DO-178C Compliance Checklist: The Pre-Gap-Analysis Self-Check

Run this before you pay anyone for a gap analysis. Six lifecycle areas, twenty-seven questions you can answer from your own evidence, and what your pattern of answers actually means.

How to score it. For each item: yes (you could show an auditor today), partial (it exists but you would not want to show it), or no. A no in planning costs more than a no anywhere else — everything downstream inherits it.

1. Planning

#	Yes / Partial / No	Ask yourself	Red flag if…
1.1	☐ ☐ ☐	Do the five plans (PSAC, SDP, SVP, SCMP, SQAP) exist, approved, for this programme — not templates from the last one?	*Plans are 'being updated' while development runs*
1.2	☐ ☐ ☐	Does the PSAC state the software level and where it came from?	*Nobody can point to the safety assessment behind the level*
1.3	☐ ☐ ☐	Do requirements, design and coding standards exist and match what the team actually does?	*The coding standard names a language version you no longer use*
1.4	☐ ☐ ☐	Are transition criteria defined — what must be true before design, code, test begin?	*Work starts when people are free, not when criteria are met*
1.5	☐ ☐ ☐	Has the SOI schedule been proposed and agreed?	*'We'll contact the authority when we're ready'*

2. Requirements

#	Yes / Partial / No	Ask yourself	Red flag if…
2.1	☐ ☐ ☐	Do high-level requirements exist for every function, each uniquely identified?	*Requirements live in a spreadsheet nobody baselined*
2.2	☐ ☐ ☐	Can you distinguish requirements from design — and do low-level requirements exist as their own level?	*One flat list serves as both*
2.3	☐ ☐ ☐	Are derived requirements identified as such and fed back to the safety process?	*The term draws blank looks*
2.4	☐ ☐ ☐	Is every requirement verifiable as written?	*Requirements say 'shall be fast', 'shall be robust'*
2.5	☐ ☐ ☐	Were requirements reviewed against a checklist, with records?	*Review happened 'in the meeting'*

3. Design and code

#	Yes / Partial / No	Ask yourself	Red flag if…
3.1	☐ ☐ ☐	Does design data exist describing architecture and low-level behaviour?	*The code is the design*

3.2	□ □ □	Does the code trace to low-level requirements, and were standards enforced?	*Static analysis runs but findings are not dispositioned*
3.3	□ □ □	Is every compiler warning either resolved or explained?	*Warnings are suppressed wholesale*
3.4	□ □ □	Are compiler, linker and their options under configuration control?	*Builds differ between machines*

4. Verification

#	Yes / Partial / No	Ask yourself	Red flag if…
4.1	□ □ □	Does every test case trace to a requirement?	*Tests were written from the code*
4.2	□ □ □	Do robustness cases exist — invalid inputs, boundary conditions?	*Only normal-range cases exist*
4.3	□ □ □	Is structural coverage measured at the right criterion for your level, with gaps analysed?	*Coverage is a number with no analysis behind it*
4.4	□ □ □	Were reviews and analyses done where the objectives require them, not just tests?	*'Verification' means 'testing' in every sentence*
4.5	□ □ □	Are verification results under CM, re-runnable, with independence where required?	*Results live in a folder called final_v2_fixed*

5. Configuration management

#	Yes / Partial / No	Ask yourself	Red flag if…
5.1	□ □ □	Is every lifecycle artifact — not just code — under configuration control?	*Requirements versions exist only in email*
5.2	□ □ □	Do baselines exist, and can you rebuild any of them?	*A rebuild of last year's release is an adventure*
5.3	□ □ □	Is there a problem-reporting system whose reports are classified and dispositioned?	*Bugs are fixed silently, no record*
5.4	□ □ □	Can you produce the Software Configuration Index for what you would ship today?	*The SCI is a job for later*

6. Quality assurance and liaison

#	Yes / Partial / No	Ask yourself	Red flag if…
6.1	□ □ □	Has SQA audited actual conformance to the plans, with records?	*SQA reviews documents but never watches work*
6.2	□ □ □	Does SQA have authority to withhold approval, and has it ever used it?	*SQA reports to the manager whose schedule it threatens*
6.3	□ □ □	Is someone named as certification liaison, with time allocated?	*The authority relationship is 'handled by the customer'*
6.4	□ □ □	Are you tracking open problem reports with the ship decision in mind?	*The open list is nobody's problem until SOI #4*

Reading your result

Pattern	What it means	Sensible next step

Mostly yes, a few partials	You are audit-ready or close; gaps are polish	Run your own thread-pulls; consider a light pre-SOI review
Yes in development, no in planning/CM	The classic strong-team-weak-process shape	Fix plans and CM first; they are cheap and everything inherits them
Partial everywhere	Evidence exists but would not survive sampling	A focused gap analysis will pay for itself; scope it by this checklist
No in requirements or verification	The expensive kind of gap — rework, not paperwork	Get help before writing more code against unverifiable requirements

What this checklist is not. It is not the DO-178C objectives — the standard's Annex A tables are the authoritative statement, per level, and this checklist deliberately paraphrases rather than reproduces them. Passing here does not demonstrate compliance; it tells you where compliance work should start.