

Applying FAA reuse policy and guidance



“And now you will hear the rest of
the story”

Mike DeWalt
Chief Scientist, Aviation Systems
Certification Services, Inc.
mike.dewalt@certification.com
Voice +1.360.376.8110

Steve Ward
Principal Engineer
Rockwell Collins, Inc.
saward@rockwellcollins.com
Voice +1.319.295.3043

Some perspectives

- Developer/Approver (DER) – Mike DeWalt (CSI)
- Integrator/Applicant/Approver (DER) – Steve Ward (RCI)
- FAA involvement

This was discussed during earlier presentation of the guidance material. During our practical application of the RSC guidance, the FAA had delegated DO-178B compliance findings but needed to be involved in the findings associated with the RSC guidance. This FAA involvement was motivated by two factors: (a) the RSC guidance is still in draft form, and (b) the FAA has an interest in both the feasibility of initial application of new guidance and its standardization. Moreover, the role of an RSC approval letter in subsequent applications implies greater scrutiny and deeper involvement on the part of the FAA.

Guidance available

- DO-178B/DO-248B
 - Early attempts (OS, legacy)
 - FAQs, DP-10 – Not much help in the general case
- O-8110.49 Chapter 12
 - For use where access to approved data is certain
- AC-RSC (by request)
 - For use where access to approved data may be uncertain

Reuse of software has long been of concern to the airborne community. DO-178B hints at reuse in Chapter 12. DO-248 confronts pieces of the problem in Discussion Paper 10 and several FAQs. Nevertheless, no overall approval strategy had ever been articulated. Any time an applicant proposed something resembling reuse of software, each authority was forced to address its issues on an *ad hoc* basis.

One approach, described in Chapter 12 of Order 8110.49, was aimed at companies wanting to approve a given batch of data once and then reuse that data on any number of subsequent internal projects. Avionics companies often encountered difficulties in trying to create a component that could be sold to other companies. While access to the data is guaranteed when the company owns the TSO, in some cases the data for the avionics software is approved as part of a TC or STC that is not owned by the avionics company. TC- or STC holders might well withhold rights to key data from competitors.

A few standardized components, such as operating systems and some libraries, are emerging. The developers would like to sell the software to multiple avionics manufacturers. The developer may be completely separate from the avionics manufacturer and the STC/TC applicant, making reusability under 8110.49 Chapter 12 impractical. To address this general problem and the issues mentioned in conjunction with the above discussion, AC-RSC was developed.

CSI

RCI

Developer perspective



2003 FAA Software Standardization Conference

4

Not a talking slide – organizational placeholder only.

Selection of guidance

- Practical elements of reuse for a developer
 - Cast as broad a net as possible (independence of platform, application, original approver, etc.)
 - Flexibility comes at a price
- Within a company/organization
 - Access to approved data – Ch 12(8110.49)/RSC

RSC developers naturally prefer broad and flexible approvals for their products. The ideal approval is independent of the target platform, independent of the application, and independent of the original approver.

Such flexibility comes at a price. The broader the approval, the greater the data- and analysis requirements. For example, approval of object code for a single target processor board is inherently less burdensome than approval aimed at multiple targets. In the latter case, analysis would have to show which objectives remained valid following a port of the code to a new target. In short, approvals can be weakened by overly broad claims and inevitably require more effort on the part of the integrator.

Indeed, an RSC developer might decide not to seek RSC approval. A common alternative is simply to provide certification data to be evaluated by each user.

Developer approval issues



This slide will address the practical issues with data submittals and interim approvals.

Issues

- Suitability for reuse
 - Issues
 - Examples (libraries, nav filter, OS, control laws)
- Coordination
- Parallel projects and launch vehicles
- Divining assumptions and integrators' roles
- Sensitivity- and vulnerability analysis
 - User-configurable data
 - Input and output data constraints
- Changes, problem reporting, ADs

The RSC developer has a software program that they wish to obtain certification approval through the RSC guidance.

The first goal is to establish whether software program is suitable for reuse. If significant parts of the program have to be rewritten and recompiled for each new application such that a large portion of the objectives are not transferable between different projects, then the program is probably not a good candidate for reuse. Some examples of good candidates would be library routines and navigation filters that accept latitude and longitude positions. However, a navigation filter that contains specific sensor dependencies might not be appropriate for a candidate. An operating system might be a good candidate if it has minimal dependencies on the underlying target platform or the platform is pervasive. The guidance requires agreement on whether a specific component is a good candidate for reuse. That agreement starts the continuing coordination process.

The guidance strongly encourages early identification of the project as a software reusable component to the regulatory authorities. This is the beginning of a coordination process between the initial integrator, the applicant, the certification authority, and other potential users. In the general case, the RSC developer does not have a direct communication path to the certification authority. The RSC developer is in a "push" mode.

An RSC developer must first establish a customer for its product. The customer should be using the product within a certification or TSO project and be willing to perform the requisite administrative activities needed to bridge the gap between the certification authorities and the RSC developer. If there a number of potential customers, there is usually a need to establish a launch vehicle. A TSO approval independent of an aircraft approval usually has the least overhead and risk. With a TC, an STC, or a TSO that requires concurrent approval with a TC or STC, the RSC approval is dependent on both the equipment approval and the aircraft-level approval. This could delay the RSC approval for reasons that have nothing to do with RSC approvability.

The final approval of an RSC provides transportability of approved objectives. In some cases, the objectives are fully transportable to another user. In others, the objectives are transportable but only if certain assumptions hold. In yet other instances, the objectives are transportable when certain activities have been completed by the user. This requires coordination with both the integrator/applicant and the certification authority. The primary means of accomplishing this is via a table in the PSAC. See extract from next page. This is at the developer's request and has to be coordinated with the integrator/applicant, with the regulatory authority, and to a lesser extent with the end user. The claim may be rejected at any level. The rejection may be based on practicality, lack of data, or incorrectness of a technical argument.

The RSC does not have a constrained environment. In order to ensure that the objectives are valid in this environment, a set of installation requirements as well as a number of restrictions may be required. Installation requirements are usually derived from the basic design. Restrictions are usually determined by analyzing the input- and output data constraints and how the user-configurable data can be set. This is sometimes referred to as a vulnerability- or sensitivity analysis.

Problem reports and changes are more complicated with an RSC. The developer must notify all users of any problems. Each user is responsible for notifying all affected customers. The customers must determine the impact on safety. Changes to an RSC will have to be approved via an STC, TC, or TSO. [Brief discussion of airworthiness directives.]

178B Obj	Obj Description	Resp. Org.	RSC Credit	Assumption	Means of Compliance for the Objective	Activities Remaining For Integrator/Applicant
6-1	Executable Object Code complies with high-level requirements. 6.4.2.1, 6.4.3	RSC Dev	Partial	The object code produced or used by the integrator is identical to that provided by RSC Dev except as modified by the linker.	RSC dev verification records,	The integrator may choose either of two methods to complete the satisfaction of this objective: <ol style="list-style-type: none"> Using the similarity argument contained in Section 6.4.1 of DO-178B, obtain agreement with the certification authority that no additional activity is required if the Integrator's target processor is identical to the target processor that was used by RSC DEV to conduct its testing. This will require a demonstration of object code identity as well. Identity could be demonstrated by configuration identification or a binary comparison. Rerun all of the RSC DEV test vectors and obtain the same results as obtained by RSC DEV.
6-5	Executable Object Code is compatible with target computer. 6.4.3a	A. RSC DEV B. N/A C. N/A	A Partial B None C. None	A. The object code is not modified except by the linking process B. The assembly code is not modified C. The Ada source code is not modified A, B, and C: Any system resource (e.g., memory, etc) and hardware requirements specified in the porting guide must be complied with.	A, B, & C Verification record for target compatibility review of test cases, test results, and test environment description in sufficient detail to permit integrator to establish similarity of their target environment to the test environment.	The Integrator may choose either of two methods to complete the satisfaction of this objective: <ol style="list-style-type: none"> Using the similarity argument contained in Section 6.4.1 of DO 178B, obtain agreement with the certification authority that no additional activity is required if the Integrator's target processor is identical to the target processor that was used by RSC DEV to conduct its testing. This will require a demonstration of object code identity as well. Identity could be demonstrated by configuration identification or a binary comparison. Rerun all of the RSC DEV test vectors and obtain the same results as obtained by RSC DEV.
5-1	Source Code complies with low-level requirements. 6.3.4a	RSC DEV	Full	None	RSC DEV verification results	None

Table 2 – Data supporting objectives and responsibilities

Objective #	RSC Dev		Integrator	Supporting Data
5-4	X			Source code review
5-5	X			Requirements trace report
5-6	X			Source code review report
5-7			X	Link map review or other method as decided by the Integrator
6-1	<u>X(P)</u>		<u>X(P)</u>	Integrator to rerun RSC DEV test vectors on integrated code in the target computer. (Sufficient data will be supplied to enable the Integrator to present an identity or similarity argument as discussed in section 6.4.1 of DO-17B)
10-1	X		<u>X(P)</u>	PSAC
10-2	X			FAA acceptance of RSC Dev PSAC
10-3	X		<u>X(P)</u>	SAS/SCI RSC approval letter

Demonstrating compliance

- PSAC/SAS/SCI and reuse table
- Data sheet
- RSC traceability matrices

This slide will address the practical consequences of using the RSC guidance to establish compliance.

AC ref.	AC 20-RSC Guideline	RSC Dev Approach
8. GUIDELINES FOR THE RSC DEV ELOPER		
8-b	For each applicable RTCA/DO-178B objective, the RSC Dev eloper should document the following information (either directly in the PSAC or by reference to other documents) with sufficient detail for certification authority concurrence and use by an integrator or an applicant who will use the RSC:	
8-b.1	Credit being sought for the objective. The PSAC or referenced document should specify if full, partial, or no credit is being sought for the objective.	RSC Dev Plan for Software Aspects of Certification, RSC Objective Credit table
8-c	The following safety-related items should be documented in the RSC PSAC and RSC SAS:	
8-c.1	The software level for the RSC,	RSC Dev Plan for Software Aspects of Certification, Section 6
8-c.2	Any safety objectives or safety-related requirements allocated to the RSC	For the initial integrator there are none. Follow on integrators will address this relative to their system safety assessment.
8-c.4	Architectural and design features supporting any portion of the safety analysis, partitioning or other protection strategies.	TBD-3
8-i.8(b)	Irrespective of any legal and proprietary issues and agreements about the delivery of software life cycle The data may also need to be available to the applicant, if the target system or RSC needs to be modified (reference 14 CFR part 21.301 through 21.305, and FAA Order 8110.4, latest revision).	Integrator will make all RSC Dev data available to the FAA
8-i.8(e)	The RSC Dev eloper should keep a list of all aviation customers buying or using their components to support continued airworthiness across multiple products. It is also recommended that the RSC Dev eloper and users set up a process to share problem reports in a timely manner.	RSC Dev Plan for Software Aspects of Certification, Table 3, Table A-5, Sections 6.1 and 7.6

CSI

Integrator/applicant/approver
perspective

RCI



Not a talking slide – organizational placeholder only.

Integrator/applicant view

- The main difference between the developer and the integrator is that the integrator is responsible for the compliance of the product, including the RSC.
- This is very important for the initial integrator to understand. After the first approval, it should be easier to integrate the developer's RSC data with follow on-integrator's data.

Note: for the rest presentation the term integrator will mean integrator/applicant.

- When I say compliance, I mean the integration, validation, data submittals, and so on, of all data.



Issues

- Coordination with the developer
- Integration of the PSACs
- Interface with the FAA
- Integration of life-cycle data
- Inclusion of AC-20.RSC in the life-cycle plans

Coordination with the developer

- The integrator has the responsibility to work with the developer to
 - Develop and coordinate the regulatory approach
 - Assure the overall compliance approach for the RSC
 - Assure the developer has plans and processes in place to meet DO-178B objectives
 - Identify the RSC life cycle data needed to support the product and continued airworthiness

- The initial integrator is going to be establishing how the plans and data are going to be dovetailed into their plans and data.
- Don't get me wrong, there will still be a fair amount of work for the follow-ons, and they better hope the initial integrator does their job well.

Integration of the PSACs

- A hierarchy of PSACs is an effective way to ensure a smooth integration of the RSC into the integrator's life cycle data.
- The integrator must have a way of aligning PSACs so all compliance objectives are met.
 - The RSC is first of all a software component in the integrator's product and the integrator must ensure the component is capable of being approved.
 - Contractual issues can compromise compliance assurance.

Integration of the PSACs

- A good approach would be to extend the RSC “Compliance Table” into the integrator PSAC and address any DO-178B objectives not covered by the developer.
- This also allows the FAA to be able to assess if all objectives are being addressed and the approach to satisfying each.

- Again this coordination needs to be tight for the initial integrator.

Interface with the FAA

- Since an RSC is a new concept, it is very important to discuss the use of an RSC with the FAA early in the program.
- The integrator PSAC needs to have a clear explanation of how the RSC is being used in the product.
- Need to gain a mutual understanding of where the FAA's concerns are with respect to the use of the RSC and the compliance approach.

- Need to familiarize the FAA with the RSC and its function in the system and find out where their concerns lie.

Interface with the FAA, cont.

- The RSC component must be recognized at the start, so that the AC-20.RSC issues can be identified and addressed.
- The PSACs are the main artifacts for conveying compliance approach to the FAA.
- Understanding the compliance approach and where to assess compliance is a key for developing the FAA audit approach.

Integration of life cycle data

- The integrator needs to work with the RSC developer to understand what life cycle data is planned to be provided with the RSC.
- The integrator is responsible for incorporating the RSC data into their own submittals.
 - RSC problem report tracking and assessment – what is the interface to the developer's process?
 - integration of the RSC into the build and how it will be verified.

Inclusion of the AC-20.RSC^{RCI} guidance

- The integrator needs to address the AC-20.RSC guidance in their plans for the target system.
- The plans will also have to include an assessment of RSC problem reports and their safety effect on the product.
 - safety is an integrator problem – key item for FAA to assess
 - need to assure the developer's obligation is to provide this data, for how long, etc.

Inclusion of AC-20.RSC

- There are a couple of AC-20.RSC issues that will need discussion with the FAA on the approach to addressing:
 - validate the assumptions and claims made by the developer, and
 - validate and verify throughput, timing, memory usage, resource usage and other resource items of the RSC and other installed components

Inclusion of AC-20.RSC, cont.

- Additionally, the plan for RSC change management needs to be defined.
 - AC-20.RSC, 13, states that the original reuse status no longer applies when the RSC is changed
 - Need to have an approach to deciding when and how the changes are allowed to occur
 - Need to define this process – ideally in the PSAC at the start – again make sure that the relationship is in place over the software life to support this.

Integrator/applicant view

- So ...
- It look like the RSC concept can help the integrator save development/verification time, but ...
- It will take more effort in addressing the RSC guidance, coordinating with FAA, documenting the compliance approach, etc.

CSI

RCI

*“Now you know the rest of the
story”*



- Questions?