ICAS 2021 SHANGHAI
International Council of the Aeronautical Sciences
32nd Congress of the International Council of the Aeronautical Sciences
September 6-10, 2021
Pudong Shangri-La, Shanghai, China

Application of Development Assurance Methodologies to Electronic Engine Control System (EECS)

Zhu Qin[1], Wang Chunxiao[2]

[1,2] Airworthiness Engineering Center, AECC Commercial Aircraft Engine Co. Ltd, Shanghai, China

Abstract

The increased level of complexity and integration introduced by modern Electronic Engine Control System (EECS) in turn impacts the challenges of engine certification. The regulatory authorities have highlighted concerns about the possibility of development errors causing or contributing to failure conditions. Meanwhile, the industry has limited experience of adopting development assurance methodologies as the means of compliance of Civil Aviation Regulations.

This paper elaborates upon the above factors and presents approaches and practical experience for the engine Type Certificate (TC) applicant that adopted development assurance methodologies (SAE ARP4754A, SAE ARP4761, RTCA DO-254, RTCA DO-178C etc.) to EECS development and certification. This paper focuses on the following major topics:

1. Providing the method to assess the scope and applicability of development assurance standards;
2. Performing a comparison analysis of the objectives identified in System, Software and Hardware development assurance guidance;
3. Proposing the most efficient compliance approach by utilization common engineering processes, tools and data;
4. Sharing lessons learned with system and item level assurance processes have been observed in the on-going TC program.

Keywords: Development Assurance, Engine Control, Complex System, ARP4754A

1. Introduction and Background of Development Assurance (DA)

Due to the complex and integrated nature of aircraft systems, the regulatory authorities have highlighted concerns about the possibility of development errors causing or contributing to aircraft Failure Conditions. To address these concerns, a methodology to mitigate development errors is required.

1.1 What is DA

Development Assurance are a set of planned and systematic actions used to substantiate, at an adequate level of confidence, that errors in requirements, design and implementation have been identified and corrected such that the system satisfies the applicable certification basis.

1.2 DA requirements defined in regulations and standards

Industry standards (for example, Society of Automotive Engineers (SAE) and Radio Technical Commission for Aeronautics (RTCA) standards) become the acceptable means of compliance in the particular certification program, in the condition that they are recognized by airworthiness authorities, through AC (advisory circular), AMC (acceptable means of compliance), IP (issue paper), CRI (certification review item) or other equivalent methods.

- SAE ARP4754A is one of the acceptable DA methodologies. The overall purpose of ARP4754A DA process is to provide a level of confidence that errors or omissions in requirements or design have been identified and corrected to the degrees that the system as implemented, satisfied applicable certification requirement [1].
- Besides ARP4754A, item level (software and hardware) design assurance standards – RTCA DO-178C “Software Considerations in Airborne Systems and Equipment Certification” [2] and RTCA DO-254 “Design Assurance guidance for Airborne Electronic Hardware” have been recognized by authorities [3].

- These Software and electronic hardware related processes (DO-178C and DO-254) are no longer considered to be adequate to mitigate the risk of EECS errors. DA activities at EECS system level are deemed to limit the likelihood of development errors contained in the system requirements used as the basis for the development of software and airborne electronic hardware items.

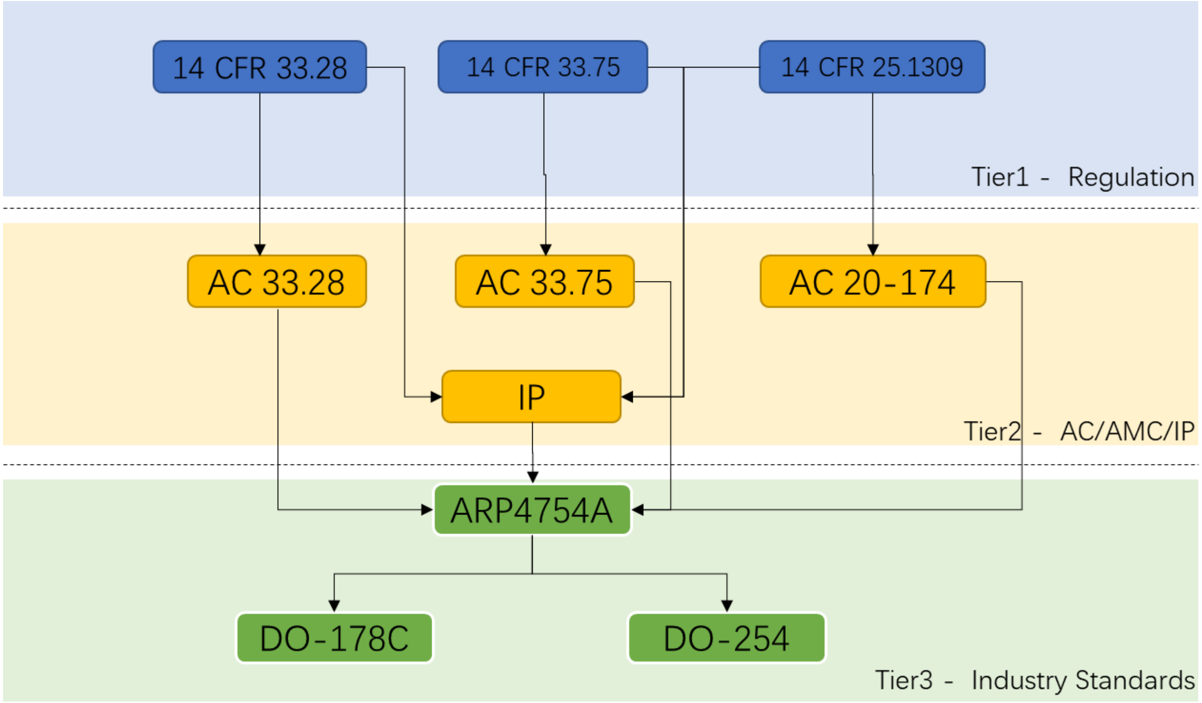

Figure 1 – Relationship between DA standards and regulations

Figure 1 explain how to apply industry DA standards to satisfy top level safety objectives in airworthiness regulations (Code of Federal Regulations (CFR), China Civil Aviation Regulations (CCAR), Certification Specifications (CS), etc.), through AC/AMC/IP. Please note that this paper uses CFR and FAA Advisory Circular as an example, the other certification authorities' regulation, guidance material and standards have the similar relationship and contents.

The purpose for this section is to explain the root source for the needs to establish development assurance process, the hierarchy of regulation and DA standards are presented in Figure1. These standards and regulations have inter-relationship with each other, so that it's necessary to identify their interface, overlaps and specific requirement when apply them in EECS programs, see section 3 for details for the proposed strategy when using all of these standards together.

2. ARP4754A Application Strategy in EECS

2.1 Introduction of EECS

According to the commands from aircraft and the condition of the engine, EECS implement the engine's start, ignition, shutdown, power management, deceleration or acceleration, variable geometry control, thermal management and other functions under all operation conditions. EECS ensures that Engine's safe and stable operation in whole flight envelope, it also produce parameters to aircraft for cockpit warnings.

Civil Aviation Engine Control Systems have been involved from mechanical hydraulic system to Full Authority Digital Engine Control (FADEC). With the development of embedded system, data bus, model-based development and other technologies, EECS will further promote to the direction of intelligentizing, distribution and electrification development.

2.2 Application Scope

The DA process and methodology provided by ARP4754A are mainly aimed at complex system. Section 4.1.1 in ARP4754A states that "A concern arose regarding the efficiency and coverage of techniques used for assessing safety aspects of highly integrated systems that perform complex and interrelated functions, particularly through the use of electronic technology and software-based techniques." [1].

For "non-complex systems", per the suggestions defined in section5.2.3.3 and section5.4 in ARP4754A [1], it's allowed to use traditional engineering technique, which means, those "simple" components can be considered as meeting IDAL A rigor when they are fully assured by a combination of testing and analysis [1].

However, it is always debatable when determine a system "simple" or "complex". ARP4754A and AMC25.1309 define "Complexity" as – "An attributes of functions, systems/items, which makes their operation, failure modes, or failure effects difficult to comprehend without the aid of analytical Complexity methods." [1]. Federal Aviation Administration (FAA)'s technical report TC-17/26 separates "design complexity" and "verification complexity", and provides a more comprehensive and quantitate methods to calculate the level of complexity based on the numbers of modules/components and interfaces [5].

According to the guidelines in above, and consider the key characteristics of engine control system development, a proposal has been initialized to assess and categorize the complexity level within EECS, thus to determine the degree of strict when using DA method in ARP4754A.

The complexity is divided into the following three categories according to the characteristics of the system and sub-system:

1) Complex System/Sub-system

The complex system/sub-system often have the following features:

a) Difficult to conduct comprehensive test or analysis;
b) The components for inter and intra systems are highly coupled;
c) Multiple parameters and complex structure
d) May occur failure propagation and cascading failures

For Complex systems, ARP4754A should be fully applied to establish and implement a set of systematic and structured development assurance process according to its applicable Development Assurance Level $^{\text{Note1}}$.

Note1: The Development Assurance Level is assigned depending on the severity classification of Failure Conditions considering the possible independence between development processes that can limit the consequences of development errors. The more severe the Failure Condition Classification, the greater the level of Development Assurance necessary to mitigate the Failure Condition. The level of rigor of development process for system functions hereafter called Function Development Assurance Level (FDAL) [1]; the level of rigor of development process for item (electronic hardware or software assurance level) called here after Item Development Assurance Level (IDAL) [1].

2) Non-Complex System/Sub-system

The non-complex system/sub-system often have the following features:

a) Includes many physical/functional components and many variable parameters
b) The degree of coupling between inter/intra components is relatively lower compared with the above type;
c) comprehensive testing and/or analysis can be executed

For non-complex systems, the application scope of ARP4754A could be tailored, but should at least satisfy ARP4754A Validation and Verification objectives to ensure that the potential design errors and their effect to aircraft/engine safety can be reduced at the acceptable level. This type of system may be considered FDAL D, regardless of the originally defined FDAL level.

3) Simple System/Sub-system

This type of system has relatively few components, and its expected functionality can be achieved through traditional exhaustive testing and/or analysis without the occurrence of unexpected functionality.

For simple systems, structured development assurance activities may not be carried out in accordance with APR4754A. As recommended in sections 5.2.3.3 and 5.4 of ARP4754A, a combination of adequate testing and analysis can be used to ensure that the intended functionality is achieved and design errors are eliminated as much as possible [1]. However, chapter 5.4 of ARP4754A requires that for such a simple system, the requirements still need to be validated according to its corresponding development assurance level [1]. And the interface between simple system and other types of systems should follow the stricter level of DA rigor, to avoid the design error in the interface have negative effort to the complex and critical components.

The following figure shows an example when adopting ARP4754A DA methods for EECS system in an Engine TC project.

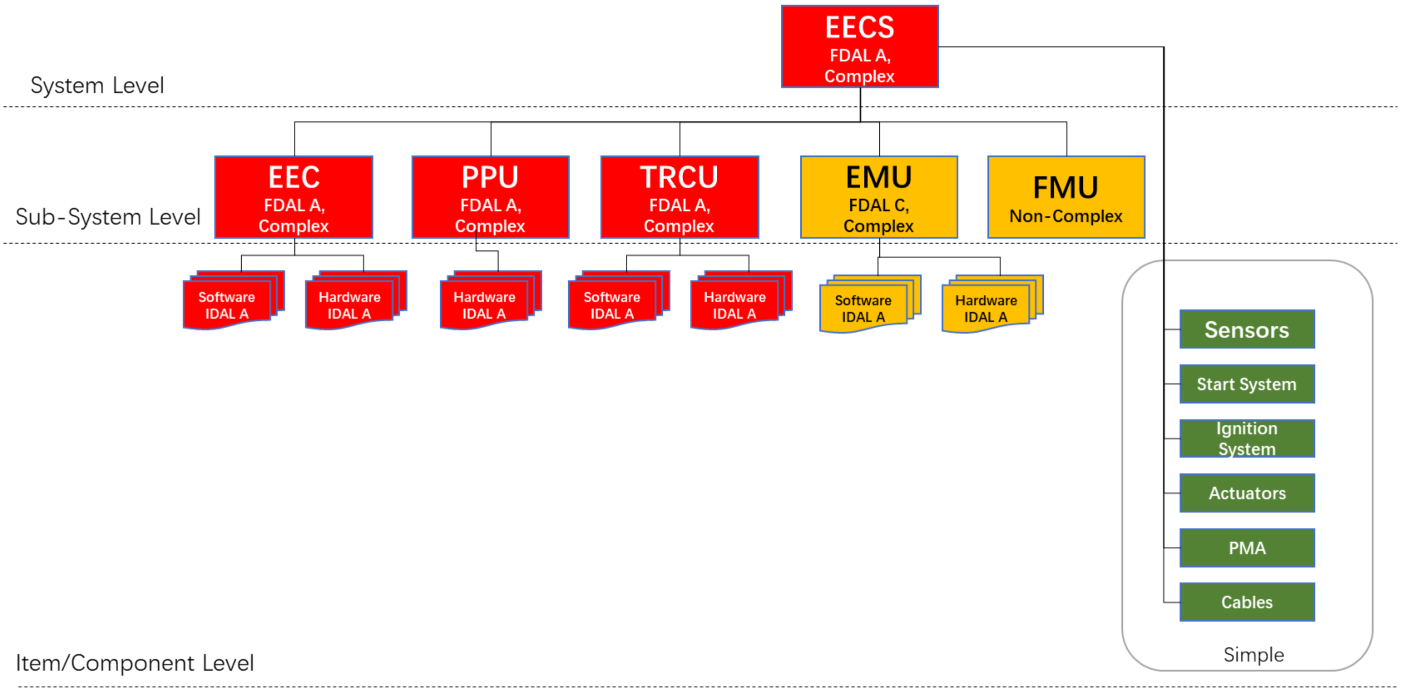

Figure 2 – An example when adopting ARP4754A DA methods for EECS system

- The whole EECS at the top level is considered as FDAL A Complex System, so it should show compliance to ARP4754A FDAL A objectives;
- Electronic Engine Control（EEC)/ Pressure Processing Unit (PPU)/ Engine Monitor Unit (EMU)/ Thrust Reverser Control Unit (TRCU) are sub-system which contains multiple software components and/or hardware components, and perform their intended functions; They are complex sub-systems, and follow different level of DA rigor per FDAL;
- Fuel Metering Unit (FMU) are composed by multiple Values, Accumulators, Strainer etc., the inter relationship are not that complex; comprehensive test and analysis could be conducted in conjunction with other 14 CFR 33 regulations. Therefore, FMU could be categorized as “non-complex” system, and ARP4754A objectives could be tailored.
- Other components in EECS (start system, ignition system, sensors, actuators, PMA, cables etc.) could be treated as “simple” systems/components. They could be fully assured by combination of test and analysis, however, the requirement including interface requirement should be validated with the rigor corresponding to the FDAL of the function.

Please note that Figure 2 just provide a typical example for FDAL/IDAL levels, these levels may be different project by project. FDAL and IDAL assurance level assignment is a top-down process starting with the Failure Condition severity classification from the Functional Hazard Assessment (FHA) and assigning the Top-level FDAL in the Preliminary System Safety Assessment (PSSA).

It’s also noticed that whether the whole Engine should be treated as “complex” or “non-complex” system is a controversial topic; it also relies on the technical maturity and service history of the engine (Original Equipment Manufacturer) OEM. Similar DA strategy and determination method introduced in this paper could be applied in the whole engine level, however this paper will be focused on EECS instead of overall engine.

2.3 Compliance Credit Accumulation

Besides on FDAL and complexity level introduced in above, there are some other factors when using ARP4754A, such as system development stages, maturity of engineering process, new and novel technology and so on. Also, it's an iterative process so that is not only always top down but may have bottom-up influences. In these cases, it's an incremental process for ARP4754A compliance through the whole system development life cycle, so that it's necessary to define an application strategy to gain ARP4754A certification credit cumulatively.

Figure 3 shows an example to define the start point for taking ARP4754A compliance credit and obtain credit cumulatively throughout the Engine Type Certification Milestone.

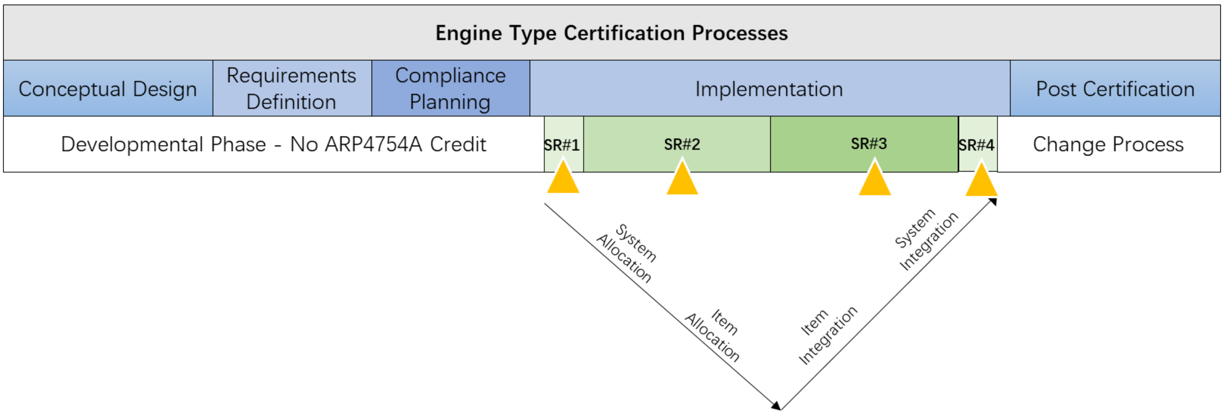

Figure 3 - ARP4754A Compliance Credit through System Development Life Cycle

Certification credit for ARP4754A compliance have been divided into different phases, as below:

- Developmental Phase

Early in the development cycle, engineers must have the ability to freely change the design and optimize the solutions with minimal restrictions and paperwork burdens. In this lifecycle stage, DA can be somewhat informal. However, the configuration control should be sufficient so that any test configuration is documented by the test engineer. There is no authority driven requirement for DA at this stage, so no ARP4754A credit will be taken.

- System Review - Planning Phase （SR#1）

In the ARP4754A planning phase, the development activities move into a pre-certification activity. All ARP4754A Plans (System Development Plan, System Validation Plan, System Verification Plan etc.) have been written and ready for authority review. After the plans been reviewed and accepted by authority, all the system development, V&V, configuration management and process assurance activities should follow the plans and standards, so it's a starting point to collect ARP4754A evidence.

- System Review - Development Phase （SR#2）

System development activities should be implemented by following the approved plans and standards. Usually, applicant is ready to get authority's involvement when a representative portion (typically at least 50 percent) of the system development activities and data (that is, system requirement, design and architecture) is completed and reviewed.

- System Review – Verification Phase （SR#3）

SR#3 review is typically conducted when a representative portion (typically at least 50 percent) of system verification and integration data is completed and reviewed.

- System Review – Final Phase （SR#4）

SR#4 is conducted after the final system development and verification is completed and the very last baseline is ready for formal system certification approval. When authority sign off on the System Accomplishment Summary and System Configuration Index, the whole ARP4754A Compliance activities is completed.

- Post Certification Phase

In the post TC phase, if there is any design change, the modification impact analysis should be conducted and the result of regression test and analysis should be re-visited by certification authority, to ensure the modification are known, fully validated and verified.

3. Harmonization of various DA guidance and regulation

As described in section1.2 above, ARP4754A is called out by advisory circulars or issue papers, so that it becomes one of the acceptable methods for instituting a development assurance process to support compliance to regulations. As the system level DA guidance, ARP4754A is in the middle between Aircraft/Engine level and item (software and hardware) level. At the higher level, “traditional” certification activities should be planned and implemented to satisfy regularity requirements, § 33.28 contains the key elements for control systems development and integration [6]. At the lower level, software and hardware design process should follow RTCA DO-178B/C and RTCA DO-254.

Compared to higher level regulation and lower-level software/hardware standards, ARP4754A are relatively new, so there is less practical experience in both showing compliance and finding compliance in the industry and authorities all around the world. The other reason for the difficulties of ARP4754A application in EECS, is that most of current mainstream Engine Control Systems are evolved from traditional Mechanical hydraulic equipment, there is even less DA application experience compared with Aircraft manufactures and their system and equipment suppliers.

Therefore, it is particularly important to sort out the relationship between the newer guidance (ARP4754A) and former DA guidance/regulations. The compliance activities for the guidance and regulations shown in Figure 4 need to be well planned overall systematically, because all of the activities the same final goal, which is to reduce the design error to the accepted level, and also, they are all acting on the same product – EECS and its sub-systems and components.

Based on the background described above, some questions were identified and summarized (as below) during the application of ARP4754A in EECS project. This paper herein approaches these questions with suggested resolutions.

1) How are ARP4754A guidelines used for regulatory compliance support?
2) How to establish a closer relationship between system development process and system safety analysis process?
3) How do ARP4754A objectives compare to other DA development life cycle objectives?

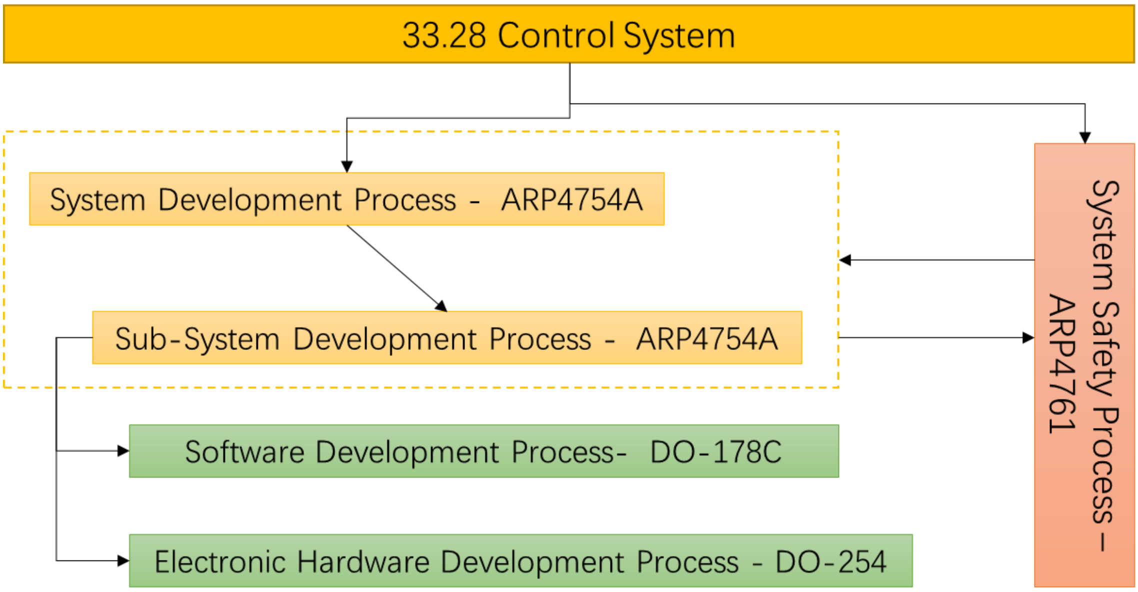
33.28 Control System
System Development Process - ARP4754A
Sub-System Development Process - ARP4754A
Software Development Process- DO-178C
Electronic Hardware Development Process - DO-254
System Safety Process – ARP4761

Figure 4 - Various DA guidance and regulation

3.1 Utilization of ARP4754A and § 33.28 Compliance Activities

Engine Control System is one of the most important portions in the overall 14 CFR 33 engine type certification. Section §33.28 regulates the general design and functioning of the EECS [6]. It does not replace or supersede other regulations governing individual EECS components. Those components, such as alternators, sensors, and actuators, are also regulated under other 14 CFR 33 sections, such as § 33.67 or §33.35 for the fuel system and § 33.91 or §33.53 for individual component tests [6].

§ 33.28 (b) Validation. — (1) Functional Aspects [6]. It specified that the applicant must substantiate by tests, analysis, or a combination thereof, that the engine control system performs the intended functions. The applicant should take special design precautions to minimize any adverse effects from hidden design faults, omissions or discrepancies within the design of the EECS, which are typically the result of incomplete or inaccurate requirements.

AC 33.28-3 accepts ARP4754A as a method for establishing the system development assurance process, and it focused on validation of requirements and verification processes of the design implementation for certification and process assurance [4].

3.1.1 Relationship between ARP4754A and § 33.28

Figure 5 elaborates the relationship between ARP4754A and §33.28 in EECS development and certification activities.

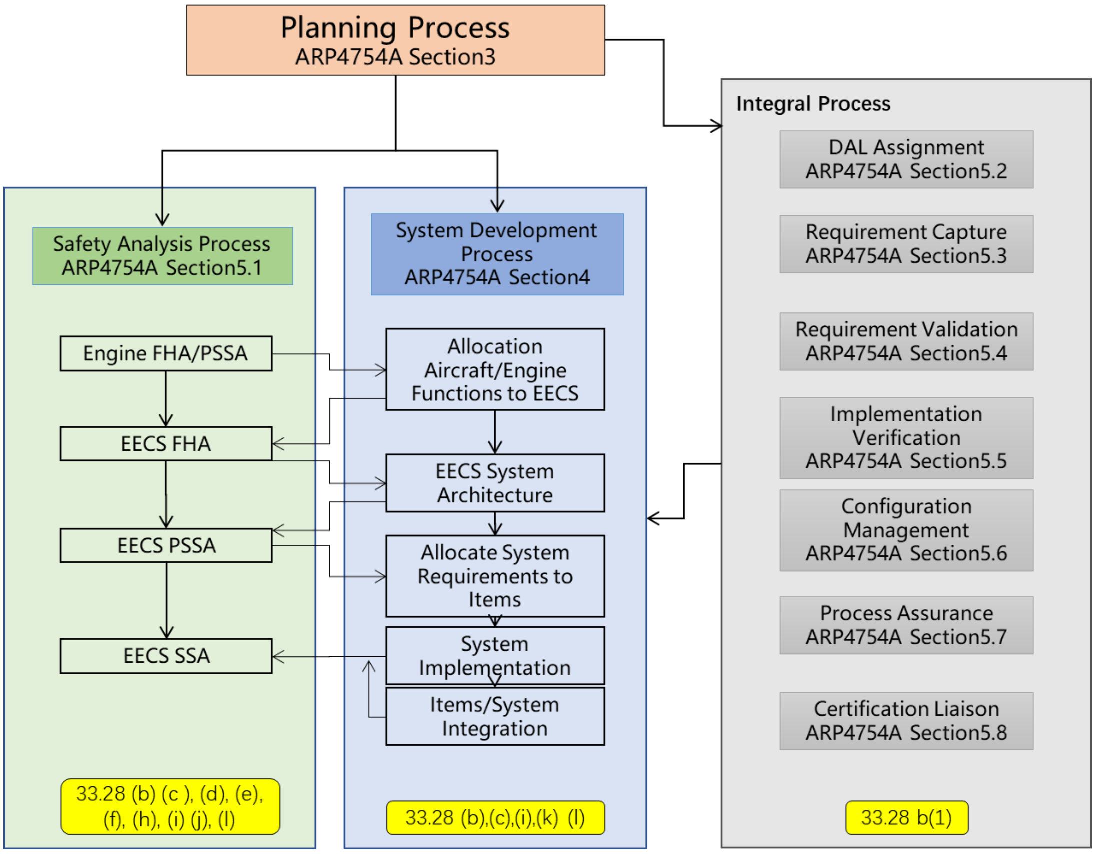
Planning Process
ARP4754A Section3
Integral Process
DAL Assignment ARP4754A Section5.2
Requirement Capture ARP4754A Section5.3
Requirement Validation ARP4754A Section5.4
Implementation Verification ARP4754A Section5.5
Configuration Management ARP4754A Section5.6
Process Assurance ARP4754A Section5.7
Certification Liaison ARP4754A Section5.8
33.28 b(1)
Safety Analysis Process ARP4754A Section5.1
Engine FHA/PSSA
EECS FHA
EECS PSSA
EECS SSA
33.28 (b) (c), (d), (e), (f), (h), (i) (j), (l)
System Development Process ARP4754A Section4
Allocation Aircraft/Engine Functions to EECS
EECS System Architecture
Allocate System Requirements to Items
System Implementation
Items/System Integration
33.28 (b),(c),(i),(k) (l)

Figure 5 - System Development Process, Integral Process and Safety Analysis Process

The overall system development assurance activities have been divided into the following categrifies.

1) Planning Process

The purpose of the development planning process is to define the means of producing EECS which will satisfy the aircraft/engine requirements and provide the level of confidence which is consistent with airworthiness requirements. The key objective of the development planning process is to define the development life cycle, including the inter-relationships between the processes, their sequencing, feedback mechanisms, and transition criteria.

Compared with the initial version of ARP4754, ARP4754A emphasis the importance of planning process. Besides certification plan (CP), ARP4754A defines other plans to cover the whole system development life cycle, as below.

- Certification Plan (CP)
- System Safety Program Plan (SSPP)
- System Development Plan (SDP)
- Requirement Validation Plan (RVP)
- System Verification Plan (SVP)
- System Configuration Management Plan (SCMP)
- System Process Assurance Plan (SPAP)

The CP required by ARP4754A can be combined into one single document with EECS CP (cover § 33.28 and other regulations). A complete set of plans should address the entire design and certification aspects for the whole development life cycle.

2) System Development Process

Section 4 (Aircraft and System Development Process) provides an overview of a generic approach for developing aircraft and aircraft systems from conceptual definition to certification. However, ARP4754A doesn't provide guidance about how to design a system.

§ 33.28 defines functional and performance requirement for EECS, include but not limited with [6]:

- Functional aspects (§ 33.28 b)
- Control transitions (§ 33.28 c)
- Automatic availability and control of engine power for 30-second One Engine Inoperative (OEI) rating. (§ 33.28 k)
- Engine shut down means. Means must be provided for shutting down the engine rapidly. (§ 33.28 l)

Regulatory requirements in § 33.28 should be flown down into EECS system requirement, then further break down into item/component level requirements and be implemented. Therefore, for system development activities should follow general rules in ARP4754A and capture specific regulatory requirements in § 33.28.

3) Safety Process

Safety analysis process and system development process are carried out in parallel.

The safety assessment process is used by a company to show compliance with certification requirements and in meeting its own internal safety standards. The process includes specific assessments conducted and updated during system development and includes how it interacts with the other system development processes. The safety assessment process should be planned and managed so as to provide the necessary assurance that all relevant failure conditions have been identified, and all significant combinations of failures that could cause those failure conditions have been considered.

Figure 5 shows the fundamental relationships between these four specific assessments and the system development processes. In reality, there are many feedback loops within and among these relationships.

The safety program plan should define the scope and the content of the safety activities that the applicable at the aircraft or system level. The Safety program plan should describe the principles of the management, validation and verification of safety requirement, it should also identify the relationship with other appropriate plans (e.g., certification plan, validation and verification plan) .

4) Integral Processes

The integral progresses are performed concurrently with the system development process throughout the life cycle. System Integral Processes don't create system products; however, these processes ensure the correctness and control of, and confidence in system life cycle processes and their outputs. Besides safety process, the other processes are safety assessment, development assurance level assignment, requirements capture, requirement validation, implementation verification, configuration management, process assurance and certification & regulatory authority coordination.

The integral processes are used to support system development process, to establishes levels of confidence that development errors that can cause or contribute to identified Failure Conditions have been minimized with an appropriate level of rigor.

Therefore, integral processes are the key to the success of overall DA processes since they can increase the 'quality' of system development and provide sufficiently disciplined methods to limit the likelihood of development errors that could impact aircraft safety.

In the "traditional" engineering development process (before ARP4754A adopted), some methods and processes in integral processes have been used for decades and become companies' internal process, standards, working instructions etc. For example, almost every company have their internal peer review process when requirements are baselined. Requirement review is one of the accepted methods for requirement validation objectives if it be deployed properly. Some other examples are system level testing and integration, and configuration management and so on. Actually, ARP4754A came from companies' best practice when apply system engineering methodologies in aviation industry.

After ARP4754A been introduced and accepted as an acceptable means of compliance by authorities, sufficient discipline is added by following the integral processes' objectives and requirements; the compliance evidence need to be recorded and all the certification activities and outputs need to be repeatable.

As a summary, the usage of ARP4754A is to drive all relevant technical disciplines, especially in integral process. That's the reason that by using the DA methods in ARP4754A can provide increasing confidence and evidence that a product or process satisfies given requirements. The integral processes defined in ARP4754A provide addition "protection" to system development process.

3.1.2 Utilization of Compliance Data

Many engineering data can be used to demonstrate both §33.28 requirement and ARP4754A objectives, while other data are generated specific for §33.28 regulation or ARP4754A. The common and specific compliance data are summarized as below:

Table 1 – Utilization of ARP4754A and Regulation Compliance Data

Common Data between ARP4754A and §33.28 [1] [6]	Other ARP4754A Specific Life Cycle Data [1]	Other §33.28 Compliance Data [6]
Certification Plan	System Development Plan, System Safety Program Plan, Requirement Plan, System Verification Plan, System Configuration Management Plan, System Process Assurance Plan	Not Applicable (NA)
System Requirement System Design Document Interface Document	Requirement Validation Data (Validation Summary, Validation Matrix)	System Requirement System Design Document Interface Document
Test Plan/Test Report/Analysis Report for: - Control System Dry Rig Test - Control System Wet Rig Test - Control System Component Test - Control System Fault Injection Test	Other type of requirement-based test Test for unintended function Verification Data (Verification Summary, Verification Matrix)	Test Plan/Test Report/Analysis Report for: - Control System Dry Rig Test - Control System Wet Rig Test - Control System Component Test - Control System Fault Injection Test - Component Test

		- Power Test
NA	NA	High Intensity Radiated Fields (HIRF)/ Electromagnetic Compatibility (EMC) /Lightning Consideration
NA	NA	Installation and maintenance Manual
System Safety Assessment (SSA)	FHA/PSSA Common Mode Analysis	Time Limited Dispatch (TLD) Analysis Report Reliability Program Plan Single Upset Event Analysis report
Certification Summary	Configuration Management Record Process Assurance Record Problem Report System Configuration Index	NA

Based upon the relationship stated previously and the common data among §33.28 and ARP4754A, it's necessary to find an efficient method to use a set of shared data to satisfy both objectives. For example, the Hardware in the Loop (HIL) test can be used as one of the key method to satisfy ARP4754A verification objectives and §33.28 objectives for b(1), c(1), i, k etc. [6], if the test plan, test procedure, test cases and test facility are carefully planned and can trace to ARP4754A and regulatory objectives.

3.2 ARP4754A and DO-178C/DO-254

3.2.1 Relationship between ARP4754A and DO-178C/DO-254

ARP4754A, DO-178C and DO-254 are the most widely used DA standards for System, Software and Hardware separately. These standards have all sorts of connection from both historical perspective and technical aspect. The basic process-oriented concept and different domains for integral processes started from DO-178 and DO-254 follow the structure in general. ARP4754A released later on and it's used as the bridge between aircraft/engine level regulation and items (software & hardware) DA standards.

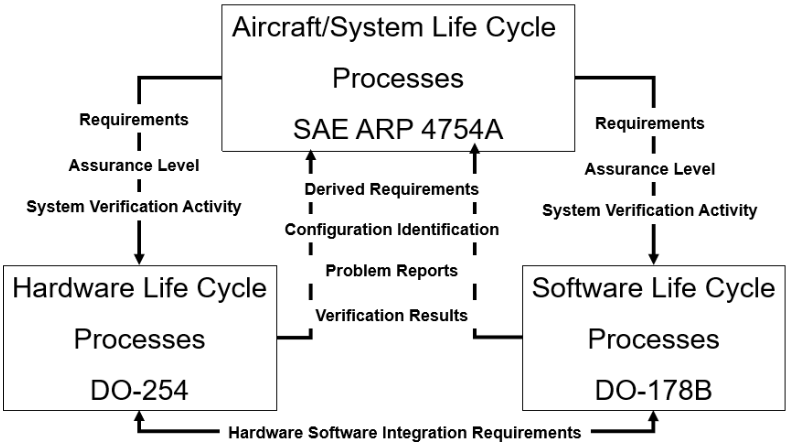

Figure 6 - Relationship between DA standards

Figure 6 is an overview of the relationship between system life cycle processes and the software life cycle processes. More information can be found in RTCA DO-178C section2.2 [2], so no more explanation here.

Here are only list some embedded System Safety considerations in real application:

DAL assignment – The safety process activities (Functional Hazardous Assessment and general safety process) are applicable to aircraft/engine/system development process. System safety process assigns IDAL, which determine the rigor to demonstrate compliance with DO-178C and/or DO-254 [2][3].

- Safety Requirement Allocation - System requirements allocated to hardware and software, including safety requirement, for example, monitor, build-in-test, redundancy requirements, etc.
- Safety Tag - Requirements that are defined to prevent failure conditions or to provide safety related functions should be uniquely identified and traceable through the levels of development. This will ensure visibility of the safety requirements at the software and electronic hardware design level.
- Derived Safety Requirement – derived requirement generated in the software and/or hardware processes shall be feedback to system safety assessment to determine if there is any impact on the system safety assessment and system requirement.
- Change impact analysis – The defects from software/hardware development and testing processes are addressed in Problem Reports (PR). PR need to be categorized and assessed from safety perspective, to identify if safety margin or previous system analysis report are impacted or not.

3.2.2 Similarities and differences

As the development assurance processes specific for aircraft/system, software and hardware domain, ARP4754A/DO-178C/DO-254 [1][2][3]have the following similarities:

- Configuration Management
- Process/Qualify Assurance
- Certification Liaison

The other processes have both differences and commonalities, summarized as Table 2 below:

Table 2 – Comparison between DA guidance

ARP4754A [1]	DO-178C [2]	DO-254 [3]
Plan (7 plans)	Plan (5 plans and 3 standards)	Plan (5 plans and 4 standards)
Safety Assessment	-	-
Assurance Level Assignment	-	-
Requirement Capture	-	-
Development - Function - System Architecture - Allocation	Requirement	Requirements
Implementation	Design Coding Integration	Conceptual Detailed Implementation
-	-	Production Transition
Requirement Validation	-	Validation

For the processes domain have high degree of similarities (configuration management, process assurance and certification liaison), they can share a common set of company procedures, processes and working instructions, for example use one process assurance working instruction, problem report management process, change impact assessment checklist throughout system, hardware and software lifecycle.

For the processes where there are differences, it’s important to understand the real differences, similarities and dependency, so that we can make sure the interface between system/software/hardware lifecycle are established and their consistency is ensured. For example, although DO-178 doesn’t include “validation process”, the requirement review, traceability analysis can be treated as the similar validation objectives in ARP4754A/DO-254 lifecycle. So, in general, the basic validation objective are consistent across the processes, the caveat is that the ARP4754A objective to validate and justify assumptions which has no equivalent objective in either of software or hardware Life Cycle Process (LCP).

4. Challenges of APR4754A Application in EECS

In addition to the software and hardware assurance processes, ARP4754A have recently received increasing attention as the system level development assurance. For the experienced Aircraft/Engine OEM, the introducing of ARP4754A may need an optimization on the existing engineering development processes and best practices. However, adopting ARP4754A DA method is a revolution to the new comers, since they are lack of structured engineering processes and sufficient engineering judgement. Hence, changes and risks are coming in both macro (e.g. how to build the overall process structure to cover ARP4754A Life Cycle) and micro (e.g. how to define good review checklist) aspects.

This section will discuss some major challenges and the proposed in the current state-of-the-practice in EECS development processes.

4.1 Interface Management

The internal and external interfaces in the engine control system are various and the interface relations are complex.

EECS have various type of internal and external interfaces, list as below:

- Internal Interfaces
  - Internal Electrical Interface
  - Internal Communication Interface
  - Mechanical hydraulic Interface
  - Software/Hardware Interface
  - … …
- External Interfaces
  - EECS to Aircraft
  - EECS to Engine Structure
  - EECS to Driving System
  - EECS to Oil System
  - EECS to Thrust Reverser
  - … …

The whole EECS will be treated as the complex system and fully apply ARP4754A, while the sub-systems/components' DA strategy could be adjusted by the rules in section2.2 [1].

Outside EECS, the whole engine structure, compressor, combustor or other components will not be treated as "complex system", so that they are not going to follow ARP4754A FDAL A's objectives.

In both cases in above, there are possible scenarios that the "complex" system (FDAL A rigor) interfaced or communicated with "simple" system/component (lower DA rigor). The following concerns were arisen:

- How to make sure the design error in "simple" system/components will not cause adverse effect on complex system/sub-system?
- How to conduct sufficient Change Impact Analysis (CIA) when interface update?
- How to evaluate if the safety objective/safety requirement/safety margin be affected by interface update?

ARP4754A provides hint in Objective 4.1's comment: Requirement Validation should "Includes coordination of interfaces between systems and between items." [1]. However, the main document doesn't provide detailed guidance. In the real project, authority has highlighted the importance of interface management should be considered within the scope of ARP4754A application.

Take all factors into consideration, here are some suggestions of interface management in real project:

- Utilize one set of management process/procedure to manage and control both requirements and interfaces
- Develop and manage an "EECS data dictionary" and interface requirements to accommodate correct demarcation of functional boundaries and to avoid inadvertent use of data or retention of orphaned data.

- The use of "master" documents to capture the development process to be applied across all systems and their interfaces is recommended.
- When conducting change impact analysis, use checklists and ensure that the changes are thoroughly analyzed by the appropriate internal and external stakeholders; Detail the process and results of the change impact analysis.

4.2 Validation and Verification Methods

V&V (Validation and Verification) processes are very important for the whole development assurance life cycle. Adopting proper validation and verification methods in the development processes is very critical to achieve the goal of reducing design errors.

ARP4754A provides multiple suggested V&V methods, as the table below. However, it doesn't provide guidance for "how" and "when" to use them.

Table 3 – Validation and Verification Methods

Validation Methods	Verification Methods
Engineering Review	Test
Traceability	Modeling
Analysis	Inspection or Review
Modeling	Analysis
Test	Similarity/Service Experience
Similarity	

In the real practice in EECS projects, here list some common challenges and solutions.

4.2.1 V&V for Assumptions and Derived Requirement

Compared with "normal" requirements (can trace up to higher level requirements), it's noticed that V&V Assumptions and Derived Requirements are aeras that still very easy to be ignored during application. Actually, design errors might be hidden during the process when we generate assumptions or derived requirements, unless the V&V activities are well defined and executed. For example, if the unvalidated adjustable parameter in EECS system requirements be flown down and implemented in software, might cause adverse effect to controlled objects (compressor, combustion etc.)

Therefore, V&V for assumptions and derived requirements should be addressed in V&V Plans, in the same manner as other requirement.

- The rationale should be explicitly stated and justified by supporting data (architecture document, ECM, trade study, etc.) for each assumption and derived requirement
- Both Assumptions and Derived Requirement should be identified (e.g. using DOORS module's attributes)
- Derived Requirements should to be evaluated against the system requirement and safety assessments processes. This step is used to identify the missing system requirement and evaluate new failure mode/effect.

4.2.2 Use common methods to satisfy both objectives

It's notified that the validation and verification methods are very similar (refer to Table 3). Also, APR4754A allow to use one method to satisfy both Validation and Verification Objectives (ARP4754A Section5.5.6.1 "Some aspects of the verification process may also support validation of specific requirements and should be coordinated with the validation plan." [1]). During EECS application, these "dual" purposes activities are very common.

For example: Dry-rig/wet-rig testing can be used to proof the correct implementation (verification purpose) and can also find the issues in the requirements themselves (correctness and completeness, validation purpose). In the earlier stage (Preliminary Design Review (PDR) or before PDR), these kinds of tests are more "validation" than "verification", while in the later phase (Critical Design Review (CDR) or after CDR) the focused purposes are shifted.

If the "dual" purposes' methods are proposed, they should be address in both V&V plans and well-coordinated in V&V procedures/cases (e.g. specify the dual-purpose in the header of test cases and test procedures) and V&V results.

4.2.3 Model Simulation’s certification credit

Model Based Development has been broadly used in EECS design. Model simulators provide the ability to execute the model directly without using the target platform. In addition to allowing early exploration of the modeled functions before it is integrated into the target platform, simulators can also provide satisfaction of some V&V objectives. In ARP4754A [1]:

- “Models of systems/items may be used to validate the requirements.” (validation)
- “Modeling may be used for system parameter evaluation, to provide early system information, or other purposes.” (verification).

However, it’s is difficult to justify when ARP4754A certification credit can (or can’t) be taken for modeling activities. Also, modeling activities is also accompanied with simulation and modeling tools. Other subsequent questions came up: can “simulation” take place of “test”, does the modeling and simulation tools need to be qualified or not.

ARP4754A 5.4.6 states that “Care should be exercised to ensure any simulation is sufficiently representative of the actual system, its interfaces, and the installation environment.” [1]. In the other words, the analysis is required to show that simulation approach is equivalent to the non-simulation approach. Comparison analysis (select a set of representative cases, execute them in simulator and real environment, then compare the result) may needed, however the negotiation with certification authority is required, case by case.

In EECS development, here are some scenarios can or can’t use model simulators to take formal credit.

Table 4 – Simulation Credit

Simulation can find	Simulation can’t do
Incorrect or incomplete algorithm	**System Integration on real hardware (e.g. system response times and input/output hardware)**
Incorrect sequencing of events and operations	**Partitional integrity (validate and verify the partitioning mechanism works properly)**
Incorrect logic decisions	
Incorrect or incomplete input conditions	
Incorrect state transitions	
Accuracy and consistency issues	
Algorithm aspect (especially for discontinuities)	
Consistency (relationship between components)	

5. Summary

The complexity in EECS development increases possibilities for development errors (i.e. mistakes in requirements, design, or implementation). Therefore, robust EECS Development Assurance (DA) activities become necessary to achieve safety objectives of regulations. SAE ARP4754A is one of the acceptable DA technologies.

This paper presents current practice and application of SAE ARP4754A in Engine Type Certification Program. The document brings considerations and approaches of some key characteristics for ARP4754A Compliance in EECS. It illustrates the method to determine ARP4754A application scope, it proposes a strategy to accumulate ARP4754A compliance evidence. This paper also addresses the relationship and common objectives between ARP4754A and §33.28/DO-178C/DO-254, and provides a valid proposal to utilize these regulations and standards together, so that certification efforts could be largely reduced. Finally, it summarized the challenges when apply ARP4754A in EECS.

6. Contact Author Email Address

Mailto: qinzhu_amy@163.com; wangchunxiao1001@163.com

7. Copyright Statement

References

[1] SAE ARP4754A. Guidelines for Development of Civil Aircraft and Systems. Warrendale, PA. SAE International, December2010

[2] RTCA DO-178C.Software Consideration in Airborne Systems and Equipment Certification. Washington D.C.RTCA Inc.December2011

[3] RTCA DO-254. Design Assurance Guidance for Airborne Electronic Hardware. Washington D.C.RTCA Inc.April2000

[4] FAA AC 33.28-3. Guidance Material For 14 CFR § 33.28, Engine Control Systems. Engine and Propeller Directorate,

[5] DOT/FAA/TC-17/26. Definition and Measurement of Complexity in the Context of Safety Assurance. William J. Hughes Technical Center Aviation Research Division Atlantic City International Airport. FAA, September2017

[6] 14 CFR 33. AIRWORTHINESS STANDARDS: AIRCRAFT ENGINES. https://www.ecfr.gov/cgi-bin/text-idx?SID=b7d1bd390dcdef076cbde849b3cf06e9&mc=true&tpl=/ecfrbrowse/Title14/14cfr33_main_02.tpl.June.2021