



Software Design Acceptance

Systems Certification and Integrity
Directorate of Aviation Engineering
Directorate General Technical Airworthiness



Overview

- Design Acceptance
- Software Considerations in Design Acceptance
- Software and Recognition of Prior Acceptance
- The Four Pillars of Design Acceptance:
 - Competency
 - Specification
 - Evidence
 - Certification





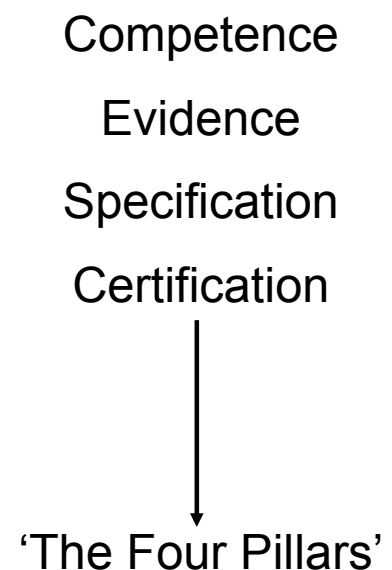
What is Design Acceptance?

- A determination of technical acceptability by the Commonwealth.
- The cornerstone of ADF technical airworthiness and fundamental reason for the DAR.
- Provides confidence that the product is safe and fit for purpose.
 - By critical examination of design evidence.
- A process, not an act!



What is Design Acceptance?

- A determination that:
 - a competent organisation
 - produced evidence to demonstrate that an aircraft design complies with
 - an approved specification and
 - is willing to stand behind the design.
- Design Acceptance is a Commonwealth function that requires Commonwealth involvement in all four pillars.
 - Cannot rely solely on competence.
 - Cannot rely solely on certification.





What is Software Design Acceptance?

- Software Design Acceptance is short hand for software considerations in Design Acceptance.
- Software Design Acceptance is the same as Design Acceptance for other technologies.
 - i.e. the same four pillars apply.
- But, due to some fundamental properties of software, there are some unique methods that must be applied to demonstrate satisfaction of the four pillars.



The Elements of Software Design Acceptance

- Competency
 - AEO
 - with sufficient OPPD to develop aviation software.
- Specification
 - Functional Behaviours
 - The Four Software Airworthiness Requirements
- Evidence
 - Evidence of Functional Behaviours
 - Evidence of Satisfaction of Airworthiness Requirements
 - Assessment of Evidence (Software Compliance Finding)
- Certification
 - As per other designs.





RPA: A complicating factor?

- Recognition of Prior Acceptance (RPA) is a TAMM regulation that allows the ADF to rely on a decision made by another competent airworthiness authority to the extent that decision is applicable to the ADF.
- Simultaneously the best and worst technical airworthiness regulation.
- There are two alternatives to completing the Design Acceptance Process:
 - satisfy the four pillars, or
 - Informed RPA.
- The most common approach involves a combination of the two.

7

Directorate General Technical Airworthiness (DGTA-ADF)



Recognition of Prior Acceptance

- To rely on informed RPA, must demonstrate the following (generally):
 - an aircraft design has been certified by a competent airworthiness authority,
 - that aircraft design has some degree of configuration consistency with the design to be acquired by the ADF,
 - the role and operating environment in which the ADF will use the design has some degree of consistency with the role and operating environment which the other airworthiness authority used as the basis for certification, and
 - the other airworthiness authority did not retain any risks that would be intolerable to the ADF.

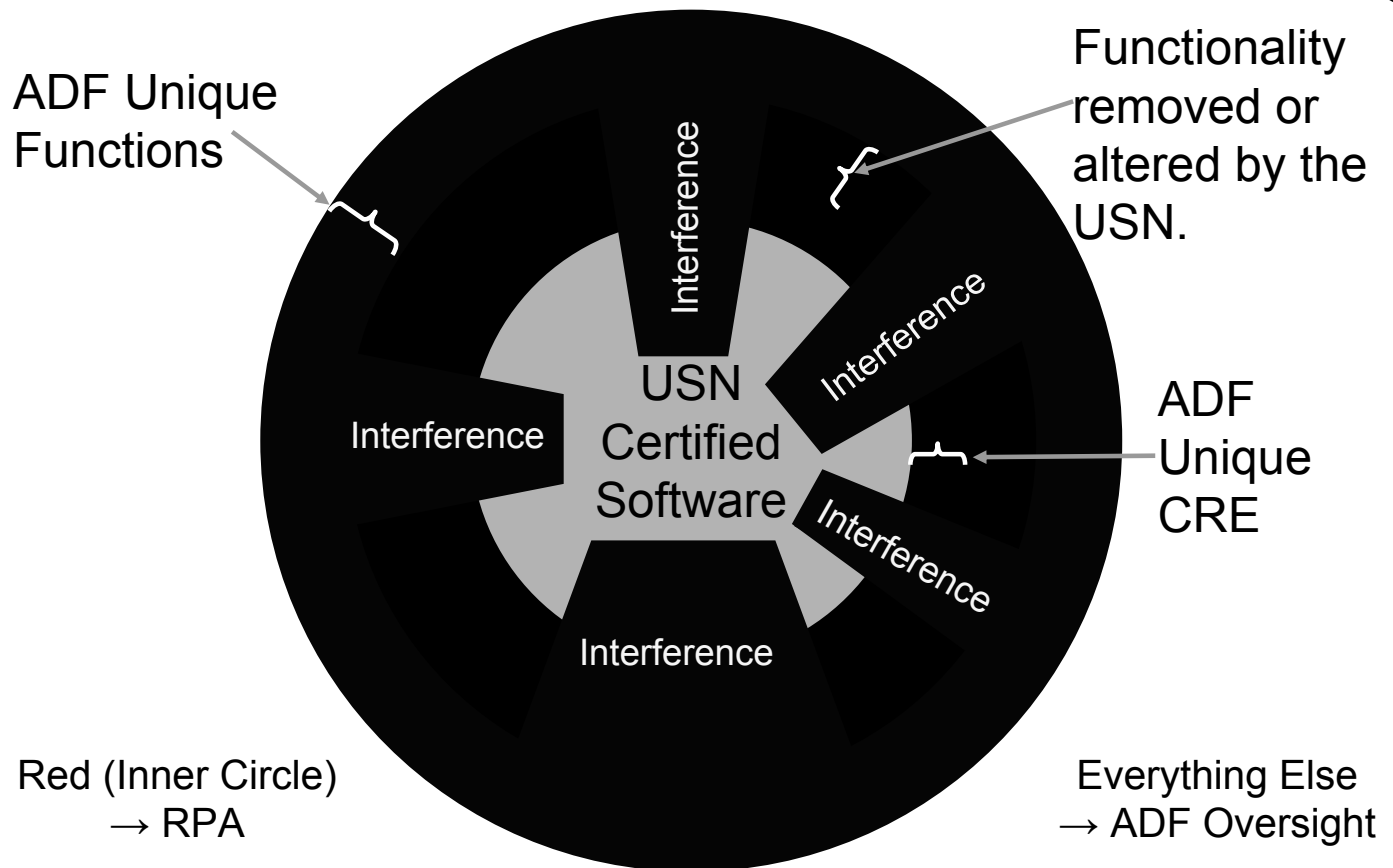
8

Directorate General Technical Airworthiness (DGTA-ADF)





The Trouble with RPA and Software – USN Example



9

Directorate General Technical Airworthiness (DGTA-ADF)



Common RPA Pitfalls

- Recurring Issue: RPA is planned, but then falls through.
 - Contract does not enable four pillars (no access to data, no AEO requirements, no airworthiness requirements).
 - No basis for Design Acceptance.
- Common RPA Issues:
 - Although the item is in service, it was never assessed by a competent airworthiness authority, only by a contractual authority.
 - Slight CRE differences can have big effects:
 - e.g. one pilot operation vs. two pilot operation
 - e.g. different usage spectrum
 - ADF unique functionality.
 - An airworthiness authority may have approved a design as suitable for flight test. The ADF can't rely on this to approve the design for service release.
 - Loss of control of schedule: can't release ADF product until other airworthiness authority has provided certification.

10

Directorate General Technical Airworthiness (DGTA-ADF)





A Common Design Acceptance Strategy

- Combination of RPA and ADF oversight.
- RPA to the maximum extent possible:
 - where the configuration is consistent
 - where the role and operating environment is consistent
 - where retained risks are tolerable
- ADF oversight (four pillars) for the remainder
 - ADF unique configuration
 - ADF unique functionality
 - ADF unique role and operating environment
 - treatment of retained risks



Software and RPA

- Identify evidence of certification
 - Flight Clearance, Airworthiness Release, TSOA, TC, etc.
 - Must be issued by a competent airworthiness authority.
- Demonstrate configuration consistency
 - Compare Software Configuration Index or Version Description Document.
 - May be differences at file level, may be differences in configuration data, may be differences at run time.
- Demonstrate role and operating environment consistency
 - Look for differences in how the aircraft is flown.
 - Generally the same process as for other technologies.
- Identify and incorporate procedures and workarounds
 - Look at flight manual, problem reports.
- Identify retained risks
 - Look at hazard log, problem reports, airworthiness instruments.





Software and Competence

- Aviation software that is part of a Major change to type design must be developed by an AEO.
 - Or an exemption should be sought (note that such an exemption would be an exemption against AEO compliance, not an exemption against the requirement to establish the competence of the organisation).
- Aviation software that is part of a Minor change to type design should be developed by an AEO.
 - Noting Regulation 2.5.6.c.
- AEO requirements alone are not sufficient to demonstrate that an organisation is competent to develop aviation software.
- Need to look deeper at the organisation to determine whether plans and procedures define a development process that is likely to produce airworthy software.



Software and Specification

- Need to specify software functional requirements.
 - i.e. what does 'fit for purpose' mean?
- Also need to specify airworthiness requirements.
 - i.e. what does safe mean?
- There are four airworthiness requirements that must be specified for any change to aviation software:
 - System Safety
 - Software Safety
 - Software Assurance
 - Software Development





- Civil Aircraft
 - System Safety and Software Safety: FAR 2x.1309 and SAE ARP 4754/4761
 - Software Assurance: DO-178B
 - Software Development: IEEE/EIA 12207
- Military Aircraft
 - System Safety: MIL-STD-882C
 - Software Safety: MIL-STD-882C and IEEE STD 1228
 - Software Assurance: Software Assurance Matrix
 - Software Development: MIL-STD-498





Software and Evidence

- The evidence pillar requires the design agency to produce evidence of requirements satisfaction.
 - It is not sufficient just to say that a requirement has been satisfied, it must be proved.
- The Design Acceptance process requires that the Commonwealth review evidence of requirements satisfaction.
 - Not a design review, a check to determine whether the evidence produced supports the claims of requirements satisfaction.
- To do this, the Commonwealth conducts compliance findings.



Compliance Findings

- A compliance finding is:
 - an engineering decision
 - based on relevant evidence
 - that an aircraft design
 - satisfies one or more airworthiness requirements.
- Compliance findings can only be conducted by Commonwealth personnel, or someone acting directly on behalf of the Commonwealth.
 - Can't contract out compliance findings.
 - But can rely on external expertise.





The Software Compliance Finding

- The software compliance finding is similar to other compliance findings.
 - It is an engineering decision based on relevant evidence that an aviation software item satisfies an airworthiness requirement.
- How many software compliance findings are required?
 - Usually just one: against software assurance.
 - Software in System Safety is covered by the System Safety Compliance finding.
 - Software Safety is usually (but not always) embedded within System Safety.
 - Consideration of software assurance objectives is generally sufficient consideration of compliance with a software development standard.



Software Compliance Finding Challenges

- The software compliance finding is heavily dependent on access to data and personnel.
 - Need to have a good plan prior to contract signature to ensure the contract supports the plan.
- It can be an onerous activity.
 - A lot of compliance findings can be done at the end of a program.
 - The software compliance finding is a continuous process that starts the day after contract signature.
 - Trying to catch up at the end is a dangerous approach:
 - Might not get it all done.
 - Shortfalls discovered late in the process can be very expensive to fix.





Conducting The Software Compliance Finding

- May need to cover:
 - Informed RPA
 - Four Pillars for ADF Unique Software
 - Non-Interference
- The mechanics:
 - Establish case for RPA.
 - Review sample design and verification evidence against assurance objectives for ADF unique software.
 - Review sample evidence of regression analysis and re-verification activities to confirm non-interference.



Reviewing Evidence Against Assurance Objectives

- Goal: verify complete and correct satisfaction of each objective.
- Four Step Process:
 - Verify that the software development organisation has claimed complete and correct satisfaction of the objective.
 - Verify that the analysis method used by the software development organisation to determine that the objective has been completely and correctly satisfied is appropriate.
 - Verify that the objective has been correctly satisfied for a sample set of evidence.
 - Verify that the process employed to satisfy the objective is sufficiently robust to assure that similar findings could be made across the entire set of evidence.





Example – Reviewing Traceability

- Objective: Source code is traceable to low-level requirements.
- Four Steps:
 - Verify that the software development agency has determined that all low-level requirements trace to one or more source code elements and that all source code elements trace to one or more low-level requirements.
 - Verify that the analysis method was appropriate and that relevant data was relied upon.
 - Select a number of low-level requirements and attempt to trace to the source code that implements them using only documented data. Attempt to do the same from source code to low-level requirements.
 - Review relevant processes to ensure that traceability data would be consistently established, recorded and reviewed.



Software Compliance Finding – How much is enough?

- A compliance finding is not a design review.
 - You should not review every piece of evidence.
 - But you do have to review evidence against every objective.
 - You should review enough evidence to obtain confidence that your finding (e.g. compliant or non-compliant) is correct.
 - But remember, you may have to convince somebody else (the DAR, DGTA) that your findings are correct.
- ‘How much is enough?’ depends on a number of factors:
 - the criticality of the software
 - the developmental history of the software
 - the competence of the software development organisation





Software Compliance Finding Competencies

- Software compliance findings can be difficult tasks.
 - Particularly at higher assurance levels.
 - The core training and experience profile for FLTLT ELECTRs (or equivalent) generally does not provide the competencies required to complete the more challenging software compliance findings.
 - That is, not all FLTLT ELECTRs (or equivalent) have the required competencies, but some do.
 - Compare with JoS which all FLTLT ELECTRs should be able to conduct.
 - The DAR or PEM should be aware of this and, where possible, seek to have the competencies obtained.
 - SC11 maintains expertise in software and can assist or conduct the more complex compliance findings.
- Specific guidance on required competencies is available in the paper 'Compliance Finding Agency Competencies' available through the DGTA website.



Software Compliance Finding Competencies

- In addition to general competencies.
 - e.g. ability to determine situations beyond their competence
- 'Low' Level of Competence
 - Demonstrated understanding of .054 S2 C7 and C17.
 - Completion of this course.
 - Completion of the software testing course.
- 'Medium' Level of Competence
 - Completion of the DO-178B, MIL-STD-498 and IEEE/EIA 12207 courses as appropriate.
 - Previous experience in the development or acquisition of airborne software systems.
- 'High' Level of Competence
 - Masters of Software Engineering (or equivalent)
 - Previous experience conducting compliance findings against the relevant standard.





Software and Certification

- The software development organisation must issue a Design Approval certificate in accordance with Regulation 3.4.3.
- No material difference between software and other technologies for this pillar.
- The DAR should check for:
 - the scope of use for which the design has been approved
 - e.g. flight test or fleet release?
 - any limitations, caveats or conditions



Key Issue – Contractual Support

- The contract or LOA must support the Design Acceptance strategy.
- Therefore, the Design Acceptance strategy must be defined prior to contract or LOA signature.
- For software, the contract must:
 - impose the four airworthiness requirements,
 - impose competence requirements (AEO plus delivery of software specific plans),
 - require delivery of or access to sufficient data to support the compliance finding process, and
 - provide access to software personnel when required.
- For software, the LOA should:
 - require delivery of airworthiness artefacts,
 - provide for support from airworthiness personnel, and
 - provide access to or delivery of documents required to support the airworthiness process.





Aligning RPA and ADF Oversight

- There are differences between the software assurance approaches of the three major military airworthiness authorities.
- ADF
 - Consistent with FAA/EASA: DO-178B or equivalent.
- USAF/USN (some variations within each service)
 - professional judgement used to determine whether software development and verification activities are sufficient
 - may or may not align with DO-178B
 - heavy reliance on structured flight test program (dedicated squadrons conducting structured flight test programs for 3, 6 or 12 months)
- UK MoD
 - Similar to ADF/FAA/EASA: DEF-STAN 00-56 requires developer to propose suitable processes, DEF-STAN 00-55 (now cancelled) used as a benchmark (requirement for formal methods at highest levels).
 - No RPA.



Aligning RPA and ADF Oversight

- Difficult to resolve the above differences:
 - If the focus is restricted to development and lab verification, there are software items that could be accepted under RPA but would be rejected if ADF oversight was applied.
 - Need to look at the bigger picture: USAF/USN do not rely as heavily on software assurance as the ADF does.
- The three approaches are different, but not incorrect.
 - There are many approaches to assuring software integrity.
- The ADF approach is reflective of our relative size:
 - We generally don't design our own aircraft, so we need to rely on aircraft designs that are acceptable to other forces.
 - We don't have a lot of aircraft, so we can't allocate an entire squadron to flight test activities: need to demonstrate software acceptability through design and lab verification as far as is possible.





Summary

- Design Acceptance is a determination of technical acceptability
 - Software Design Acceptance is the same, but there are some pitfalls to be aware of.
- Design Acceptance Strategies usually involve a combination of Informed RPA and ADF oversight.
- Informed RPA may have limited value for software if the ADF configuration is unique.
- Software Design Acceptance
 - Competence: AEO plus software specific OPPD
 - Specification: Four software airworthiness requirements
 - Evidence: Conduct a compliance finding
 - Certification: as per other technologies
- The Software Design Acceptance Strategy and Compliance Finding approach must be determined pre-contract:
 - It is essential that the contract enable the Design Acceptance process.



Questions

