# Moderne SDE vs. Zertifizierungserhalt und Obsoleszenz

## Spagat in der embedded Avionik-SW-Entwicklung

Dr. Ludger Janauschek, Airbus Defence and Space GmbH

**Bei der SW-Entwicklung eingebetteter Systeme in der Luftfahrt sind verschiedenste Standards zu berücksichtigen. Einer der bekanntesten zivilen Standards ist DO-178C.**
**Der vollständige Entwicklungsprozess muss die Vorgaben der Standards erfüllen, dies beinhaltet insbesondere Tools und Methoden aus dem Bereich SDE.**
**Tools, HW und Methoden im Bereich SDE befinden sich aktuell und permanent in Weiterentwicklung. Um eine moderne und effiziente SW-Entwicklung zu erreichen, sollte man hier aktuell bleiben.**
**Gerade hierdurch ergibt sich ein Konflikt mit dem Erhalt der Zertifizierung, da diese meist auf den Einsatz von bewährten und abgenommenen Tools und Methoden setzt und damit den Zustand der SDE „einfriert". Projektlaufzeiten von mehr als 20 Jahren erschweren diese Situation.**
**Dieser Vortrag wird die Ursachen dieses Spagats zwischen Aktualität und eingefrorenem Zustand, wie auch Methoden – teils Chancen – zur Bewältigung darstellen, aber auch Grenzen aufzeigen.**

Die Nutzungsdauer von Flugzeugmustern überschreitet regelmäßig 40 Jahre. Einige Avionik-Systeme sind in diesen Flugzeugmustern über die gesamte Nutzungsdauer in Betrieb; die meisten für mindestens für 20 Jahre. Zusätzlich zur Nutzungsdauer muss noch die Planungs- und Entwicklungsphase berücksichtigt werden. Hier werden die Entscheidungen für die Technologie (Target-HW, Target-SW, Host-HW, Host-SW/Tools) getroffen und festgelegt.
Über die gesamte Nutzungsdauer hinweg erfolgen jedoch gerade bei der Avionik-SW Anpassungen an die jeweiligen aktuellen Anforderungen. Hierfür ist eine funktionierende SDE sicherzustellen. Um den technologischen Fortschritt in der SW-Entwicklung nutzen zu können, ist eine moderne SDE erstrebenswert.
Bei diesen langen Laufzeiten ist das Thema Obsoleszenz ein ständiger Risikofaktor. Obsoleszenz trifft hier sowohl die Target-HW, wie auch die Host-HW und Host-SW. Kernthema dieses Vortrags ist die SDE, wodurch im Folgenden der Schwerpunkt auf den Host-Aspekten liegt. Die SDE umfasst hier: Host-HW, Host-Betriebssystem mit detaillierter Angabe des Patch-Levels, Cross-Compiler für Target-HW inklusive Debugger, Compiler für Host-Simulation inklusive Debugger, IDE und/oder Editoren, Versionsverwaltungssystem, Testtools.
Rahmenbedingung für die HW-/SW- und Tool-Auswahl werden durch Luftfahrtstandards gesetzt. In der Luftfahrt werden Zertifizierungen für das Flugzeugmuster nach dem jeweiligen Standard der Musterzulassung weiter fortgeführt, d.h. bei einem 40 Jahre alten Flugzeugmuster ist der 40 Jahre alte Standard relevant [1]. Dies gilt in der Regel für Anpassungen. Bei größeren Änderungen oder Neueinbauten werden jedoch regelmäßig die aktuellen Standards angewendet. Dies führt dazu, dass sich bei einem älteren Flugzeugmuster für verschiedene Komponenten viele Standards aus unterschiedlichen Zeiten angesammelt haben. Im Literaturverzeichnis ist eine

Übersicht über Standards gegeben. Es ist keine Seltenheit, dass gleichzeitig Standards von 1980 bis zur Jetztzeit zu berücksichtigen sind.

Dies ist ein wesentlicher Grund für die lange Nutzungsdauer von Avionik-Rechnern. Für den Ersatz muss nicht nur eine neue Zertifizierung im Umfang vergleichbar mit dem der Zertifizierung des Altgeräts durchgeführt werden, sondern darüber hinaus nach den meist strengeren aktuellen Standards. Die aktuellen Standards für Systementwicklung (ein Avionik-Rechner ist ein System bestehend aus mindestens den „items“ HW und SW) ist ARP4754A und für die SW-Entwicklung der Standard DO-178C; hier wird durch „objectives“ und „activities“ der Rahmen für den SW-Entwicklungsprozess festgelegt. Falls Entwicklungsschritte dem menschlichen Ingenieur durch Tools abgenommen werden, so muss das Toolergebnis verifiziert werden oder das Tool muss qualifiziert werden. Die Tool-Qualifizierung ist aus dem DO-178C [13] herausgelöst und im „technical supplement“ DO-330 [14] beschrieben. Analog zu den „software levels“ A bis E der Avionik-SW nach DO-178C gibt es im DO-330 „tool qualification level“ TQL-1 bis TQL-5, welche unter Berücksichtigung der Tool „criteria“ ausgewählt werden. Der Entwicklungsprozess für Tools ist im Wesentlichen der gleiche wie für Avionik-SW gemäß DO-178C, angepasst durch DO-330.

Die Nutzung der Tools und der Betrieb einer SDE ist z.B. für ein spezielles Umfeld in LTR 7030 [5] geregelt. Im zugehörigen Leitfaden [6] wird das Thema SW-Obsoleszenz angesprochen. SW-Obsoleszenz liegt hiernach vor bei technischem Veralten, durch geänderte gesetzliche Vorgaben oder aufgrund wirtschaftlicher Entscheidungen (z.B. Abkündigungen). Gemäß des Leitliniendokuments tritt SW-Obsoleszenz auf, wenn notwendige HW und/oder SW nicht mehr funktioniert oder notwendige Modifikationen (z.B. zur Behebung von Fehlern, zur Anpassung an Betriebsumgebung, zur Weiterentwicklung des Tools, …) nicht mehr durchgeführt werden können. Als Ursachen für SW-Obsoleszenz wird neben der Obsoleszenz der HW folgendes genannt: fehlende/unvollständige Dokumentation, fehlende Nutzungsrechte, fehlendes Personal.

Zum Thema Personal gibt es Regelungen auch in folgenden Standards: DO-178C [13], AQAP 2210 [8], EN9100 [2]. Hier wird im Wesentlichen gefordert, dass das Personal für die jeweiligen Tätigkeiten eingewiesen oder geschult ist und dies zu dokumentieren ist.

Der Langzeitbetrieb einer SDE erfordert zusammengefasst die In-Betrieb-Haltung von HW und SW und das Vorhandensein von geschultem Personal. Mögliche Ursachen für Obsoleszenz in diesen drei Bereichen sind:

- HW: keine Lieferung von Ersatzteilen, kein Support, Abhängigkeit von HW-Konfigurations-Zustand
- SW: Abhängigkeit von SW-Konfigurationszustand (Betriebssystem, Patches), Bindung an die HW, kein Support, Bindung an eine Programmiersprache
- Personal: Fluktuation, Rente, Schulungsanbieter für Spezial-Know-How Avionik-SW und Betrieb der speziellen SDE verschwindet vom Markt.

Meist liegen zudem Abhängigkeitsketten vor, wie z.B. Tool – Betriebssystem – HW. D.h. ein eingefrorenes Tool (z.B. nach Ablauf des Supportvertrages) kann nur mit einer alten Betriebssystemkonfiguration (Version, Patch-Zustand) laufen und diese alte Betriebssystemversion hat wiederum Einschränkungen auf die Host-HW.

Mögliche Lösungen für Obsoleszenz sind:

- Virtualisierung, d.h. Betriebssystemimages inklusive Tools werden eingefroren und auf einer kompatiblen Host-HW (Betriebssystem-Virtualisierung) oder mit einer Zwischenschicht auf einer anderen Host-HW (HW-Virtualisierung) betrieben
- Lagerhaltung, d.h. der notwendige HW-Zustand wird in ausreichender Menge eingelagert.
- geschultes Personal durch Sicherstellung von Schulungen/Einweisungen – für z.B. 20 Jahre alte Tools und Methoden – durch einen externen Schulungsanbieter oder Übernahme dieser Fähigkeiten in-house
- Konzentration auf eine minimale SDE.

Die Möglichkeit von Support durch Hersteller von HW und SW ist meist dadurch stark eingeschränkt bzw. nicht existent, dass Supportverträge jeweils einen aktuellen Patch-Level der HW, des Betriebssystems und der Anwendungs-SW verlangen. Bei einer eingefrorenen SDE ist dies jedoch nicht möglich. Regelmäßig muss hier der fehlende Support durch In-House-Know-How abgedeckt werden. Dies ist ein hohes Risiko.

Die oben genannten Lösungsansätze haben auch ihre Schattenseiten. Der Betrieb von virtuellen Images ist davon abhängig, dass es moderne HW gibt, die diese alten Images betreiben kann. Der Aufwand an Lagerhaltung bedeutet hohe Kosten und Platzbedarf. Auch altern HW und Datenträger im Regal.
Daher ist es wünschenswert, die SDE nicht zu alt werden zu lassen, sondern sich an die aktuellen Entwicklungen rechtzeitig anzuhängen. Dies bedeutet die Herausforderungen der Requalifizierung von Tools nach DO-330 und der Rezertifizierung von Avionik-SW nach DO-178C.

Bei der Umstellung/Modernisierung von Methoden und Prozessen, z.B. Strukturierte Analyse (SA/RT) auf UML und objektorientiertes Design, ist zudem zu beachten, dass hierdurch zusätzlich die „technical supplements“ DO-331 [15] und DO-332 [16] für modellbasierte Methoden und objektorientierte Technologie zu berücksichtigen sind.

**Zusammenfassung**
Die Avionik-SW-Entwicklung stellt einige spezielle Anforderungen und Herausforderungen an die SDE und deren Langzeitbetrieb. Einige Lösungsmöglichkeiten wurden aufgezeigt, jedoch auch Grenzen. Ziel dieses Konferenzbeitrages ist es auch, mit anderen Beteiligten in diesem Thema in Austausch zu kommen, um weitere Lösungen zu erarbeiten und zu finden.

**Abkürzungsverzeichnis**

| | |
|---|---|
| AC | Advisory Circular |
| AQAP | Allied Quality Assurance Publications |
| ARP | Aerospace Recommended Practice |
| HW | Hardware |
| IDE | integrierte Entwicklungsumgebung |
| RTCA | Radio Technical Commission for Aeronautics |
| SA/RT | Structured Analysis with Real-Time Extensions |
| SDE | Softwareentwicklungsumgebung |
| SW | Software |
| UML | Unified Modelling Language |

**Literatur- und Quellenverzeichnis**


[1] AC 20-115C, Advisory Circular: Airborne Software Assurance, U.S. Department of Transportation, Federal Aviation Administration, 2013

[2] DIN EN 9100:2009, Qualitätsmanagementsysteme – Anforderungen an Organisationen der Luftfahrt, Raumfahrt und Verteidigung, Berlin, 2009

[3] Flühr H., Avionik und Flugsicherungstechnik: Einführung in Kommunikationstechnik, Navigation, Surveillance, Springer Vieweg Verlag, 2013

[4] Hinsch M., Industrielles Luftfahrtmanagment: Technik und Organisation luftfahrtteschnischer Betriebe, Springer Vieweg Verlag, 2012

[5] LTR 7030-001, Luftfahrttauglichkeitsrichtline Software, Bundesamt für Wehrtechnik und Beschaffung, Ausgabe 3, Okt. 2001

[6] LTR 7030 Leitfaden, Leitfaden zur Luftfahrttauglichkeitsrichtline Software, Bundesamt für Wehrtechnik und Beschaffung, Version 2, Aug. 2005

[7] NATO AQAP 2110, NATO Quality Assurance Requirements for Design, Development and Production, 2009

[8] NATO AQAP 2210, NATO Supplementary Software Quality Assurance Requirements to AQAP 2110, 2006

[9] Rierson L., Developing Safety-Critical Software, CRC Press, Boca Raton, 2013

[10] RTCA DO-178, Software Considerations in Airborne Systems and Equipment Certification, Washington, DC, RTCA, Inc., 1982

[11] RTCA DO-178A, Software Considerations in Airborne Systems and Equipment Certification, Washington, DC, RTCA, Inc., 1985

[12] RTCA DO-178B, Software Considerations in Airborne Systems and Equipment Certification, Washington, DC, RTCA, Inc., 1992

[13] RTCA DO-178C, Software Considerations in Airborne Systems and Equipment Certification, Washington, DC, RTCA, Inc., 2011

[14] RTCA DO-330, Software Tool Qualification Considerations, Washington, DC, RTCA, Inc., 2011

[15] RTCA DO-331, Model-Based Development and Verification Supplement to DO-178C and DO-278A, Washington, DC, RTCA, Inc., 2011

[16] RTCA DO-332, Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A, Washington, DC, RTCA, Inc., 2011

[17] SAE ARP4754A, Guidelines for Development of Civil Aircraft and Systems, Warrendale, SAE Aerospace, 2010

**Autor**

Dr. Ludger Janauschek verfügt über eine 10-jährige Erfahrung in der SW-Entwicklung für Avionik-Systeme mit Themenschwerpunkten in SW-Architektur, SW-Design, SW-Coding, Komponententest, Projektmanagement und Personalführung und 8-jährige Erfahrung im Bereich Grundlagenforschung. Er ist Lehrbeauftragter (TH) im Bereich System-/SW-Engineering.

**Kontakt**

Email: ludger.janauschek@airbus.com