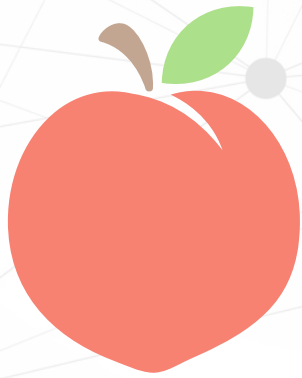




CS Peach

Powered by Cremencing.com

FREE CHEAT SHEET CREMENCING.COM

S/4HANA Readiness Checklist for ABAP Developers



CS Peach
cremencing.com
2026 Edition

"Do not start an S/4HANA migration blind. Run these 15 checks first — they are the same ones an experienced consultant runs before quoting the work."

15-POINT CODE READINESS CHECKLIST

Tickeachoneyourcustomcodepasses.Everyunticked box is migration risk.

- 01

☐

SELECT without ORDER BY. SELECTs that rely on implicit sort order. HANA does not guarantee row order without ORDER BY — add it wherever order matters.
- 02

☐

Pool & cluster table access. Direct reads on former pool/cluster tables (BSEG, KONV) — now transparent. Check secondary-index and SELECT * assumptions.
- 03

☐

Simplification-list tables. Tables merged or removed in S/4HANA — VBUK/VBUP folded into VBAK/VBAP, new MATDOC, etc. Cross-check the Simplification Item Catalog.
- 04

☐

Material number field length. Code assuming 18-char MATNR. The 40-char extension breaks hardcoded lengths and MOVES.
- 05

☐

Native SQL & DB hints. ADBC / EXEC SQL / DB-specific hints tied to the old database. Replace with Open SQL or released CDS.
- 06

☐

SELECT * on wide tables. Full-row reads where only a few fields are used. On a columnar store this is a real performance cost.
- 07

☐

FORM / PERFORM routines. Obsolete in ABAP Cloud. Refactor into methods before any cloud-ready move.
- 08

☐

Header lines & OCCURS. Internal tables with header lines, OCCURS, obsolete WRITE lists — all flagged by ATC.
- 09

☐

Unreleased FMs & classes. Function modules / classes not released for cloud. Verify each against the released-API list.
- 10

☐

Direct standard-table writes. Any INSERT/UPDATE/MODIFY on SAP standard tables. Must go through released APIs / BAPIs.
- 11

☐

Modifications vs enhancements. Core modifications create SPDD/SPAU load at every upgrade. Convert to enhancements / BADIs.
- 12

☐

Client-dependent hardcoding. Hardcoded client, language, or system values that break in a converted landscape.
- 13

☐

Access bypassing CDS / auth. Custom reads that skip CDS views and their built-in authorization. Re-point to released CDS.
- 14

☐

Dynpro / SAP GUI-only UIs. Classic Dynpro and list UIs with no Fiori path. Plan OData + Fiori for anything user-facing.
- 15

☐

Missing ABAP Unit coverage. No automated tests around custom logic — you cannot safely refactor for S/4HANA without them.

AM I READY?

Count your ticks

/ 15
READINESS SCORE

- 12–15

Ready. RunATCtoconfirm,thenplan the conversion with confidence.
- 7–11

Fixable. Real work, but nothing structural. Budget a remediation sprint.
- 0–6

Stop. Do a full custom-code audit before committing a migration date.

RUN THESE SAP TOOLS, NOT GUESSWORK

- SAP Readiness Check — landscape-level report.
- Custom Code Migration app — ATC with S/4HANA checks over your Z-code.
- ATC variant S4HANA_READINESS — the developer's daily check.
- Simplification Item Catalog — what changed, table by table.

Classic ABAP — what you have now

- Direct database and standard-table access allowed
- FORM routines, WRITE lists, header lines still work
- Any FM or class can be called
- Modifications to SAP code are possible (with SPAU pain)
- Runs on the classic on-premise stack

ABAP Cloud — where SAP is going

- Released APIs and CDS only — no direct table access
- Object-oriented only — no FORM, no classic lists
- Strict syntax + ATC gate before activation
- Extensions sit on top — clean core, no modifications
- Same model on-premise, private & public cloud

5 things to rewrite

these will not survive as-is

- 1 Native SQL / DB hints — move to Open SQL or CDS.
- 2 FORM/PERFORM logic — refactor into methods.
- 3 Direct writes to standard tables — use released BAPIs/APIs.
- 4 Core modifications — convert to BADIs / enhancements.
- 5 Dynpro-only UIs — re-expose as OData + Fiori.

5 things to keep

these carry forward cleanly

- 1 Open SQL that already uses ORDER BY and field lists.
- 2 Well-written classes and interfaces.
- 3 BAdI / enhancement-based extensions.
- 4 CDS views and OData services you already built.
- 5 Code with ABAP Unit test coverage — safe to refactor.

Checks to run first

before you touch any code

- SAP Readiness Check — the landscape report.
- ATC with S4HANA_READINESS — per-object findings.
- Custom Code Migration app on BTP / on-prem.
- Simplification Item Catalog — table-level impact.
- SPDD / SPAU — size your modification load.



Laeeq Siddique

Free resource by Cremencing Solutions cremencing.com
CS Peach-Alconsultant for SAP ABAP teams



CREMENCING
SOLUTIONS



FREE CHEAT SHEET CREMENCING.COM

ABAP for HANA 2.0, Clean Core & ABAP Cloud



"Passing the checklist gets you migrated. Mastering these three topics makes you the developer they cannot replace after the migration."

ABAP for HANA 2.0

what makes a developer irreplaceable

Code push-down

Stop pulling rows into ABAP to loop over them. **Push the work to HANA** with CDS and AMDP so the database does the heavy lifting.

CDS views first

Model your data and calculations as **CDS** views. They run in HANA, carry authorization, and feed OData — the backbone of modern ABAP.

AMDP for heavy logic

When CDS is not enough, **ABAP Managed Database Procedures** let you write SQLScript that runs inside HANA — without leaving ABAP.

The golden rules of HANA SQL

No **SELECT ***, always field lists, always **ORDER BY** where order matters, aggregate in the DB, avoid nested SELECTs in loops.

Why it makes you irreplaceable

Most ABAPers still code like it is 2010. The one who writes HANA-optimized CDS + AMDP is the one the project cannot ship without.

"I don't loop over 100,000 rows in ABAP anymore. I let HANA aggregate and return 12 rows. That is the whole shift."

Clean Core Extensibility

extend without breaking upgrades

What clean core means

Never modify SAP standard. Build extensions on top, through released interfaces, so every upgrade stays smooth and cheap.

The 3 extensibility tiers

On-stack (ABAP Cloud) for tight logic, **side-by-side (BTP)** for decoupled apps, **key-user** for business-config tweaks. Pick the lightest that fits.

Released APIs only

Extend using SAP's **released APIs, BAdIs, and extension points** — check api.sap.com. If it is not released, do not build on it.

Wrap, don't touch

Need custom behaviour on a standard object? **Wrap it behind a released BAdI or event** — never patch the standard code path.

Why it pays

A clean-core system upgrades in days, not months. That saved cost is exactly what you sell to management.

"Every modification you avoid today is an upgrade weekend you get back next year."

ABAP Cloud vs Classic

know the line, cross it deliberately

One language, two rulebooks

ABAP Cloud is the **same language with strict rules**: released APIs only, OO only, ATC-gated. Classic ABAP is the old permissive world.

The three-tier model

ABAP Cloud is the target for all new dev. **Classic** stays for legacy. The **tier interface** lets cloud code call classic safely where unavoidable.

What is banned in Cloud

No direct table access, no **FORM**, no **WRITE** lists, no unreleased FMs, no Dynpro. If ATC flags it, it does not belong in cloud.

Start new work in Cloud

Write every new object in ABAP Cloud from today — even on-premise. It future-proofs the code for any cloud move.

The mindset shift

Classic asks "can I?" Cloud asks "am I allowed, and is it released?" **That discipline is the whole point.**

SAY THIS IN AN S/4HANA INTERVIEW

- **"Is your code S/4HANA ready?"** — "I run ATC with the S4HANA_READINESS variant and work through the findings by simplification item."
- **"How do you optimize for HANA?"** — "Push logic down with CDS and AMDP, field lists only, aggregate in the database."
- **"What is clean core?"** — "Extend through released APIs and BAdIs, never modify standard, so upgrades stay cheap."

KEY TOOLS & TRANSACTIONS

- **Readiness** — SAP Readiness Check · **ATC** · Custom Code Migration app
- **HANA dev** — **CDS** views · **AMDP** · Eclipse ADT · SQL Monitor **SQLM**
- **Clean core** — api.sap.com · **SE18/SE19** BAdIs · Extensibility apps
- **Analysis** — Simplification Item Catalog · **SPDD/SPAU**

THE ONE PRINCIPLE THAT TIES IT TOGETHER

- **Migrate the code, then modernize the developer.**
- Passing the 15-point checklist gets your system to S/4HANA. It does not make you valuable there.
- HANA-optimized CDS + AMDP, clean-core extensibility, and ABAP Cloud discipline — **those three make you the developer the project keeps.**
- **That is the difference between surviving the migration and owning it.**



Laeeq Siddique

Free resource by Cremencing Solutions cremencing.com
CS Peach-Alconsultant for SAP ABAP teams

