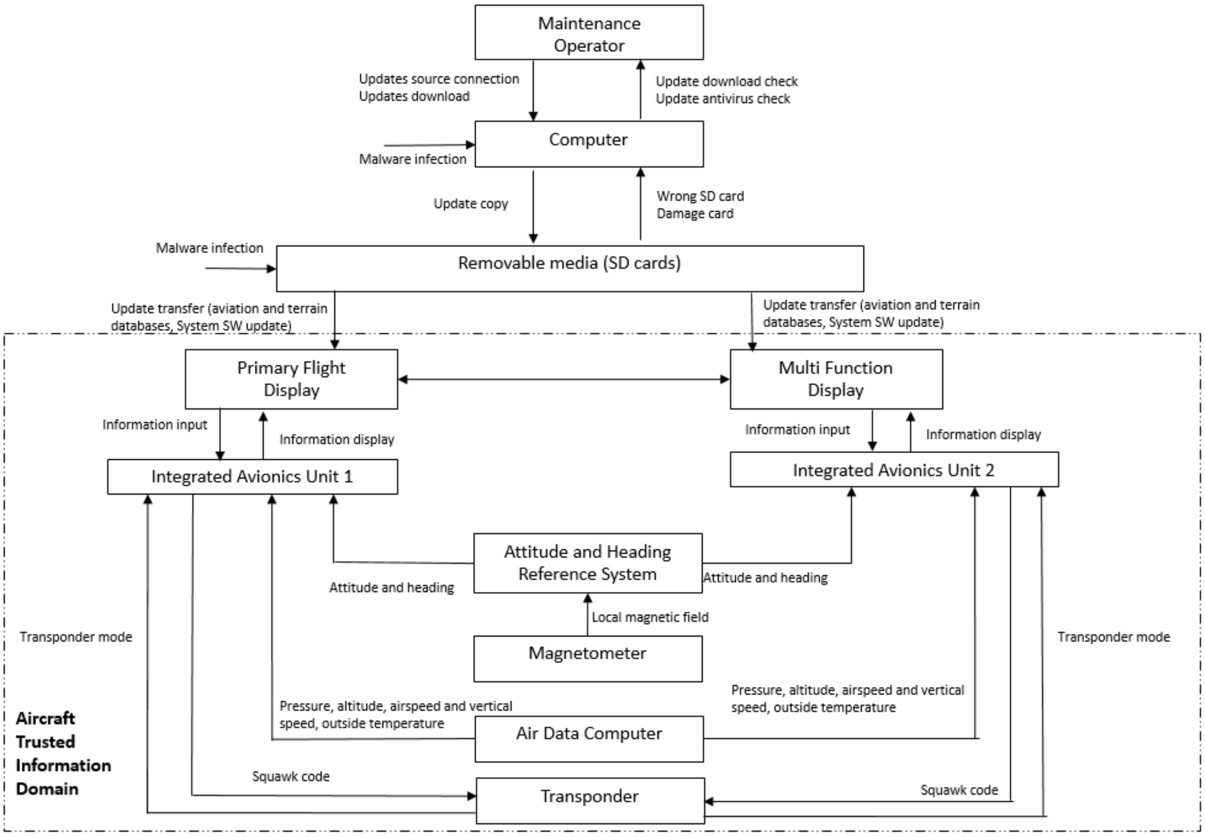

Maintenance Operator
Updates source connection
Updates download
Update download check
Update antivirus check
Computer
Malware infection
Update copy
Wrong SD card
Damage card
Malware infection
Removable media (SD cards)
Update transfer (aviation and terrain databases, System SW update)
Update transfer (aviation and terrain databases, System SW update)
Primary Flight Display
Multi Function Display
Information input
Information display
Information input
Information display
Integrated Avionics Unit 1
Integrated Avionics Unit 2
Attitude and Heading Reference System
Attitude and heading
Attitude and heading
Local magnetic field
Transponder mode
Magnetometer
Transponder mode
Pressure, altitude, airspeed and vertical speed, outside temperature
Aircraft Trusted Information Domain
Pressure, altitude, airspeed and vertical speed, outside temperature
Air Data Computer
Squawk code
Transponder
Squawk code

Figure 18. Control structure composition

4.3 Unsecure Control Actions

An Unsecure Control Action (UCA) is a control action that, in a particular context and worst-case environment, will lead to a hazard. The inputs and outputs for UCA are shown at the Figure 19 [7].

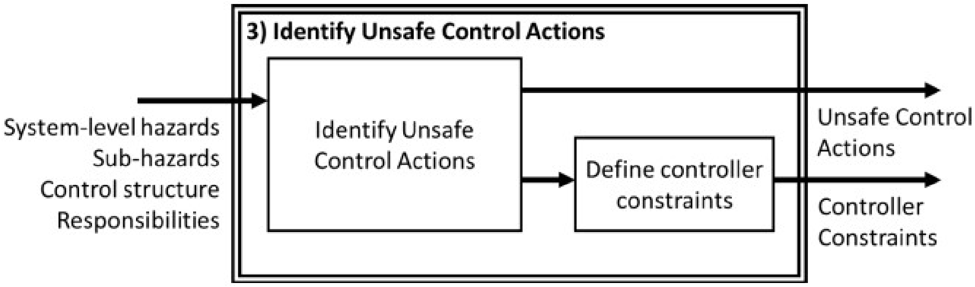


Figure 19. Overview of UCA analysis [7].

Control Action	Not providing causes hazard	Providing causes hazard	Too early, too late, wrong order	Stopped too soon, applied too long
Update download	UCA1: Maintenance operator did not provide update download when the previous version expired [H5]	UCA2: Maintenance operator provided update download from unofficial source with compromised update [H1, H4]	UCA3: Maintenance operator provided update too late when it is already out of date [H5] UCA4: Maintenance operator provided update too early before official release while it still had bugs [H1, H5]	UCA5: Maintenance operator stopped too soon the update download due to interruption during downloading [H4, H5]

Update copy	UCA6: Computer does not provide update copy when it is necessary [H2]	UCA7: Computer provides update copy with malware infection [H4]	UCA8: Computer provided update copy too late when the installed software already expired [H5]	UCA9: Computer update copy stopped too soon due to interruption during downloading to the hardware [H4, H5]
Update transfer	UCA10: Removable media did not provide update transfer when inserted in the slot and requested by the software [H5]	UCA11: Removable media provided update transfer with malware during information transfer to the software [H4, H5]	UCA12: Removable media provided update transfer too late due to its malfunction when the software already expired [H5]	UCA13: Removable media update transfer stopped to soon due to SD card extraction during transfer process [H4, H5]
Information input	UCA14: Primary Flight Display did not provide information input during avionics preparation phase [H5]	UCA15: Primary Flight Display provided information input with erroneous data during insertion [H5]	UCA16: Primary Flight Display provided information input too late during avionics setup [H5] UCA17: Primary Flight Display provided input	UCA18: Primary Flight Display information input stopped too soon due to interruption during avionics setup [H5]

			information in wrong order during avionics setup [H4, H5]	
Squawk code	UCA19: Integrated Avionics Unit did not provide squawk code when inserted [H4]	UCA20: Integrated Avionics Unit provided incorrect squawk code (other than inserted) when entered [H4]	UCA21: Integrated Avionics Unit provided too late the squawk code when entered [H4]	UCA22: Integrated Avionics Unit stopped too soon providing the squawk code during critical phases of flight [H4] UCA23: Integrated Avionics unit stopped too late providing the old (previous) squawk code during critical phases of flight [H4]

Table 3 shows UCAs for the Aircraft information services domain controller

The Table 4 below shows translated into constraints on the behavior of each controller identified UCAs.

Unsecure Control Actions	Controller Constraints
UCA1: Maintenance operator did not provide update download when the previous version expired [H5]	C1: Maintenance operator must provide update download when the previous version expired [UCA1]

UCA2: Maintenance operator provided update download from unofficial source with compromised update [H1, H4]	C2: Maintenance operator must not provide update download from unofficial source with compromised update [UCA2]
UCA3: Maintenance operator provided update too late when it is already out of date [H5]	C3: Maintenance operator must not provide update too late when it is already out of date [UCA3]
UCA4: Maintenance operator provided update too early before official release while it still had bugs [H1, H5]	C4: Maintenance operator must not provide update too early before official release while it still has bugs [UCA4]
UCA5: Maintenance operator stopped too soon the update download due to interruption during downloading [H4, H5]	C5: Maintenance operator must not stop too soon the update download due to interruption during downloading [UCA5]
UCA6: Computer does not provide update copy when it is necessary [H2]	C6: Computer must provide update copy when it is necessary [UCA6]
UCA7: Computer provides update copy with malware infection [H4]	C7: Computer must not provide update copy with malware infection [UCA7]
UCA8: Computer provided update copy too late (1 day) when the installed software already expired [H5]	C8: Computer must not provide update copy too late (>1 day) when the installed software already expired [UCA8]
UCA9: Computer update copy stopped too soon due to interruption during downloading to the hardware [H4, H5]	C9: Computer update copy must not stop too soon due to interruption during downloading to the hardware [UCA9]
UCA10: Removable media did not provide update transfer when inserted in the slot and requested by the software [H5]	C10: Removable media must provide update transfer when inserted in the slot and requested by the software [UCA10]
UCA11: Removable media provided update transfer with malware during information transfer to the software [H4, H5]	C11: Removable media must not provide update transfer with malware during information transfer to the software [UCA11]
UCA12: Removable media provided update transfer too late due to its malfunction when the software already expired [H5]	C12: Removable media must not provide update transfer too late due to its malfunction when the software already expired [UCA12]

UCA13: Removable media update transfer stopped to soon due to SD card extraction during transfer process [H4, H5]	C13: Removable media update transfer must not be stopped to soon due to SD card extraction during transfer process [UCA13]
UCA14: Primary Flight Display did not provide information input during avionics preparation phase [H5]	C14: Primary Flight Display must provide information input during avionics preparation phase [UCA14]
UCA15: Primary Flight Display provided information input with erroneous data during insertion [H5]	C15: Primary Flight Display must not provide information input with erroneous data during insertion [UCA15]
UCA16: Primary Flight Display provided information input too late during avionics setup [H5]	C16: Primary Flight Display must not provide information input too late during avionics setup [UCA16]
UCA17: Primary Flight Display provided input information in wrong order during avionics setup [H4, H5]	C17: Primary Flight Display must not provide input information in wrong order during avionics setup [UCA17]
UCA18: Primary Flight Display information input stopped too soon due to interruption during avionics setup [H5]	C18: Primary Flight Display information input must not be stopped too soon due to interruption during avionics setup [UCA18]
UCA19: Integrated Avionics Unit did not provide squawk code when inserted [H4]	C19: Integrated Avionics Unit must provide squawk code when inserted [UCA19]
UCA20: Integrated Avionics Unit provided incorrect squawk code (other than inserted) when entered [H4]	C20: Integrated Avionics Unit must not provide incorrect squawk code (other than inserted) when entered [UCA20]
UCA21: Integrated Avionics Unit provided too late the squawk code when entered [H4]	C21: Integrated Avionics Unit must not provide too late the squawk code when entered [UCA21]
UCA22: Integrated Avionics Unit stopped too soon providing the squawk code during critical phases of flight [H4]	C22: Integrated Avionics Unit must not stop too soon providing the squawk code during critical phases of flight [UCA22]
UCA23: Integrated Avionics unit stopped too late providing the old (previous) squawk code during critical phases of flight [H4]	C23: Integrated Avionics unit must not stop too late providing the old (previous) squawk code during critical phases of flight [UCA23]

Table 4. UCAs and its Controller constraints

4.4 Loss scenarios

A loss scenario describes the causal factors that can lead to the UCAs and to hazards. Table 5 below shows scenarios how the control algorithm may cause the UCAs.

UCA1: Maintenance operator did not provide update download when the previous version expired [H5]
Scenario 1 for UCA1: Maintenance operator did not provide update download [UCA1] due to absence of the reminder. As a result, the database in the airplane is out of date [H5] Scenario 2 for UCA1: Maintenance operator did not provide update download [UCA1] because he is not authorized for such work. As a result, the organization doesn't have a qualified person to conduct the work and the airplane database will not be updated [H5] Scenario 3 for UCA1: Maintenance operator did not provide update download [UCA1] because he incorrectly believes that he already did this. This flawed process model will occur if the received feedback shows that the download already exists. Such a feedback may be caused by a malware on computer leading to not providing update when previous version expired [H5]
UCA2: Maintenance operator provided update download from unofficial source with compromised update [H1, H4]
Scenario 1 for UCA2: Maintenance operator downloaded the update from an unofficial source [UCA2] because the company did not specify which source to be used. As a result, the downloaded update was compromised [H1, H4] Scenario 2 for UCA2: Maintenance operator provided update download from unofficial source [UCA2] because he believes the source he is using is the correct one. This flawed process will occur if the received feedback from the antivirus does not indicate any problem. Such a feedback may be caused by lack of/or expired software protection leading to compromised update [H1, H4]
UCA3: Maintenance operator provided update too late when it is already out of date [H5]
Scenario 1 for UCA3: Maintenance operator provided update too late after database expiration [UCA3] because the company did not mention when the update should be downloaded. As a result, the database in the airplane is out of date [H5]

Scenario 2 for UCA3: Maintenance operator provided update too late after database expiration [UCA3] because he incorrectly believes that downloaded update is still current. This flawed process model will occur if the received feedback shows incorrect update version. Such a feedback may be caused by a malware on the computer that changed the file version [H5]
UCA4: Maintenance operator provided update too early before official release while it still had bugs [H1, H5]
Scenario 1 for UCA4: Maintenance operator provided the update before official release [UCA4] because he downloaded it at variable periods. Such an action would be possible if the company did not specify the period boundaries. As a result, the update could contain incomplete information [H1, H5] Scenario 2 for UCA4: Maintenance operator provided the update before official release [UCA4] because he incorrectly considers that the update is reliable. This flawed process will occur if the received feedback indicates no problems. Such a feedback may be caused by inability of the computer software to check the update on integrity [H1, H5]
UCA5: Maintenance operator stopped too soon the update download due to interruption during downloading [H4, H5]
Scenario 1 for UCA5: Maintenance operator update downloading was interrupted [UCA5] because of electrical and/or internet blackouts. As a result, the update integrity was violated [H4, H5] Scenario 2 for UCA5: Maintenance operator interrupted the downloading process before he made sure that the process is finished [UCA5] because there was no indication. As a result, the update integrity was violated [H4, H5] Scenario 3 for UCA5: Maintenance operator update download was interrupted [UCA5] but there is no error indication. This flawed process will occur if the received feedback believes that the process was finished successfully. Such a feedback may be caused by lack of/or expired software protection leading to compromised update [H1, H4]
UCA6: Computer does not provide update copy when it is necessary [H2]
Scenario 1 for UCA6: The computer did not provide update copy when it is necessary [UCA6] because of the physical failure. As a result, the update copy cannot be performed [H2]

Scenario 2 for UCA6: The computer did not provide update copy when it is necessary [UCA6] because the computer software responsible for the process of copy is damaged. As a result, the update copy cannot be performed [H2] Scenario 3 for UCA6: The computer did not provide update copy when it is necessary [UCA6] because the computer software, after scanning the removable media, detected that it is unsatisfactory (contains virus and/or does not have enough memory space). As a result, the update copy cannot be performed [H2] Scenario 4 for UCA6: The computer did not provide update copy when it is necessary [UCA6] because the computer incorrectly believes that the copy has been already made. This flawed process model will occur if the received feedback indicates that the update is already copied to the removable media. Such a feedback may be caused by a malware on computer/removable media leading to not providing update copy when necessary [H2]
UCA7: Computer provides update copy with malware infection [H4]
Scenario 1 for UCA7: Computer provides update copy with malware infection [UCA7] because the antivirus on the computer did not detect any malware software. As a result, the update copy was infected [H4] Scenario 2 for UCA7: Computer provides update copy with malware infection [UCA7] because the removable media does not have any software to check for the infections. This flawed process model will occur if the received feedback from the removable media to computer does not indicate any inconsistencies [H2]
UCA8: Computer provided update copy too late when the installed software already expired [H5]
Scenario 1 for UCA8: Computer provided update copy too late when the installed software already expired [H5] because the computer was physically inoperable at the necessary moment. As a result, the update was out of date [H5] Scenario 2 for UCA8: Computer provided update copy too late when the installed software already expired [H5] because the process of copy took more time than usually because of the damaged software. As a result, the update was out of date [H5]
UCA9: Computer update copy stopped too soon due to interruption during downloading to the hardware [H4, H5]

Scenario 1 for UCA9: Computer update copy stopped too soon due to interruption during downloading to the hardware [UCA9] because during the process of copy the computer was shut down (failure and or electrical blackout). As a result, the process of copy wasn't completed fully [H4, H5] Scenario 2 for UCA9: Computer update copy stopped too soon due to interruption during downloading to the hardware [UCA9] because a person interrupted it forcibly. As a result, the process of copy wasn't completed fully [H4, H5] Scenario 3 for UCA9: Computer update copy stopped too soon due to interruption during downloading to the hardware [UCA9] because the software erroneously considers that the copy was completed. This flawed process model will occur if the received feedback from the hardware contains errors or is missing completely. Such a feedback may be caused by malware on the hardware leading to interruption of the copy [H4, H5]
UCA10: Removable media did not provide update transfer when inserted in the slot and requested by the software [H5]
Scenario 1 for UCA10: Removable media did not provide update transfer when inserted in the slot and requested by the software [UCA10] because of the physical damage. As a result, the update couldn't be transferred [H5] Scenario 2 for UCA10: Removable media did not provide update transfer when inserted in the slot and requested by the software [UCA10] because it was inserted in the wrong slot. Such an action may be caused by lack of knowledge and missing prescribed procedures by the company [H5] Scenario 3 for UCA10: Removable media did not provide update transfer when inserted in the slot and requested by the software [UCA10] because it cannot be detected by the PFD software [H5]
UCA11: Removable media provided update transfer with malware during information transfer to the software [H4, H5]
Scenario 1 for UCA11: Removable media provided update transfer with malware during information transfer to the software [UCA11] because the removable media antivirus is missing and/or couldn't recognize malware. Such an action would be possible if the antivirus is not updated timely or it is missing completely. As a result, the removable media transferred a malware [H4, H5]
UCA12: Removable media provided update transfer too late due to its malfunction when the software already expired [H5]

Scenario 1 for UCA12: Removable media provided update transfer too late due to its malfunction when the software already expired [UCA12] because the removable media was physically damaged. As a result, the update was transferred after the software already expired [H5]
UCA13: Removable media update transfer stopped to soon due to SD card extraction during transfer process [H4, H5]
Scenario 1 for UCA13: Removable media update transfer stopped to soon due to SD card extraction during transfer process [UCA13], because maintenance personnel wasn't instructed about the correct process of update. Such an action would be possible if the company didn't train and/or didn't specify the procedures. As a result, the update wasn't transferred completely [H4, H5] Scenario 2 for UCA13: Removable media update transfer stopped to soon due to SD card extraction during transfer process [UCA13], because the PFD erroneously believes that the transfer was done completely [H4, H5]
UCA14: Primary Flight Display did not provide information input during avionics preparation phase [H5]
Scenario 1 for UCA14: Primary Flight Display did not provide information input during avionics preparation phase [UCA14] because of the physical damage. As a result the input of the information was not possible partly or completely [H5] Scenario 2 for UCA14: Primary Flight Display did not provide information input during avionics preparation phase [UCA14] because the flight crew did not set anything. Such an action would be possible if the company didn't specify the avionics setup procedure [H5] Scenario 3 for UCA14: Primary Flight Display did not provide information input during avionics preparation phase [UCA14] although everything was done correctly by the crew because it doesn't recognize any input information. Such an action would be possible if there is a malware present in the software [H5]
UCA15: Primary Flight Display provided information input with erroneous data during insertion [H5]

Scenario 1 for UCA15: Primary Flight Display provided information input with erroneous data during insertion [UCA15] because of the physical damage of external/internal components. As a result, the PFD entered wrong data [H5] Scenario 2 for UCA15: Primary Flight Display provided information input with erroneous data during insertion [UCA15] because the flight crew entered the information incorrectly. Such an action would be possible if the company didn't specify the avionics setup procedure [H5] Scenario 3 for UCA15: Primary Flight Display provided information input with erroneous data during insertion [UCA15] because the information was entered incorrectly on purpose by an intruder. Such an action would be possible if the company doesn't have a strong security protection [H5] Scenario 4 for UCA15: Primary Flight Display provided information input with erroneous data during insertion [UCA15] because it receives wrong feedback from Integrated Avionics Unit. This flawed action would be possible if the feedback received from the sensors compromises the entered information [H5] Scenario 5 for UCA15: Primary Flight Display provided information input with erroneous data during insertion [UCA15] because the controllers erroneously believes that the entered information is correct. Such an action would be possible if there is a malware present in the software [H5]
UCA16: Primary Flight Display provided information input too late during avionics setup [H5]
Scenario 1 for UCA16: Primary Flight Display provided information input too late during avionics setup [UCA16] because of the physical damage of the information transfer channels. As a result, the information was entered with a delay [H5] Scenario 2 for UCA16: Primary Flight Display provided information input too late during avionics setup [UCA16] because the flight crew didn't enter the information in time. Such an action would be possible if the company didn't specify the avionics setup procedure [H5] Scenario 3 for UCA16: Primary Flight Display provided information input too late during avionics setup [UCA16] because the received feedback from the Integrated Avionics Unit is unambiguous and it takes too long to process it and display. Such a flawed action would be possible if the software coding wasn't optimized [H5]

Scenario 4 for UCA16: Primary Flight Display provided information input too late during avionics setup [UCA16] because the received feedback from the Integrated Avionics Unit is incorrect and it takes too long to reach the correct information. Such a flawed action would be possible if the software coding wasn't optimized [H5] Scenario 5 for UCA16: Primary Flight Display provided information input too late during avionics setup [UCA16] because the received feedback from the Integrated Avionics Unit takes too long to reach the correct destination. Such a flawed action would be possible if the software coding wasn't optimized [H5]
UCA17: Primary Flight Display provided input information in wrong order during avionics setup [H4, H5]
Scenario 1 for UCA17: Primary Flight Display provided input information in wrong order during avionics setup [UCA17] because of the physical damage of the information transfer channels. As a result, the information was entered in wrong order [H4, H5] Scenario 2 for UCA17: Primary Flight Display provided input information in wrong order during avionics setup [UCA17] because the flight crew didn't enter the information in correct order. Such an action would be possible if the company didn't specify the avionics setup procedure [H4, H5] Scenario 3 for UCA17: Primary Flight Display provided input information in wrong order during avionics setup [UCA17] because the received feedback from the Integrated Avionics Unit was coming in different order. Such a flawed action would be possible if the software integrity was compromised by a malware [H4, H5] Scenario 4 for UCA17: Primary Flight Display provided input information in wrong order during avionics setup [UCA17] because the received correct feedback from the Integrated Avionics Unit was interpreted in the different way. Such a flawed action would be possible if the software integrity was compromised by a malware [H4, H5]
UCA18: Primary Flight Display information input stopped too soon due to interruption during avionics setup [H5]
Scenario 1 for UCA18: Primary Flight Display information input stopped too soon due to interruption during avionics setup [UCA18] because the input wasn't possible any longer due to physical damage of the PFD. As a result, the minimum necessary information input wasn't accomplished [H5] Scenario 2 for UCA18: Primary Flight Display information input stopped too soon due to interruption during avionics setup [UCA18] because the flight crew was unable to set it up

correctly. Such an action would be possible if the company didn’t specify the avionics setup procedure [H5]
UCA19: Integrated Avionics Unit did not provide squawk code when inserted [H4]
Scenario 1 for UCA19: Integrated Avionics Unit did not provide squawk code when inserted [UCA19] because of the physical failure of the transponder. As a result, the squawk code was not transmitted [H4] Scenario 2 for UCA19: Integrated Avionics Unit did not provide squawk code when inserted [UCA19] because the Integrated Avionics Unit did not recognize that the squawk code was entered. Such a flawed action would be possible if the software integrity was compromised by a malware [H4] Scenario 3 for UCA19: Integrated Avionics Unit did not provide squawk code when inserted [UCA19] because the Integrated Avionics Unit erroneously considers that the squawk code is transmitted, when in fact it is not. Such a flawed action would be possible if the software integrity was compromised by a malware [H4]
UCA20: Integrated Avionics Unit provided incorrect squawk code (other than inserted) when entered [H4]
Scenario 1 for UCA20: Integrated Avionics Unit provided incorrect squawk code (other than inserted) when entered [UCA20] because of the physical damage of the information transfer channels. As a result, the information was processed incorrectly [H4] Scenario 2 for UCA20: Integrated Avionics Unit provided incorrect squawk code (other than inserted) when entered [UCA20] because the Integrated Avionics Unit incorrectly considers that the entered squawk code is different. Such a flawed action would be possible if the software integrity was compromised by a malware [H4]
UCA21: Integrated Avionics Unit provided too late the squawk code when entered [H4]
Scenario 1 for UCA21: Integrated Avionics Unit provided too late the squawk code when entered [UCA21] because of the physical damage of the information transfer channels. As a result, the squawk code wasn’t sent in correct time [H4] Scenario 2 for UCA21: Integrated Avionics Unit provided too late the squawk code when entered [UCA21] because the Integrated Avionics Unit wasn’t able to recognize in time that the code was entered. Such a flawed action would be possible if the software integrity was compromised by a malware [H4]

UCA22: Integrated Avionics Unit stopped too soon providing the squawk code during critical phases of flight [H4]
Scenario 1 for UCA22: Integrated Avionics Unit stopped too soon providing the squawk code during critical phases of flight [UCA22] because of the physical damage of the transponder. As a result, the squawk code was unavailable when needed [H4] Scenario 2 for UCA22: Integrated Avionics Unit stopped too soon providing the squawk code during critical phases of flight [UCA22] because the flight crew pressed the wrong button. Such an action would be possible if the company didn't specify any procedure for the correct use of the avionics [H4] Scenario 3 for UCA22: Integrated Avionics Unit stopped too soon providing the squawk code during critical phases of flight [UCA22] because the Integrated Avionics Unit erroneously considers that it is still transmitting the squawk code. Such a flawed action would be possible if the software integrity was compromised by a malware [H4]
UCA23: Integrated Avionics Unit stopped too late providing the old (previous) squawk code during critical phases of flight [H4]
Scenario 1 for UCA23: Integrated Avionics Unit stopped too late providing the old (previous) squawk code during critical phases of flight [UCA23] because of the physical damage of the connectors between Integrated Avionics Unit and transponder. As a result, the squawk code was sent when not needed [H4] Scenario 2 for UCA23: Integrated Avionics Unit stopped too late providing the old (previous) squawk code during critical phases of flight [UCA23] because the Integrated Avionics Unit erroneously considers that the squawk code transmission is stopped. Such a flawed action would be possible if the software integrity was compromised by a malware [H4]

Table 5. Scenarios how the control algorithm may cause the UCAs

4.5 Measures propose to ensure informational security in software maintenance

During the STPA analysis were identified 61 loss scenarios, which are based on 23 UCAs and provoke risk for aircraft information infrastructure maintenance procedure. In order to mitigate negative impact of identified UCAs and ensure software maintenance security, the following correction measures recommended for implementation are described in the table below:

Loss scenarios	Requirements
Scenario 1 for UCA1: Maintenance operator did not provide update download [UCA1] due to absence of the reminder. As a result, the database in the airplane is out of date [H5]	The Organization should create company policy and timely reminders that the software should be updated before expiration date.
Scenario 2 for UCA1: Maintenance operator did not provide update download [UCA1] because he is not authorized for such work. As a result, the organization doesn't have a qualified person to conduct the work and the airplane database will not be updated [H5]	The Organization should delegate the responsibility for the software update to a trained and authorized personnel.
Scenario 3 for UCA1: Maintenance operator did not provide update download [UCA1] because he incorrectly believes that he already did this. This flawed process model will occur if the received feedback shows that the download already exists. Such a feedback may be caused by a malware on computer leading to not providing update when previous version expired [H5]	The Organization should create company policy and carefully monitor the process of operating system software update that address security vulnerabilities as well as antivirus databases.
Scenario 1 for UCA2: Maintenance operator downloaded the update from an unofficial source [UCA2] because the company did not specify which source to be used. As a result, the downloaded update was compromised [H1, H4]	The Organization should create company policy and carefully monitor the process of software update. The Maintenance personnel should not download any software from any source but official.
Scenario 2 for UCA2: Maintenance operator provided update download from unofficial source [UCA2] because he believes the source he	The Organization should create company policy and carefully monitor the process of antivirus software update.

is using is the correct one. This flawed process will occur if the received feedback from the antivirus does not indicate any problem. Such a feedback may be caused by lack of/or expired software protection leading to compromised update [H1, H4]	The Organization should maintain a security awareness training program for maintenance personnel.
Scenario 1 for UCA3: Maintenance operator provided update too late after database expiration [UCA3] because the company did not mention when the update should be downloaded. As a result, the database in the airplane is out of date [H5]	The Organization should create company policy and carefully monitor the deadline of software update. The Maintenance personnel should monitor the software expiration date and the unscheduled update availability.
Scenario 2 for UCA3: Maintenance operator provided update too late after database expiration [UCA3] because he incorrectly believes that downloaded update is still current. This flawed process model will occur if the received feedback shows incorrect update version. Such a feedback may be caused by a malware on the computer that changed the file version [H5]	The Organization should create company policy and carefully monitor the process of malware protection software update. The Maintenance personnel should monitor the software expiration date and the update availability.
Scenario 1 for UCA4: Maintenance operator provided the update before official release [UCA4] because he downloaded it at variable periods. Such an action would be possible if the company did not specify the period boundaries. As a result, the update could contain incomplete information [H1, H5]	The Organization should create company policy and timely reminders that the software should be updated. The Maintenance personnel should monitor the software expiration date and the update availability.

Scenario 2 for UCA4: Maintenance operator provided the update before official release [UCA4] because he incorrectly considers that the update is reliable. This flawed process will occur if the received feedback indicates no problems. Such a feedback may be caused by inability of the computer software to check the update on integrity [H1, H5]	The Organization should create company policy and timely reminders that the software should be updated. The Maintenance personnel should monitor the software expiration date and the update availability and its integrity.
Scenario 1 for UCA5: Maintenance operator update downloading was interrupted [UCA5] because of electrical and/or internet blackouts. As a result, the update integrity was violated [H4, H5]	The Organization should create detailed step-by-step course of actions for the software update process. The Maintenance personnel should not deviate from the update process instructions.
Scenario 2 for UCA5: Maintenance operator interrupted the downloading process before he made sure that the process is finished [UCA5] because there was no indication. As a result, the update integrity was violated [H4, H5]	The Organization should create detailed step-by-step course of actions for the software update process. The Maintenance personnel should not deviate from the update process instructions.
Scenario 3 for UCA5: Maintenance operator update download was interrupted [UCA5] but there is no error indication. This flawed process will occur if the received feedback believes that the process was finished successfully. Such a feedback may be caused by lack of/or expired software protection leading to compromised update [H1, H4]	The Organization should create detailed step-by-step course of actions for the software update process. The Maintenance personnel should not deviate from the company instructions. The Manufacturer should create a self-test for the check of the system integrity.

Scenario 1 for UCA6: The computer did not provide update copy when it is necessary [UCA6] because of the physical failure. As a result, the update copy cannot be performed [H2]	The Organization should create detailed step-by-step course of actions for the software update process. The Organization should provide back-up computer in case of the main computer failure. The Maintenance personnel should not use any other computer available.
Scenario 2 for UCA6: The computer did not provide update copy when it is necessary [UCA6] because the computer software responsible for the process of copy is damaged. As a result, the update copy cannot be performed [H2]	The Organization should create detailed step-by-step course of actions for the software update process. The Organization should provide back-up computer in case of the main computer failure. The Maintenance personnel should not use any computer available.
Scenario 3 for UCA6: The computer did not provide update copy when it is necessary [UCA6] because the computer software, after scanning the removable media, detected that it is unsatisfactory (contains virus and/or does not have enough memory space). As a result, the update copy cannot be performed [H2]	The Organization should create detailed step-by-step course of actions for the software update process. The Maintenance personnel should not disregard this warning and should not use the removable media for its intended purpose.
Scenario 4 for UCA6: The computer did not provide update copy when it is necessary [UCA6] because the computer incorrectly believes that the copy has been already made. This flawed process model will occur if the received feedback indicates that the update is already copied to the removable media. Such a feedback may be	The Organization should create detailed step-by-step course of actions for the software update process. The Organization should provide back-up computer in case of the main computer failure.

caused by a malware on computer/removable media leading to not providing update copy when necessary [H2]	The Maintenance personnel should not use any other computer available.
Scenario 1 for UCA7: Computer provides update copy with malware infection [UCA7] because the antivirus on the computer did not detect any malware software. As a result, the update copy was infected [H4]	The Organization should create detailed step-by-step course of actions for the software update process. The Organization should ensure that the computer antivirus is able to protect the software from viruses. The Organization should not use the computer for software update for any other tasks.
Scenario 2 for UCA7: Computer provides update copy with malware infection [UCA7] because the removable media does not have any software to check for the infections. This flawed process model will occur if the received feedback from the removable media to computer does not indicate any inconsistencies [H2]	The Organization should create detailed step-by-step course of actions for the software update process. The Organization should ensure that the computer antivirus is turned on, up to date and able to protect the software from viruses. The Organization should not use the computer for software update for any other tasks.
Scenario 1 for UCA8: Computer provided update copy too late when the installed software already expired [H5] because the computer was physically inoperable at the necessary moment. As a result, the update was out of date [H5]	The Organization should create detailed step-by-step course of actions for the software update process. The Organization should provide back-up computer in case of the main computer failure. The Maintenance personnel should not use any other computer available.

Scenario 2 for UCA8: Computer provided update copy too late when the installed software already expired [H5] because the process of copy took more time than usually because of the damaged software. As a result, the update was out of date [H5]	The Organization should create detailed step-by-step course of actions for the software update process. The Organization should provide back-up computer in case of the main computer failure. The Maintenance personnel should not use any other computer available.
Scenario 1 for UCA9: Computer update copy stopped too soon due to interruption during downloading to the hardware [UCA9] because during the process of copy the computer was shut down (failure and or electrical blackout). As a result, the process of copy wasn't completed fully [H4, H5]	The Organization should create detailed step-by-step course of actions for the software update process. The Organization should provide back-up computer in case of the main computer failure. The Maintenance personnel should not use any other computer available.
Scenario 2 for UCA9: Computer update copy stopped too soon due to interruption during downloading to the hardware [UCA9] because a person interrupted it forcibly. As a result, the process of copy wasn't completed fully [H4, H5]	The Organization should create detailed step-by-step course of actions for the software update process. The Organization should provide back-up computer in case of the main computer failure.
Scenario 3 for UCA9: Computer update copy stopped too soon due to interruption during downloading to the hardware [UCA9] because the software erroneously considers that the copy was completed. This flawed process model will occur if the received feedback from the hardware contains errors or is missing completely. Such a feedback may be caused by	The Organization should create detailed step-by-step course of actions for the software update process. The Organization should provide back-up computer in case of the main computer failure.

malware on the hardware leading to interruption of the copy [H4, H5]	The Manufacturer should create a self-test for the check of the system integrity.
Scenario 1 for UCA10: Removable media did not provide update transfer when inserted in the slot and requested by the software [UCA10] because of the physical damage. As a result, the update couldn’t be transferred [H5]	The Organization should create detailed step-by-step course of actions for the software update process. The Organization should use special removable media and create procedures of its correct usage.
Scenario 2 for UCA10: Removable media did not provide update transfer when inserted in the slot and requested by the software [UCA10] because it was inserted in the wrong slot. Such an action may be caused by lack of knowledge and missing prescribed procedures by the company [H5]	The Organization should create detailed step-by-step course of actions for the software update process. The Organization should use special removable media and create procedures of its correct usage.
Scenario 3 for UCA10: Removable media did not provide update transfer when inserted in the slot and requested by the software [UCA10] because it cannot be detected by the PFD software [H5]	The Organization should create detailed step-by-step course of actions for the software update process. The Organization should use special removable media and create procedures of its correct usage.
Scenario 1 for UCA11: Removable media provided update transfer with malware during information transfer to the software [UCA11] because the removable media antivirus is missing and/or couldn’t recognize malware. Such an action would be possible if the antivirus is not updated timely or it is missing completely.	The Organization should create detailed step-by-step course of actions for the software update process. The Organization should use special removable media and create procedures of its use. The Manufacturer should create a self-test for the check of the system integrity.

As a result, the removable media transferred a malware [H4, H5]	
Scenario 1 for UCA12: Removable media provided update transfer too late due to its malfunction when the software already expired [UCA12] because the removable media was physically damaged. As a result, the update was transferred after the software already expired [H5]	The Organization should create detailed step-by-step course of actions for the software update process. The Organization should use special removable media and create procedures of its use.
Scenario 1 for UCA13: Removable media update transfer stopped to soon due to SD card extraction during transfer process [UCA13], because maintenance personnel wasn't instructed about the correct process of update. Such an action would be possible if the company didn't train and/or didn't specify the procedures. As a result, the update wasn't transferred completely [H4, H5]	The Organization should create detailed step-by-step course of actions for the software update process. The Organization should use special removable media and create procedures of its use. The Maintenance personnel should not stop the update process until it is not finished.
Scenario 2 for UCA13: Removable media update transfer stopped to soon due to SD card extraction during transfer process [UCA13], because the PFD erroneously believes that the transfer was done completely [H4, H5]	The Organization should create detailed step-by-step course of actions for the software update process. The Organization should use special removable media and create procedures of its use. The Manufacturer should create a self-test for the check of the system integrity.
Scenario 1 for UCA14: Primary Flight Display did not provide information input during avionics preparation phase [UCA14] because of the physical damage. As a result the input of the	The Organization should create detailed step-by-step course of actions for the avionics preparation.

information was not possible partly or completely [H5]	The Organization should not permit the airplane to operate and should seek for maintenance.
Scenario 2 for UCA14: Primary Flight Display did not provide information input during avionics preparation phase [UCA14] because the flight crew did not set anything. Such an action would be possible if the company didn't specify the avionics setup procedure [H5]	The Organization should create detailed step-by-step course of actions for the avionics preparation. The flight crew should not operate the airplane without avionics preparation.
Scenario 3 for UCA14: Primary Flight Display did not provide information input during avionics preparation phase [UCA14] although everything was done correctly by the crew because it doesn't recognize any input information. Such an action would be possible if there is a malware present in the software [H5]	The Organization should create detailed step-by-step course of actions for the avionics preparation. The Organization should not permit the airplane to operate and should seek for maintenance.
Scenario 1 for UCA15: Primary Flight Display provided information input with erroneous data during insertion [UCA15] because of the physical damage of external/internal components. As a result, the PFD entered wrong data [H5]	The Organization should create detailed step-by-step course of actions for the avionics preparation. The Organization should not permit the airplane to operate and should seek for maintenance. The Manufacturer should create a self-test for the check of the system integrity.
Scenario 2 for UCA15: Primary Flight Display provided information input with erroneous data during insertion [UCA15] because the flight crew entered the information incorrectly. Such an action would be possible if the company didn't specify the avionics setup procedure [H5]	The Organization should create detailed step-by-step course of actions for the avionics preparation.

Scenario 3 for UCA15: Primary Flight Display provided information input with erroneous data during insertion [UCA15] because the information was entered incorrectly on purpose by an intruder. Such an action would be possible if the company doesn't have a strong security protection [H5]	The Organization should create detailed step-by-step course of actions for the avionics preparation. The Organization should check who has the physical access to the airplane. The Manufacturer should create a self-test for the check of the system integrity.
Scenario 4 for UCA15: Primary Flight Display provided information input with erroneous data during insertion [UCA15] because it receives wrong feedback from Integrated Avionics Unit. This flawed action would be possible if the feedback received from the sensors compromises the entered information [H5]	The Manufacturer should create a self-test for the check of the system integrity. The Organization should create a procedure and training for the flight crew to detect the display of erroneous data.
Scenario 5 for UCA15: Primary Flight Display provided information input with erroneous data during insertion [UCA15] because the controllers erroneously believes that the entered information is correct. Such an action would be possible if there is a malware present in the software [H5]	The Manufacturer should create a self-test for the check of the system integrity. The Organization should create a procedure and training for the flight crew to detect the display of erroneous data.
Scenario 1 for UCA16: Primary Flight Display provided information input too late during avionics setup [UCA16] because of the physical damage of the information transfer channels. As a result, the information was entered with a delay [H5]	The Manufacturer should create a self-test for the check of the system integrity.
Scenario 2 for UCA16: Primary Flight Display provided information input too late during avionics setup [UCA16] because the flight crew	The Organization should create detailed step-by-step course of actions for the avionics preparation.

didn’t enter the information in time. Such an action would be possible if the company didn’t specify the avionics setup procedure [H5]	
Scenario 3 for UCA16: Primary Flight Display provided information input too late during avionics setup [UCA16] because the received feedback from the Integrated Avionics Unit is unambiguous and it takes too long to process it and display. Such a flawed action would be possible if the software coding wasn’t optimized [H5]	The Organization should create detailed step-by-step course of actions for the avionics preparation. The Manufacturer should create a self-test for the check of the system integrity.
Scenario 4 for UCA16: Primary Flight Display provided information input too late during avionics setup [UCA16] because the received feedback from the Integrated Avionics Unit is incorrect and it takes too long to reach the correct information. Such a flawed action would be possible if the software coding wasn’t optimized [H5]	The Manufacturer should create a self-test for the check of the system integrity.
Scenario 5 for UCA16: Primary Flight Display provided information input too late during avionics setup [UCA16] because the received feedback from the Integrated Avionics Unit takes too long to reach the correct destination. Such a flawed action would be possible if the software coding wasn’t optimized [H5]	The Manufacturer should create a self-test for the check of the system integrity.
Scenario 1 for UCA17: Primary Flight Display provided input information in wrong order during avionics setup [UCA17] because of the physical damage of the information transfer	The Manufacturer should create a self-test for the check of the system integrity.

channels. As a result, the information was entered in wrong order [H4, H5]	
Scenario 2 for UCA17: Primary Flight Display provided input information in wrong order during avionics setup [UCA17] because the flight crew didn't enter the information in correct order. Such an action would be possible if the company didn't specify the avionics setup procedure [H4, H5]	The Organization should create detailed step-by-step course of actions for the avionics preparation. The Manufacturer should create a self-test for the check of the system integrity.
Scenario 3 for UCA17: Primary Flight Display provided input information in wrong order during avionics setup [UCA17] because the received feedback from the Integrated Avionics Unit was coming in different order. Such a flawed action would be possible if the software integrity was compromised by a malware [H4, H5]	The Manufacturer should create a self-test for the check of the system integrity.
Scenario 4 for UCA17: Primary Flight Display provided input information in wrong order during avionics setup [UCA17] because the received correct feedback from the Integrated Avionics Unit was interpreted in the different way. Such a flawed action would be possible if the software integrity was compromised by a malware [H4, H5]	The Manufacturer should create a self-test for the check of the system integrity.
Scenario 1 for UCA18: Primary Flight Display information input stopped too soon due to interruption during avionics setup [UCA18] because the input wasn't possible any longer due to physical damage of the PFD. As a result,	The Organization should create detailed step-by-step course of actions for the avionics preparation. The Organization should not permit the airplane to operate and should seek for maintenance.

the minimum necessary information input wasn't accomplished [H5]	
Scenario 2 for UCA18: Primary Flight Display information input stopped too soon due to interruption during avionics setup [UCA18] because the flight crew was unable to set it up correctly. Such an action would be possible if the company didn't specify the avionics setup procedure [H5]	The Organization should create detailed step-by-step course of actions for the avionics preparation.
Scenario 1 for UCA19: Integrated Avionics Unit did not provide squawk code when inserted [UCA19] because of the physical failure of the transponder. As a result, the squawk code was not transmitted [H4]	The Manufacturer should create a self-test for the check of the system integrity. The Organization should not permit the airplane to operate and should seek for maintenance.
Scenario 2 for UCA19: Integrated Avionics Unit did not provide squawk code when inserted [UCA19] because the Integrated Avionics Unit did not recognize that the squawk code was entered. Such a flawed action would be possible if the software integrity was compromised by a malware [H4]	The Manufacturer should create a self-test for the check of the system integrity. The Organization should not permit the airplane to operate and should seek for maintenance.
Scenario 3 for UCA19: Integrated Avionics Unit did not provide squawk code when inserted [UCA19] because the Integrated Avionics Unit erroneously considers that the squawk code is transmitted, when in fact it is not. Such a flawed action would be possible if the software integrity was compromised by a malware [H4]	The Manufacturer should create a self-test for the check of the system integrity. The Organization should not permit the airplane to operate and should seek for maintenance.
Scenario 1 for UCA20: Integrated Avionics Unit provided incorrect squawk code (other than inserted) when entered [UCA20] because of the	The Manufacturer should create a self-test for the check of the system integrity.

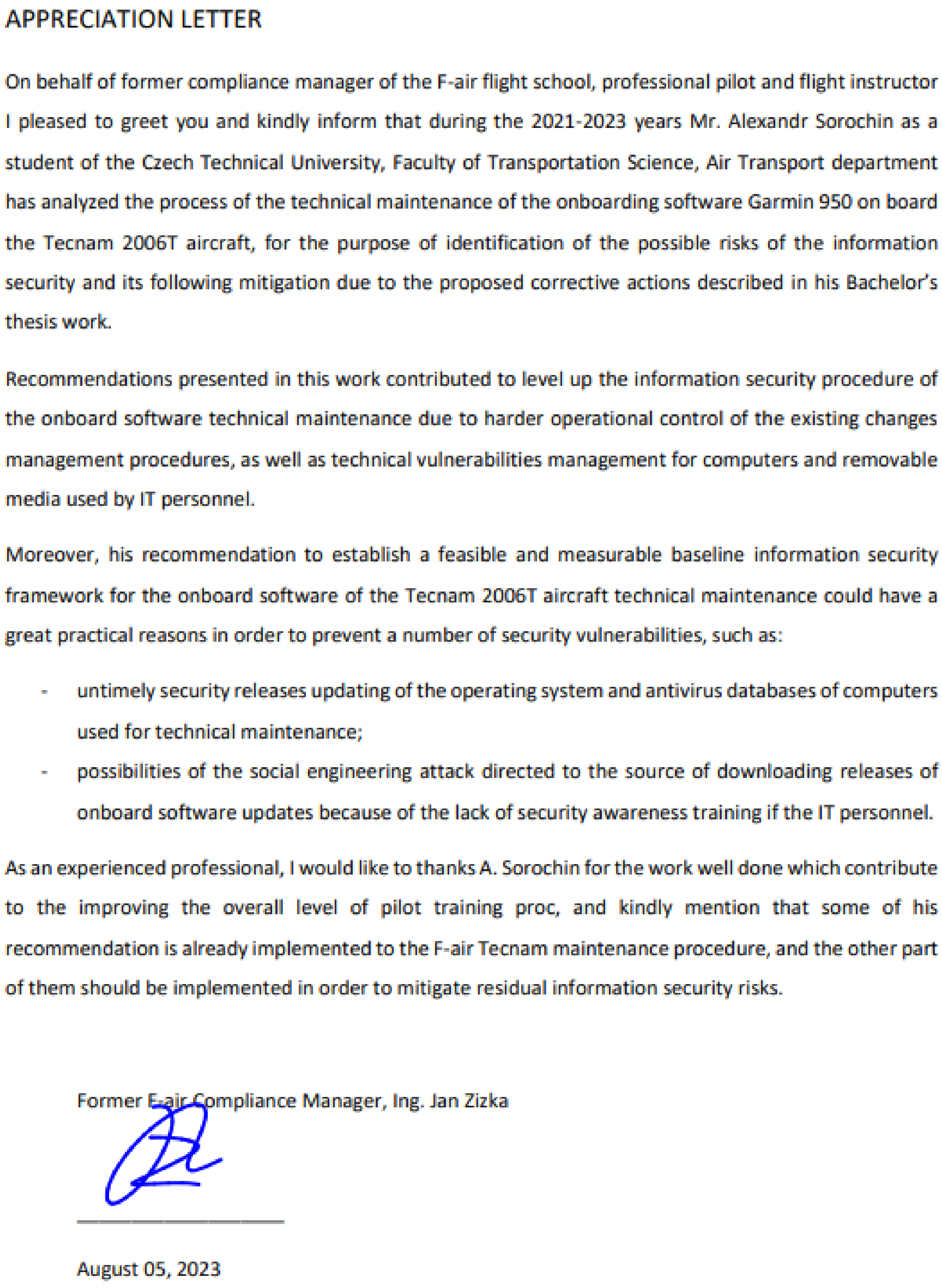
physical damage of the information transfer channels. As a result, the information was processed incorrectly [H4]	The Organization should not permit the airplane to operate and should seek for maintenance.
Scenario 2 for UCA20: Integrated Avionics Unit provided incorrect squawk code (other than inserted) when entered [UCA20] because the Integrated Avionics Unit incorrectly considers that the entered squawk code is different. Such a flawed action would be possible if the software integrity was compromised by a malware [H4]	The Manufacturer should create a self-test for the check of the system integrity. The Organization should not permit the airplane to operate and should seek for maintenance.
Scenario 1 for UCA21: Integrated Avionics Unit provided too late the squawk code when entered [UCA21] because of the physical damage of the information transfer channels. As a result, the squawk code wasn't sent in correct time [H4]	The Manufacturer should create a self-test for the check of the system integrity. The Organization should not permit the airplane to operate and should seek for maintenance.
Scenario 2 for UCA21: Integrated Avionics Unit provided too late the squawk code when entered [UCA21] because the Integrated Avionics Unit wasn't able to recognize in time that the code was entered. Such a flawed action would be possible if the software integrity was compromised by a malware [H4]	The Manufacturer should create a self-test for the check of the system integrity. The Organization should not permit the airplane to operate and should seek for maintenance.
Scenario 1 for UCA22: Integrated Avionics Unit stopped too soon providing the squawk code during critical phases of flight [UCA22] because of the physical damage of the transponder. As a result, the squawk code was unavailable when needed [H4]	The Manufacturer should create a self-test for the check of the system integrity. The Organization should not permit the airplane to operate and should seek for maintenance.

Scenario 2 for UCA22: Integrated Avionics Unit stopped too soon providing the squawk code during critical phases of flight [UCA22] because the flight crew pressed the wrong button. Such an action would be possible if the company didn't specify any procedure for the correct use of the avionics [H4]	The Manufacturer should create a self-test for the check of the system integrity. The Organization should not permit the airplane to operate and should seek for maintenance. The Organization should create detailed step-by-step course of actions for the avionics preparation.
Scenario 3 for UCA22: Integrated Avionics Unit stopped too soon providing the squawk code during critical phases of flight [UCA22] because the Integrated Avionics Unit erroneously considers that it is still transmitting the squawk code. Such a flawed action would be possible if the software integrity was compromised by a malware [H4]	The Manufacturer should create a self-test for the check of the system integrity. The Organization should not permit the airplane to operate and should seek for maintenance.
Scenario 1 for UCA23: Integrated Avionics Unit stopped too late providing the old (previous) squawk code during critical phases of flight [UCA23] because of the physical damage of the connectors between Integrated Avionics Unit and transponder. As a result, the squawk code was sent when not needed [H4]	The Manufacturer should create a self-test for the check of the system integrity. The Organization should not permit the airplane to operate and should seek for maintenance.
Scenario 2 for UCA23: Integrated Avionics Unit stopped too late providing the old (previous) squawk code during critical phases of flight [UCA23] because the Integrated Avionics Unit erroneously considers that the squawk code transmission is stopped. Such a flawed action would be possible if the software integrity was compromised by a malware [H4]	The Manufacturer should create a self-test for the check of the system integrity. The Organization should not permit the airplane to operate and should seek for maintenance.

5. Proposed solution validation

Validation of the proposed measures was implemented by the former Compliance Manager of F-Air, professional pilot and flight instructor. He pointed out that a part of the proposed solutions are already implemented in the company, some of them were found useful and should be implemented, and the rest of the requirements are valid for the manufacturer, which is out of the flight training school area of responsibility.

The full version of his validation expertize is presented below in the Figure 20.



APPRECIATION LETTER

On behalf of former compliance manager of the F-air flight school, professional pilot and flight instructor I pleased to greet you and kindly inform that during the 2021-2023 years Mr. Alexandr Sorochin as a student of the Czech Technical University, Faculty of Transportation Science, Air Transport department has analyzed the process of the technical maintenance of the onboarding software Garmin 950 on board the Tecnam 2006T aircraft, for the purpose of identification of the possible risks of the information security and its following mitigation due to the proposed corrective actions described in his Bachelor's thesis work.

Recommendations presented in this work contributed to level up the information security procedure of the onboard software technical maintenance due to harder operational control of the existing changes management procedures, as well as technical vulnerabilities management for computers and removable media used by IT personnel.

Moreover, his recommendation to establish a feasible and measurable baseline information security framework for the onboard software of the Tecnam 2006T aircraft technical maintenance could have a great practical reasons in order to prevent a number of security vulnerabilities, such as:

- untimely security releases updating of the operating system and antivirus databases of computers used for technical maintenance;
- possibilities of the social engineering attack directed to the source of downloading releases of onboard software updates because of the lack of security awareness training if the IT personnel.

As an experienced professional, I would like to thanks A. Sorochin for the work well done which contribute to the improving the overall level of pilot training proc, and kindly mention that some of his recommendation is already implemented to the F-air Tecnam maintenance procedure, and the other part of them should be implemented in order to mitigate residual information security risks.

Former F-air Compliance Manager, Ing. Jan Zizka

August 05, 2023

Figure 20. Appreciation letter

6. Conclusion

In this paper, we have analyzed security aspects of a software maintenance process by a novel approach – System-Theoretic Process Analysis. The novel contribution of this work is to formalize an approach to analyze the dependencies between software maintenance process and the capabilities of a cyber-attacker to the aircraft information infrastructure.

In accordance with STPA analysis results, were identify the full set of scenarios that lead to system losses during the software maintenance due to UCAs caused by the human factor. The STPA analysis helps to identify the loss scenarios where different mitigation strategies that lead to high level system losses, in case of poor technical vulnerabilities management and outdated antivirus databases software installed on equipment used for aircraft maintenance, as well as insufficient security requirements paperwork for maintenance operational procedures for technicians.

Recommendations presented at this work contributed to level up the information security procedure of the onboard software technical maintenance due to harder operational control of the existing changes management procedures, as well as technical vulnerabilities management for computers and removable media used by IT personnel.

The measures proposed to ensure information security in maintenance process were appreciated by experienced F-air Flying school ingeneers who validate the effectiveness of the above-mentioned corrective measures dedicated to keep the aircraft software security within the acceptable limits.

List of references

[1] V.Stepanov, In case of error, reboot the plane, May 7, 2015, available from: https://tjournal.ru/flood/55010-aircraft-software.

[2] Leslie Meredith, Malware implicated in fatal Spanair plane crash, August 20, 2010, available from: https://www.nbcnews.com/id/wbna38790670.

[3] V. Kosyanchuk, N. Selvesyuk, E. Zybin, R. Khammatov, S. Karpenko, The concept for information security of aircraft equipment, The Cyber security issues magazine, nr.4(28), 2018.

[4] International Civil Aviation Organization Doc 9859, Safety Management Manual, Fourth Edition, 2018.

[5] Garmin G950 Pilot's Guide for the Tecnam P2006T, 190-01146-00 Rev. A.

[6] G900X/G950 Installation and Maintenance Manual – G900X/G950 Installation Overview Page, 190-00719-00, revision D.

[7] Nancy G. Leveson, John P. Thomas, STPA Handbook, 2018.

[8] Changyong YANG, Software Safety Testing Based on STPA, 3rd International Symposium on Aircraft Airworthiness, ISAA 2013.