DO-178C 与 DO-178B 的差别

王云明

［文章来源：http://www.yunmingwang.cn/blog/article.asp?id=241］

1. 前言

在民用航空机载系统的适航体系中，DO-178（机载系统和设备审定中的软件要求）是机载软件的适用标准。它最早由 RTCA 和 EUROCAE 发布于 1982 年，此后在 1985 年和 1992 年进行了两次改版，分别为 DO-178A 和 DO-178B（EUROCAE 的编号为 ED-12A 和 ED-12B）。

DO-178B 虽然具有较好的稳定性，但随着机载软件研制新技术的层出不穷，人们开始意识到需要对 DO-178B 作一定的补充和修订，以适应新的软件研制技术。为此，RTCA 和 EUROCAE 联合成立了 SC-205 和 WG-71，来完成这项艰巨而富有挑战的工作。

SC-205/WG-71 联合委员会于 2005 年 3 月第一次召开联合会议，此后每年两次定期召开联合会议来商讨和修订 DO-178B。随着工作的开展，大家普遍认识到任务的艰巨和进度的落后，因此从 2008 年开始改为每年三次联合会议。

目前，可以认为 DO-178C 和相关补充文档已逐渐进入了最后定稿和审稿的阶段。后面可能还会有文字的上修改，但已经不会再有颠覆性的变化。这样，我们就可以开始谈论将来 DO-178C 和现有 DO-178B 之间的差异了。

本文旨在简要分析二者的差异，作为向广大软件适航关注者和爱好者的预报吧。

2. 改正错误

一个新的版本出来，要把老版本中的错误改掉，这个不用多说。关键是要举例子给大家分享：

知道 DO-248B 的人应该清楚，其第二章是对 DO-178B 的勘误，列举了当时已经发现的 DO-178B 中的一些比较明显的错误。这些错误在 DO-178C 中已经全部改正过来了。

DO-178B 中还有一些很难发现的错误。说它们很难发现，我的意思是说，光光靠阅读几遍标准是发现不了的，只有真正深入钻研过并实践过 DO-178B 的专家才能发现。举个例子来说，DO-178B 附件 A 的表 A-1 的第五个目标说明，对于 D 级软件不需要编写软件标准；而表 A-9 中第一个目标则要求，即使对 D 级软件，QA 活动仍需要保证软件开发进程和软件综合过程符合已批准的软件计划与软件标准。二者之间一个潜在的问题是，D 级软件可以不写软件标准，那 QA 活动就谈不上要保证符合软件标准。像这样的细节，没有深入研究和实施过 DO-178B 的人是发现不了的。当然，对于这样的“错误”，其实也可以“正确”地解读：也即，如果 D 级软件没有软件标准，那么就已经符合了软件标准了。DO-178C 改正了不少这类错误，一方面说明了 DO-178C 是个严格的标准，另一方面也说明制订标准的人员严谨的办事作风。

3. 澄清内容

错误要改正，晦涩的地方也需要澄清。DO-178C 澄清了许多 DO-178B 中写得不清楚、不容易理解、或很容易误解的章节或段落。例如：

DO-178B, 6.4.4.2.b:

> *The structural coverage analysis may be performed on the Source Code, unless the software level is A and the compiler generates object code that is not directly traceable to Source Code statements. Then, additional verification should be performed on the object code to establish the correctness of such generated code sequences. A compiler-generated array-bound check in the object code is an example of object code that is not directly traceable to the Source Code.*

这段文字很难读懂，并且，在 DO-178B 这么多年的应用中，很多人对此一知半解，甚至误解。要正确解读和理解这段文字，除了充分分析句子结构，揣摩和体会其用意，还要结合 DO-178B 中的其它相关文字，如 4.4.2 节。

在 DO-178C 中，专家们对内容进行了修改，对文字进行了抛光，读起来感觉好多了，结果如下：

DO-178C, 6.4.4.2.b:

> *Structural coverage analysis may be performed on the Source Code, object code, or Executable Object Code. Independent of the code form on which the structural coverage analysis is performed, if the software level is A and a compiler, linker, or other means generates additional code that is not directly traceable to Source Code statements, then additional verification should be performed to establish the correctness of such generated code sequences.*
>
> *Note: A compiler-generated array-bound check is an example of generating additional object code that is not directly traceable to the Source Code.*

诸如此类的文字修改和内容澄清还有很多，我们就不再一一列举了。

4. 一套补充文档

自 DO-178B 颁布以来，有许多新的软件研制技术或研制方法出现，如面向对象技术、形式化方法、基于模型的开发和验证等等。尽管 DO-178B 贯彻“面向目标”原则，尽量少地涉及具体的软件研制方法或研制技术，使得该标准能比较长期地稳定，但是，也有一些软件研制方法或技术对这些“目标”产生了冲击，使得有些目标不再适用，或者需要增加新的目标，或者给这些目标赋予新的指南和解释。为接纳这些新方法和新技术的使用并对其提出相应指南，同时继续保持该标准的稳定性和可扩展性，DO-178C 委员会决定不对 DO-178B

标准的正文直接进行修改和扩充以适应这些新技术和新方法。相反地，将来的 DO-178C 会有一份核心文档（Core Document，其内容类似于 DO-178B）和若干补充文档（Supplements，针对每一项新技术或方法的应用应调整的目标和相关指南）。

值得指出的是，DO-178C 的核心文档与补充文档应该是结合使用的。例如某申请人根据自己项目的实际情况使用了新技术 X 和 Y，那么，它应该同时使用 DO-178C 的核心文档以及 X 和 Y 相应的补充文档。

这样做的可扩展性是显而易见的：如果五年以后又有一项新的软件开发技术，可以不修改 DO-178C 的任何内容，而只需要出一个新的补充文档就可以了。

DO-178B 第 12 章的内容涉及到了一些“其它考虑”，预见了一些可能的新技术使用和替代方法。在 DO-178C 中，由于有了独立的补充文档，所以对第 12 章的改动比较大。

5. 更严格的措辞

参加过我做的 DO-178B 基础培训的学员可能都记得，我很欣赏 DO-178B 中极其严格的措词。比较典型的例子是附件（ANNEX）和附录（APPENDIX）的区分、软件开发（software development）和软件研制（software production）的区别、集成过程（不能叫软件集成过程）、审定联络过程（不能叫软件审定联络过程）等等。

我自己是一个绝对的 perfectionist（完美主义者），所以我很关注，从而也很欣赏，DO-178B 的严格措词。于此同时，我也感叹，那些参加 DO-178B 标准制订的专家大概也都象我一样是完美主义者。亲身参加了 DO-178C 制订的工作组以后，我发现，这些完美主义者简直就是偏执狂，对完美的追求远出乎我的想象程度。这不是贬义，其实是一种褒扬，写标准的人，真的应该具有这种专业的完美主义者的品质。

很多时候，一百几十号人为一句话、为一个词、甚至为一个标点，争得面红耳赤达数十分钟，然后表决，采用一票否决制（也就是说，只要有一个人投反对票，这次修改就不能通过，这就是 consensus 的概念）。

从上面的描述大家可以想象得出来，经过这样磨砺出来的 DO-178C 标准的措词的严格程度。下面举几个例子吧：

- DO-178C 严格区分了 Guidelines 和 Guidance 两个不同的概念。在这一区分下，DO-178C 是 guidance 而不是 guidelines。因此，在 DO-178C 的正文中，已经不再出现 Guideline 一词。
- DO-178C 比较严格地区分了 Purpose、Goal 和 Objective 等单词。在不同的情况下，准确地选用了其中一个单词。
- DO-178C 比 DO-178B 更严格了区分了 Activity 和 Process 这两个概念。
- 还有一点，在 DO-178C 中，专有术语（如各项软件生命周期数据）全部要求首字母大写，比如源代码（专有术语）必须写成“Source Code”而不能写成“source code”或“Source code”；目标代码（非专有术语）要写成“object code”（除了出现在句首，句子首字母大写）而不是写成“Object Code”；可执行目标代码（专有术语）要写成“*Executable Object Code*”而不能写成“*Executable object code*”。

怎么样，很偏执狂吧？但是，我欣赏！

6. 强调文档的完整性

DO-178C 比 DO-178B 更加显式地强调了标准的完整性，认为只有综合理解整个文档的所有内容才算真正理解这个标准。我很赞同这个观点。确实，DO-178B 标准各章节的内容互相引用、互相支持、互相补充又互相一致形成一个统一的整体，不是通盘地综合理解是消化不了其精髓的。

我们知道，附件 A 是 DO-178B 标准中很重要的一部分内容。为了在附件 A 里也充分体现完整性，DO-178C 把活动（activity）加入了附件 A 的表格中，参见下面表格中红色的一列。

	Objective		Activity	Applicability by Software Level				Output		Control Category by Software Level			
	Description	Ref	Ref	A	B	C	D	Data Item	Ref	A	B	C	D
1	Communication and understanding between the applicant and the certification authority is established.	9.a	9.1.b 9.1.c	O	O	O	O	PSAC	11.1	①	①	①	①
2	The means of compliance is proposed and agreement with the Plan for Software Aspects of Certification is obtained.	9.b	9.1.a 9.1.b 9.1.c	O	O	O	O	PSAC	11.1	①	①	①	①
3	Compliance substantiation is provided.	9.c	9.2.a 9.2.b 9.2.c	O	O	O	O	SAS SCI	11.20 11.16	① ①	① ①	① ①	① ①

7. 内容的改动

说了前面这么多，固然都是 DO-178C 相对于 DO-178B 的变动，但估计有些人看了觉得很不过瘾，因为到现在为止好象还没有看到很实质性的内容的变化。其实，在 DO-178C 中对 DO-178B 进行实质性的内容变动还真的不多，理由有二：

1. DO-178B 制订的时候，也是采用 consensus（一致同意、一票否决）的投票方式，经过许多专家的深思熟虑，代表了工业界、局方、供应商等各界的共同意见。它作为民航机载软件研制和审定的国际标准，颁布并使用近二十年了，没有发现什么重大的缺陷，因此说，DO-178B 还是一个成熟且稳定的标准。对它做改版的主要原因是应对那些新技术和新方法的出现，而不是修改 DO-178B 标准的内容

2. 制订 DO-178C 的专家组成员中，有许多就是二十年前制订 DO-178B 标准的人员，只不过头发少了许多，也白了许多。这样，自然就不会有太颠覆性的内容变动了。

如果非要说有些什么内容的实质变化，多少还是有一些，现按照改动的大小，列举几个。

7.1. 工具鉴定级别

DO-178B 对工具的鉴定写得不多。它把工具分成了二类，即开发工具和验证工具。然后分别就这二类工具简要地给出了工具鉴定的指南。它主要强调的宗旨是，工具的鉴定应该能够提供使用工具后省略的或自动化的过程同样高的可信度

可以说，DO-178C 对工具鉴定的指南的变动是最大的。但这些变动其实并不违背原来 DO-178B 所述的内容。它把工具鉴定的级别分成了五级，即 TQL-1 到 TQL-5。TQL-1 是对工具鉴定要求最高的级别，相当于 DO-178B 中所述的用于 A 级软件的开发工具的鉴定。而 TQL-5 则是鉴定级别最低的，相当于 DO-178B 中所述的用于 C 级或 D 级软件的验证工具的鉴定。

所以真正的差异在于，当一个验证工具用于 A 级或 B 级软件时，如果该验证工具出错带来的危害可能比较严重的时候（具体定义参见 DO-178C），需要对验证工具的鉴定提高要求，达到 TQL-4。

把工具鉴定分成五级后，DO-178C 的专家组专门为工具的鉴定写了一个补充文档（supplement）。最近，专家组又把工具鉴定的补充文档单独成文，除非以后还有变化，工具鉴定将会成为一个独立的标准，RTCA 和 EUROCAE 会给它分配一个新的号码，如 DO-XXX/ED-XXX。因此，该“补充文档”也可称为“补充标准”。

7.2. 隐藏的目标显式化

大家知道，在 DO-178B 的附件 A 里列举了机载软件研制的 66 个目标。研究过 DO-178B 或者参加过我的基础培训的人员应该知道，除了那些目标之外还有一些所谓的“隐藏的目标”，比如说：

1. 对于 A 级软件来说，我们还需要做目标代码与源代码追踪，这也是一个目标，但没有在附件 A 中出现
2. 对于 QA 来说，应该在软件计划过程中保证软件计划和软件标准已经编写完成并对它们进行了一致性的检查，这也没有在附件 A 中出现

在 DO-178C 中，这些“隐藏的目标”被那些眼光毒辣、追求完美的专家们挑了出来，并写进了附件 A。

7.3. MC/DC 的定义

DO-178C 对 MC/DC 的定义做了调整。除了 DO-178B 原来定义的 MC/DC（通常也叫 Unique-Cause MC/DC）之外，也接受了 Masking MC/DC 和 Short Circuit。

详细的技术细节这里就不说了，有兴趣的朋友可以自己查阅资料。

7.4. 派生需求的反馈

大家知道，DO-178B 要求把软件生命周期中的派生需求（不管是派生的高层需求还是派生的低层需求），都要反馈到系统安全评估过程，以分析这些派生需求对系统安全带来的影响。

DO-178C 对这点做了纠正，它要求这些派生需求反馈到系统生命周期（的各个过程）。当然，这其中包括系统安全评估过程。

7.5. 软件与系统的关系

DO-178B 的第二章阐述了在软件研制过程中与系统相关一些话题，特别涉及到软件生命周期与系统生命周期之间的信息交流、失效状态与软件级别的关系、系统架构与软件级别的关系（如多版本非相似软件、软件分区等）、各类不同软件对系统的影响（如现场可加载软件、用户可更改软件、商业成品软件等）、系统层与软件层的相互验证，这些内容在 DO-178C 里有了一些改动，以更好地反应目前航空工业的一些现实做法。

这些改动主要基于与 ARP4654 标准的专家组的沟通和协调。在 DO-178C 修订的同时 ARP4754 也在修订，因此双方之间通过协调与沟通达成一致是很有必要的。

7.6. 其它

DO-178C 还有一些其它的改动，例如：

- 可追踪性（在第 11 章增加了一项软件生命周期数据，见 DO-178C § 11.21）
- 带参的软件 / 数据（在第 11 章增加了一项软件生命周期数据，见 DO-178C § 11.22；在表 A5 增加了二个目标，见 A5-8 和 A5-9）
- 涉及供应商管理的一些内容。例如：如果申请人采用了 DO-178C 作为符合性方法，那么其所有供应商也应同样采用 DO-178C 作为符合性方法（DO-178C § 1.4.C）

8. 总结

在 DO-178 发展的历史上，DO-178B 可以认为是对 DO-178A 一次颠覆性的修改。现在正在修订即将出版的 DO-178C 则不同，它完全继承、并且尽可能地尊重 DO-178B，在此基础上改正了一些错误、澄清了一些内容、编制了一套补充文档、更强调文档的完整性、更追求严格的措辞，当然也不排除一些内容的实质变动。

本文拟列举 DO-178C 与 DO-178B 的差异，权且作为对 DO-178C 的一个预报，若有归纳不妥或纰漏之处，敬请大家指正。

［文章来源：http://www.yunmingwang.cn/blog/article.asp?id=241］