MANAGING DO-178 COMPLIANCE WITH IBM RATIONAL PLATFORM

Leverage state-of-the-art best practices to develop avionics software

ZARZĄDZANIE ZGODNOŚCIĄ Z DO-178 PRZY UŻYCIU PLATFORMY IBM RATIONAL

Wykorzystanie najlepszych praktyk do tworzenia oprogramowania dla awioniki

Mehmet Kerim Çakmak

Technical Sales, CEE Region

Abstract: DO178 is a standard for avionic software for the aerospace industry which can be used as a guidance to determine if the software product will perform reliability in airborne environment. The standard has 5 different certification levels each of which has a certain set of objectives. The certification levels are determined depending on how critical the system or subsystem is. To address DO-178 objectives, companies need a defined systems and software engineering process that can delineate workflows, inputs, outputs, roles and responsibilities. The major objectives outlined in the DO-178 standard include;

- Requirements engineering
- Design and development.
- Validation and verification
- Engineering tasks, such as configuration and change management

In this presentation, we will try to explain the main principles of the DO178 standard very briefly and try to give an insight about how the objectives of this standard can be satisfied.

Keywords - Safety, DO-178, requirements engineering, systems engineering, aerospace

Streszczenie: DO-178 jest normą dla oprogramowania awionicznego dla przemysłu lotniczego, która może być stosowana w charakterze wytycznych w celu ustalenia, czy oprogramowanie będzie wykazywać niezawodność w środowisku lotniczym. Norma ma pięć różnych poziomów certyfikacji, z których każdy ma określony zestaw celów. Poziomy certyfikacji są ustalone w zależności od tego, na ile krytyczny jest system lub podsystem. W celu spełnienia celów DO-178 przedsiębiorstwa potrzebują zdefiniowanych systemów i procesu tworzenia oprogramowania, które mogą prezentować przebiegi procesów, wartości wejściowe, wyjściowe, role i zakresy odpowiedzialności. Główne cele nakreślone w normie DO-178 obejmują:

- ustalenie wymagań
- projektowanie i rozwój
- walidację i weryfikację
- zadania inżynierskie, takie jak konfiguracja i zarządzanie zmianami.

W niniejszej prezentacji spróbujemy wyjaśnić w skrócie główne zasady normy DO-178 oraz sposób, w jaki cele tej normy mogą zostać spełnione, a odpowiednie poziomy certyfikacji osiągnięte dla tworzonego systemu lub podsystemu. and the relevant certification level can be achieved for a system or subsystem under development.

Słowa kluczowe – Bezpieczeństwo, DO-178, ustalenie wymagań, projektowanie systemów, lotnictwo i kosmonautyka.

IBM Rational DO-178B overview

The DO-178B standard identified considerations for the development of avionics software, including the demonstration of adherence of up to 66 objectives. Despite the term, this standard is a set of *de facto* requirements for the certification of airborne software. The standard classifies the safety level of airborne software into five classes, shown in Table 1.

To meet these objectives, the standard identifies five key plans and a set of work products that must be produced to demonstrate compliance. The plans are typically, but not necessarily, written as textual documents and can generally be reused from project to project with little change once they have been created. The other documents, however, are used as specific guidance or project evidence and are very labor-intensive to create by hand. Effective use of tooling can facilitate the development and management of these work products. Table 2 lists the primary DO-178B lifecycle data and the tooling used to create and manage in the IBM® Rational® DO-178B solution.

The DO-178B standard is concerned with the safety of software delivered within avionics systems to be certified. Its primary goal is to help enable the consistency of the development of deterministically safety software. Since this is an unachievable goal, the standard instead focuses on *process assurance* rather than *system assurance*. This is done through a combination of detailed traceability among related work products, reviews, audits, independence of verification, path testing, and where appropriate, the use of tested, qualified tools. The basic perspective of the standard is that a development organization is guilty of failing to meet the standard unless clear evidence can be provided otherwise. It is the additional level of rigor—and the production and maintenance of this evidence—that generates most of the incremental cost of adopting the standard.

DO-178B is not prescriptive in terms of process or tooling and is not at all concerned with business objectives such as profitability, productivity and efficiency. As the developers of these systems, we, however, must concern ourselves with them, or we run the risk of going out of business.

Your first DO-178B project will likely add between a 25 percent and 40 percent increase in cost over a non-certified project, but the industry average is 75 percent to a whopping 150 percent. This high cost has a number of root cases, but chief among them are a lack of understanding of the overall process, lack of early involvement of the Federal Aviation Administration (FAA)—or its representative, a Designated Engineering Representative (DER)—and low productivity because of ineffective tooling and inefficient processes.

Table 1: DO-178B safety levels

Safety Level	Description	Probability of failure	Objectives	Objectives with independence
A Catastrophic	Failure condition results in a catastrophic condition for the aircraft; major fatalities, including crew	$\leq 10^{-9}$	66	25
B Hazardous	Failure condition results in a major failure condition for the aircraft, reducing the capability of the crew to cope; potential fatal injuries but not to flight crew	$\leq 10^{-7}$	65	14
C Major	Failure condition results in a minor failure condition for the aircraft, resulting in a reduction of safety margins; possibly including injuries	$\leq 10^{-5}$	57	0 (but independent reviews recommended)
D Minor	Failure conditions have no impact on aircraft operational capability and would involve crew actions within their capabilities; possible discomfort to passengers or crew	10^{-5}	28	0
E No Effect	No safety impact, no affect on operational capability of the aircraft or increase in crew workload		0	0

Certain activities stand out in terms of the cost and effort required to provide adequate evidence of compliance. The first of these is traceability, which relates specific data in different activities to ensure that they are consistent. The DO-178B standard requires, for example, that all software requirements are met by the code, and that all code is at hand to meet the software requirements. This means the system has no requirements not met, nor dead code, nor code for which there are no requirements. Traceability also can allow for change impact analysis and helps us reason about the cost and impact of requirements, software and platform changes.

Another activity that can be extraordinarily expensive is coverage testing. DO-178B requires structural code coverage (SC) testing of each line of code visited during verification testing for level C software, decision coverage (DC) testing of each branch that is tested for level B, and Modified Condition Decision Coverage (MCDC) testing of each condition in a branch independently for level A.

The difference can be seen when looking at a basic coding example:

If (a || b && c)

x++;

SC testing requires a single test that executes the TRUE case, for example, a = b = c = 1. DC testing requires two test cases, one TRUE and one FALSE, for example, a = b = c = 1 and

a = b = c = 0. MCDC testing requires, in general, N+1 test cases, where N is the number of conditions in the logic. In

this case four tests are required not the eight possible variants, as is believed by some, as shown in the truth table below¹:

The goal of the Rational DO-178B solution is to help provide evidence of the consistent development of safe software in a cost-effective, efficient and productive manner. This goal is achieved through a combination of process guidance, with project management and governance tooling integrated with engineering tools.

Rational DO-178B solution

The Rational DO-178B solution is designed to address the concerns of systems and software development in the aerospace industry through a combination of best practices, customizable process solutions, tools, tool integrations and services.

Best practices and methods

The Harmony process is a software development process composed from a set of best practices. For avionics applications, we have taken a subset of the practices, mapped them, in detail, to the DO-178B standard, and called the delivery process the Integrated Software Development Process for DO-178B (ISDP-178). The process content is created and managed with the Rational Method Composer (RMC). The content can be published in many different formats, the most common as an HTML website as shown in Figure 1.

Table 2: Truth table

A	B	C	OUTCOME
0	0	0	FALSE
1	0	0	TRUE
0	1	0	FALSE
0	1	1	TRUE

Identifying, codifying and executing the MCDC test cases for a complex piece of software is clearly an extremely expensive undertaking if done manually.

DO-178A required DC testing of the executable object code for level A, but DO-178B can allow MCDC at the source code level as long as it is combined

with a source-to-binary comparison for all relevant code.

Fig.1 ISDP-178

Figure 2 shows the high level overview of work activities for an avionics project. Pre-spiral planning creates schedules and various plans, including the five key plans required for DO-178B compliance:

- Plan for software aspects of certification
- Software development plan
- Software configuration management plan
- Software quality assurance plan
- Software verification plan

The Configuration Management work activity details how updates to the work product baselines are managed and audited. The Define and Deploy the Development Environment specifies how tools and processes are installed, configured and managed. The Control Project activity manages and governs the project and helps update risk management plan, safety and reliability analyses, tools and processes. The Manage Change activity defines the workflow for managing changes to baselined work products.

Most of the developer's work happens within the microcycle activity. This activity is iterated with a four- to six-week cycle time.

The key best practices included within the ISDP-178 process consist of detailed process descriptions and various forms of guidance, including checklists, templates, definitions, tool mentors and examples.

Fig. 2 ISDP process overview

The most important best practices within the ISDP-178 process include:

- Incremental development
- Continuous integration
- Executable requirements modeling
- Continuous execution
- High-fidelity design modeling
- Five views of architecture
- Test-driven development
- Risk management
- Dynamic planning
- Outcome-based metrics for project governance

While detailed description of these practices is beyond the scope of this paper², we will briefly discuss how this process realizes the incremental development process, how requirements are managed and how high-fidelity modeling integrates with other best practices.

The incremental workflow that is known as the *microcycle* in the ISDP-178 process is shown in Figure 3. The time frame for this iteration is generally four to six weeks long. The primary output of the microcycle is a verified and validated version of the avionics software. However, in addition to the tested code baseline,

evidentiary work products are also produced, including requirements and design, traceability matrix, configuration data, quality assurance reports and audits, review results and so on, as outlined in Table 2.

Fig. 3 Microcycle workflow

A difference between a traditional waterfall lifecycle and the ISDP-178 is the incremental nature of requirements development. The software requirements to be realized and validated within a microcycle are fully elaborated only within that microcycle. This is done in the “Define Iteration Software Requirements” activity of Figure 3. This workflow is elaborated in Figure 4.

Fig. 4 Define iteration software requirements

Most of the tasks shown in Figure 4 deal with the mechanics of analyzing the requirements for completeness, consistency, accuracy and correctness through use-case analysis. This work results in new or updated requirements in the software requirements specification in the tasks shown along the right side of the figure.

Figure 5 shows the workflow for the High-Fidelity Modeling activity in Figure 3. In this activity, the functionally correct software baseline is created and unit tested through the application of Test-Driven Development. The inner loop of Figure 5, known as the *nanocycle*, is applied at a high frequency, typically 20 - 30 minutes. In this loop, a small increment is made in the developer's software design—in the Identify Classes and Objects and Refine Collaboration tasks—and code is generated in the Translate task. Consequently, unit tests are

developed and applied to the evolving software in the Develop Unit Test Plan and Validate Collaboration tasks. Thus, small increments are made in the functionality of the software, and the software is then incrementally unit tested to avoid defects. The design tasks—Architectural Design, Collaboration Design and Detailed Design—optimize this functionally correct software baseline at different levels of abstraction. The continuous integration of the software takes place in parallel with this, as shown in the Configuration Management activity in Figure 2. This means that the evolving baselines from different developers are regularly integrated and tested together.

Fig.5 High fidelity modeling

DO-178B mapping

Best practices guide developers and other engineering staff as they develop the various work products needed for a certifiable software product. This guidance includes work product templates, checklists, and other kinds of guidelines. In addition, the process has a detailed mapping of the 66 objectives of the DO-178B standard to the process content. This helps facilitate the creation of the plans to meet the standard, as well as reviews and audits of the process to support certification needs. Figure 6 shows one of the pages of links. Each objective of the standard has its own page, with links to relevant work task, work products, checklists and other forms of guidance.

Fig. 6 Mapping of DO-178B and ISDP-178 process

Process planning

The definition of the process—deciding what you need to do, and in what order while consuming and producing which work products—is only part of the management problem. Filling out the planning side of the equation and details of the application of the process results in a *project plan*. This can be represented in many ways, such as a work item list, a PERT chart or a detailed Gantt chart. This plan assigns dates, efforts and resources so that a project can be estimated and managed.

Tasks defined in the best practice workflows are specified in the Rational Method Composer tool. This allows for both process definition and customization and for automation in planning and enactment. The work tasks defined in RMC result in *task templates* in the planning tools of the IBM Rational Team Concert™ (RTC)

software that are the building blocks of the project plan. These task templates, initiated for particular work efforts, assigned to resources, and given cost estimates, form the backbone of a project plan.

Project enactment

As a project is run, the plan is realized through a process called *project enactment*: the controlled execution of the project plan. This is often a weak area for many organizations, but it is crucial for success. Project enactment is enabled within the Rational Aerospace solution through a combination of monitoring task outcomes, visualizing progress, risks and obstacles, and when appropriate, updating the plan to reflect reality.

The first of these aspects is monitoring, which comes into the picture in the form of Key Performance Indices (KPIs) that measure progress. Common metrics include requirements complete, requirements volatility, project velocity and defect density. These metrics usually can be gathered transparently, without distracting the engineers from their work, through the automation of information gathering and fusion of that information. That is, the engineers needn't stop and fill out reports since the information is gathered naturally as work items are completed and testing is executed.

This leads to the second aspect: visualization of project data. It is important to realize that not all stakeholders can equally obtain all project information. Project managers, customers, team leads, discipline leads, engineers, quality assurance and testing engineers have different responsibilities and are concerned about different information. In RTC, stakeholder-specific dashboards can be created to visualize different aspects of the project data to show only relevant information to each project stakeholder. These dashboards can be updated automatically and dynamically as a project moves forward, requiring no additional effort other than their original definition.

The last aspect of project enactment is updating the plans. A key problem in the management of the creation of avionics software is that plans are made with incomplete information, as with all software projects, and are therefore wrong in at least some aspects. If your plan is to reflect "truth on the ground" of your project, you must update it to reflect actual effort expended, dates of task completion and defect data. Because the RTC tool is designed to integrate the work environment with near-real-time project data and the project plan, it is simpler to update the plan with that project data, improving the accuracy of the project plan as the project progresses.

Process customization

Of course, all companies developing avionics software don't do so in a cultural or process vacuum. This means that the process must be tailored to fit into the overarching process and business culture of your company. Since the process assets are held in the Rational Method Composer software, it is simpler to modify the process content and align it with your company's processes and required work products and goals.

IBM Jazz platform tooling

Process is a crucial piece of the puzzle since it defines what needs to be done, who does it and when it must be done. Once the process is defined, tooling can automate the creation and manipulation of work products. Good tool integration allows for data exchange without information loss. Figure 7 shows the primary tools of the Rational Aerospace solution.

Fig.7 Rational Aerospace solution tooling

The IBM DOORS® tool is a functionality-rich requirements management tool in the industry. In the Rational Aerospace solution, DOORS is responsible for requirements management and the maintenance of traceability.

The IBM Rhapsody® tool provides state-of-the-art modeling capabilities for UML and SysML, systems engineering and software development. These capabilities include model-based debugging and testing facilities, safety and reliability analysis, integration with other modeling tools such as MATLAB Simulink, and production-quality code generation.

The Rational Quality Manager (RQM) tool provides test management, including test authoring, optimization, scheduling and execution. RQM supports collaboration with metrics tracking and quality dashboards.

The Rational Test Real Time (RTRT) tool executes system and software tests, including the laborious structure, decision and modified condition decision coverage (MC/DC) testing. It is fully qualified as a DO-178B verification tool. Rational Method Composer (RMC) software supports authoring, management and publishing process assets. The ISDP-178 process content is available in RMC format and can be used to define the processes and to generate process templates to be used for project planning and governance in Rational Team Concert. The Rational Team Concert tool runs atop the Jazz™ open framework and provides project planning and governance solutions. With built-in configuration and change management facilities, it supports project planning through the addition of tasks based on task templates imported from your RMC-compliant process definition. Project governance is supported with highly customizable dashboards to monitor project progress metrics.

Lifecycle data

Tooling exists to automate tasks and help with the creation and management of important work products. The DO-178B process defines a number of different and important work products in projects. Some of the key work products, referred to as lifecycle data, are shown in Table 3, along with the Rational tooling and practices that address them.

Table 3. DO-178B lifecycle data

Abbreviation	Document title	Rational solution tool(s)	Rational solution practice or task	Delivered to FAA?
PSAC	Plan for software aspects of certification	Microsoft Word Template, Rational Method Composer, Rational Team Concert	Pre-spiral planning, create certification plan, review certification plan	Yes
SDP	Software development plan	Microsoft Word Template, Rational Method Composer, Rational Team Concert	Pre-spiral planning, create SW development plan, review SW development plan	No
SVP	Software verification plan	Microsoft Word Template, Rational Method Composer, Rational Team Concert	Pre-spiral planning, create SW verification plan, review SW verification plan	No
SCMP	Software configuration management plan	Microsoft Word Template, Rational Method Composer, Rational Team Concert	Pre-spiral planning, create configuration management plan, create change management plan, review configuration management plan, review change management plan	No
SQAP	Software quality assurance plan	Microsoft Word Template, Rational Method Composer, Rational Team Concert	Pre-spiral planning, create QA plan, review QA plan	No
SRS	Software requirements standard	IBM Rational DOORS, Rational Method Composer, Rational Team Concert	Update and maintain SW requirements, perform SW requirements review	No

Managing DO-178 compliance with IBM rational platform
Zarządzanie zgodnością z DO-178 przy użyciu platformy IBM rational

Abbreviation	Document title	Rational solution tool(s)	Rational solution practice or task	Delivered to FAA?
SDS	Software design standard	IBM Rational DOORS, IBM Rational Rhapsody, Rational Method Composer, Rational Team Concert	High-fidelity modeling, architectural design RT, collaboration design RT, detailed design RT, identify classes and objects	No
SCS	Software coding standard	Rational DOORS, Rational Rhapsody, Rational Method Composer, Rational Team Concert	High-fidelity modeling, architectural design RT, collaboration design RT, detailed design RT, translate	No
SRD	Software requirements data	Rational DOORS	Update and maintain SW requirements, perform SW requirements review	No
SDD	Software design description	Rational Rhapsody	High-fidelity modeling, architectural design RT, collaboration design RT, detailed design RT, identify classes and objects	No
SC	Source code	Rational Rhapsody, Rational Method Composer	High-fidelity modeling, architectural design RT, collaboration design RT, detailed design RT, translate	No
EOC	Executable object code	Project compiler ³	High-fidelity modeling, translate, continuous integration, integrate and build	No
SVCP	Software verification cases and procedures	IBM Rational Quality Manager, IBM Rational Test RealTime	High-fidelity modeling, prepare for verification and validation	No
SVR	Software verification results	Rational Quality Manager, Rational Test RealTime	Verification and validation, run tests, analyze test failure	No
SECI	Software lifecycle environment configuration index	Rational Team Concert or IBM Rational ClearCase® or IBM Rational Synergy	Define and deploy the development environment, set up tools, verify tool configuration and installation, set up configuration management environment, refine and deploy the development environment, change and deliver configuration items	No
SCI	Software configuration index	Rational Team Concert or IBM Rational ClearQuest or Rational Synergy	Configuration management, change and deliver configuration items, perform configuration audit	Yes
STM	Software trace matrix	Rational DOORS, Rational Rhapsody, Rational Quality Manager	Develop requirements, link stakeholder requirements to system requirements, trace use cases to system requirements, define iteration SW requirements, update and maintain SW requirements, identify objects and classes, translate, develop tests	No
PR	Problem reports	Rational Team Concert or Rational ClearQuest or IBM Rational Change	Manage change, submit change request, review change request, assign change request, resolve change request, verify change request, close change request	No

Abbreviation	Document title	Rational solution tool(s)	Rational solution practice or task	Delivered to FAA?
SCMR	Software configuration management records	Rational Team Concert or Rational ClearCase or Rational Synergy	Configuration management, change and deliver configuration items, perform configuration audit	No
SQAR	Software quality assurance records	Rational Team Concert or Rational ClearCase or Rational Synergy	Review stakeholder requirements, review system requirements, perform UC review, perform SW requirements review, perform review, perform test review	No
SAS	Software accomplishment summary	Word Template, Rational Method Composer, Rational Team Concert	Perform increment review	Yes

What about tool qualification?

A common question regarding tooling for avionics projects has to do with tool qualification. Tool qualification is required when a tool removes manual steps without subsequent verification of the output. The DO-178B standard identifies two tool categories of interest. The first, development tools, refers to tools that could introduce defects into the products. Such tools perform development activities; for example, include requirement management, modeling, editing, compilation and linking. To be qualified, a development tool must have a known and correct output for all possible inputs. Because this is, at the very least, extremely expensive, almost no one uses qualified tools for development. For example, there are no qualified compilers in the market because of the difficulty in such a high degree of quality assurance. As a result, development tools are used, but the output of the tool must be verified. Ultimately, this is done during software verification to ensure that the baselined code has the right behavior.

The other kind of potentially qualified tools are verification tools. These are tools used for testing. It is very common to use qualified verification tools such as Rational Test Real Time to automate the testing process.

Summary

The Rational aerospace solution is an integration of best practices, tooling and tool integrations to help facilitate the development of certifiable avionics software. The practices are captured in the ISDP-178 delivery process and are modified using the Rational Method Composer to align them with different business environments and cultures. ISDP-178 uses the state-of-the-art best practices for embedded software development, including incremental development, continuous integration, test-driven development and high-fidelity modeling, to capture the development work products in a rigorous and efficient way. These practices result in task templates that can be used in Rational Team Concert for both project planning and governance. The practices map to the DO-178B standard to support the creation of plans for certifiable software.

Various practices are supported with integrated automation by a set of tools for requirements, design, coding and testing. These tools help facilitate industry standard OLSC interfaces running on top of the Jazz platform and provide information exchange among tools and stakeholders.

In short, the Rational aerospace solution provides a suitable environment for the development of certified avionics software.

References

- [1] Software Considerations in Airborne Systems and Equipment Certification, RTCA/DO-178B, RTCA, Inc., 1992
- [2] Second Annual Report for Clarification of DO-178B, “Software Considerations in Airborne Systems and Equipment Certification”, RTCA/DO-248A, RTCA, Inc., 2000; “Design Assurance Guidance for Airborne Electronic Hardware,” RTCA/DO-254, RTCA, Inc., 1992
- [3] Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment, SAE/ARP4761, Society of Automotive Engineers, 1996
- [4] Certification Considerations for Highly-Integrated or Complex Aircraft Systems, SAE/ARP4754A Society of Automotive Engineers, 2010
- [5] Boehm, Barry A Spiral Model of Software Development and Enhancement ACM SIGSOFT Software Engineering Notes, August 1986
- [6] Douglass, Bruce Powel Real-Time Agility, Addison-Wesley, 2009
- [7] Douglass, Bruce Powel Real-Time UML 3rd Edition: Advances in the UML for Real-Time Systems, Addison-Wesley, 2004
- [8] Ericson, Clifton Hazard Analysis Techniques for System Safety, Wiley-Interscience, 2005
- [9] Royce, Winston Managing the Development of Large Software Systems, Proceedings of IEEE WESCON, August 1970

Mehmet Kerim Çakmak, PhD. Born in 1973. Completed his formal education in 2004 with PhD degree in High Energy and Particle Physics from the Middle East Technical University in Ankara. He's got many and various certificates in Information Technologies (IT). Recently employed at the IBM, Ankara, as the Client Technical Professional – CEE Region. Earlier experience gained at the Ayrotek Ltd., and Bildem Ltd., Ankara, as the IT consultant.