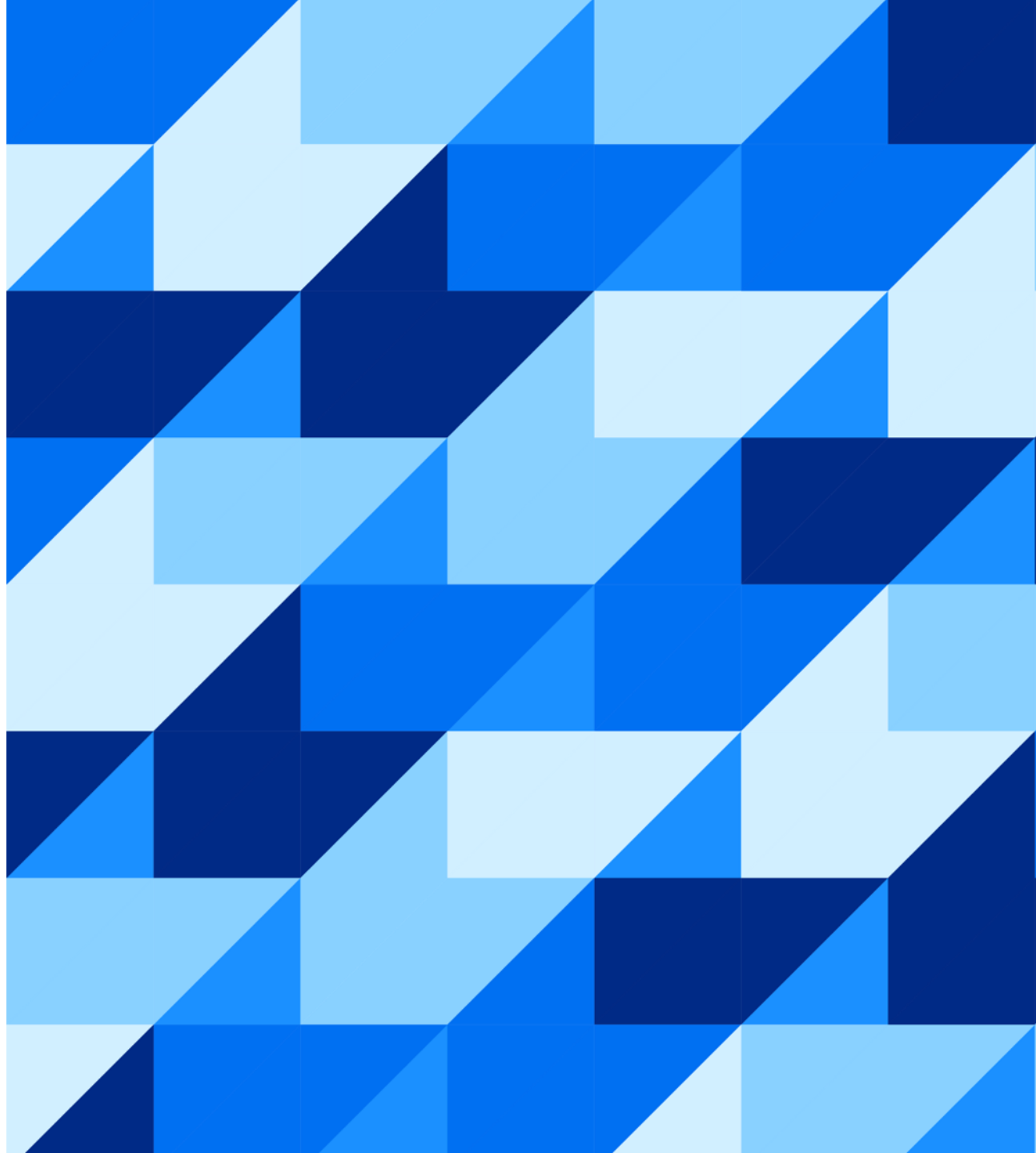




Clean Core und ABAP – Projekterfahrungen

Lukas Bretschneider & Daniel Huser, SAP
June 04, 2025

Public



Agenda



Clean Core – Update 2025

What is new? What has changed?



Measuring Clean Core

Focusing on Governance, outlook on system measurability



Lessons Learned & Best Practices

In reality, clean core relies on experts & governance

Clean Core is a set of 5 guiding principles

The **clean core principles** help ensure customers' business-critical systems remain **agile, cost-effective, and ready to adopt innovations** that drive business forward.

By adopting standard processes, while incorporating stable **extensions and integrations for differentiating processes**, the clean core principles help customers flexibly adapt to changing business requirements and adopt new capabilities

Keep **competitiveness** while **reducing complexity**.

1

Business Processes

Decouple extensions from standard.

2

Extensibility

Control data according to **latest standards**.

3

Data

Keep the landscape **reliable and flexible**.

4

Integration

Keep the operations **effective and efficient**.

5

Operations

Understanding the Need for Clean Extensions:

What It Is and Why It Matters

2

Extensibility

What is a clean extension?

Clean extensions are de-coupled from the core by leveraging released APIs. A clear governance process ensures to leverage the best available option within the SAP S/4HANA extensibility model. This can happen on-stack with ABAP Cloud or side-by-side with SAP BTP.

Clean extensions allow you to differentiate from the competition while staying upgrade-stable

- De-Coupling extensions eases your **upgrade process** and enables you to consume **innovation faster**
- Reducing technical debt and gaining transparency on extensions will allow you **more flexibility**
- Technical de-coupling of extensions can **reduce your time-to-market** from an extension idea to the realization
- Leverage the options of the three-tier extensibility model by using **on-stack** as well as **side-by-side** extensibility (with SAP BTP) capabilities, covering **low-code to pro-code** extension technologies



How to achieve clean extensions?

2

Extensibility

Create decoupled extensions to ensure **upgrade stability** – leveraging **released APIs** from SAP

- Leverage modern extensibility technologies and the **three tier extensibility model**
- Use **released APIs** wherever possible to make extensions not breaking your upgrades – and upgrades not breaking your extensions
- Leverage **on-stack and side-by-side extensibility best** – depending on your use case

Ensure continuous governance – from requirement to design over implementation to deployment

- **Requirement:** Ensure extensions are only created if they are valuable and distinguish you from your competitors
- **Design:** Establish design guidance to ensure the cleanest option is considered
- **Implementation:** Encourage your developers to stick with your design and clean core approach – and document any technical debt accordingly
- **Deployment:** Introduce mandatory code checks including a well-managed exemption process

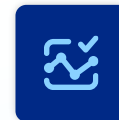
Ensure technical debt is always an informed decision – and manage it professionally

- Ensure new technical debts are **limited** and **always documented** with a clear **justification**
- Ensure the right **balance between governance/strictness and practicability**
- **Housekeeping:** Measure your technical debt and set yourself **ambitious but realistic** goals to reduce it – but be aware: decades of classic custom code will not disappear over night!

LAB PREVIEW

Governance & Operational Maturity

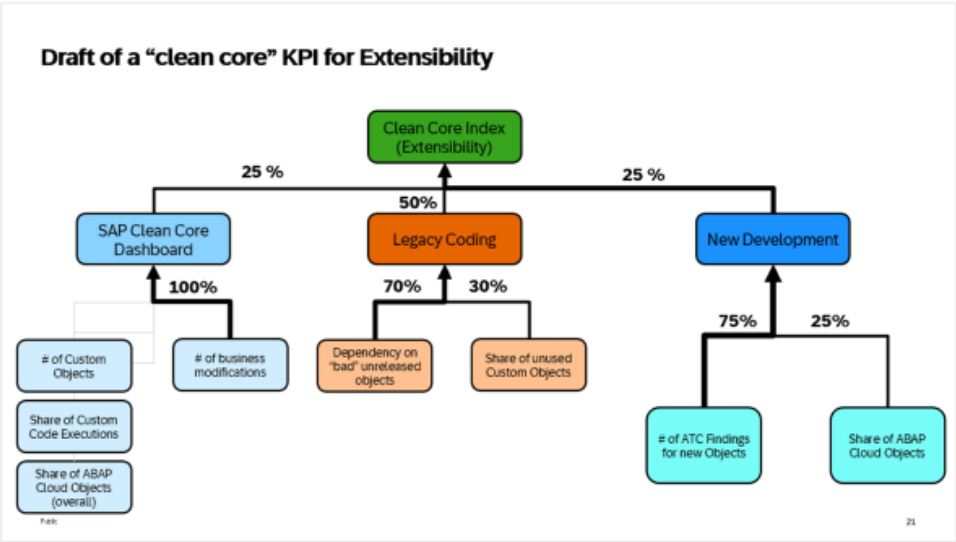
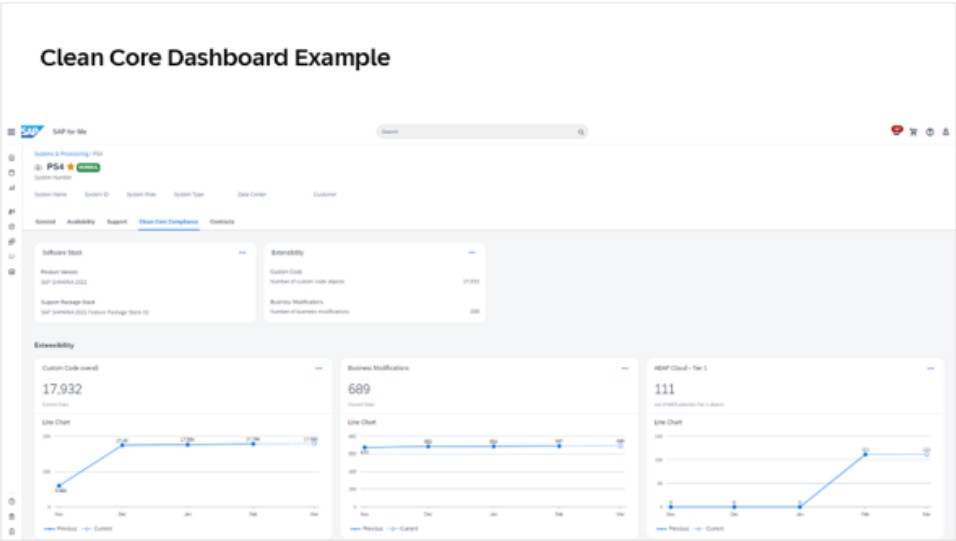
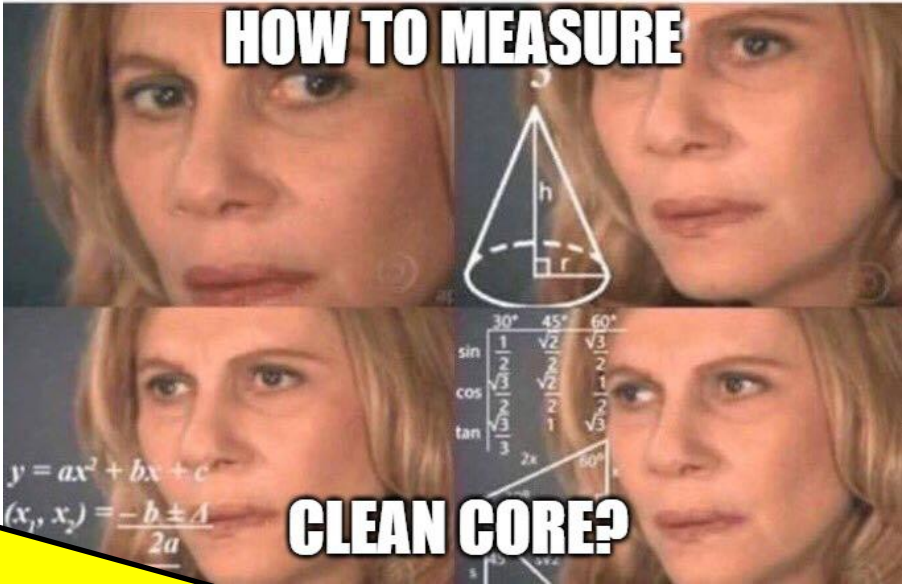
1. Governance Maturity
2. System Setup



KPI Categories

1. Clean Core Share
2. Technical Debt Score
3. Unused Code Share
4. (Business) Modifications

The bad... measurability



ABAPConf 2024
([Youtube](#))

How to Measure Clean Core?

How can I assess my current state of cleanliness, track my progress and improvements over time?



Governance & Maturity

Recommendations regarding tooling, governance and know-how required to get and stay clean.

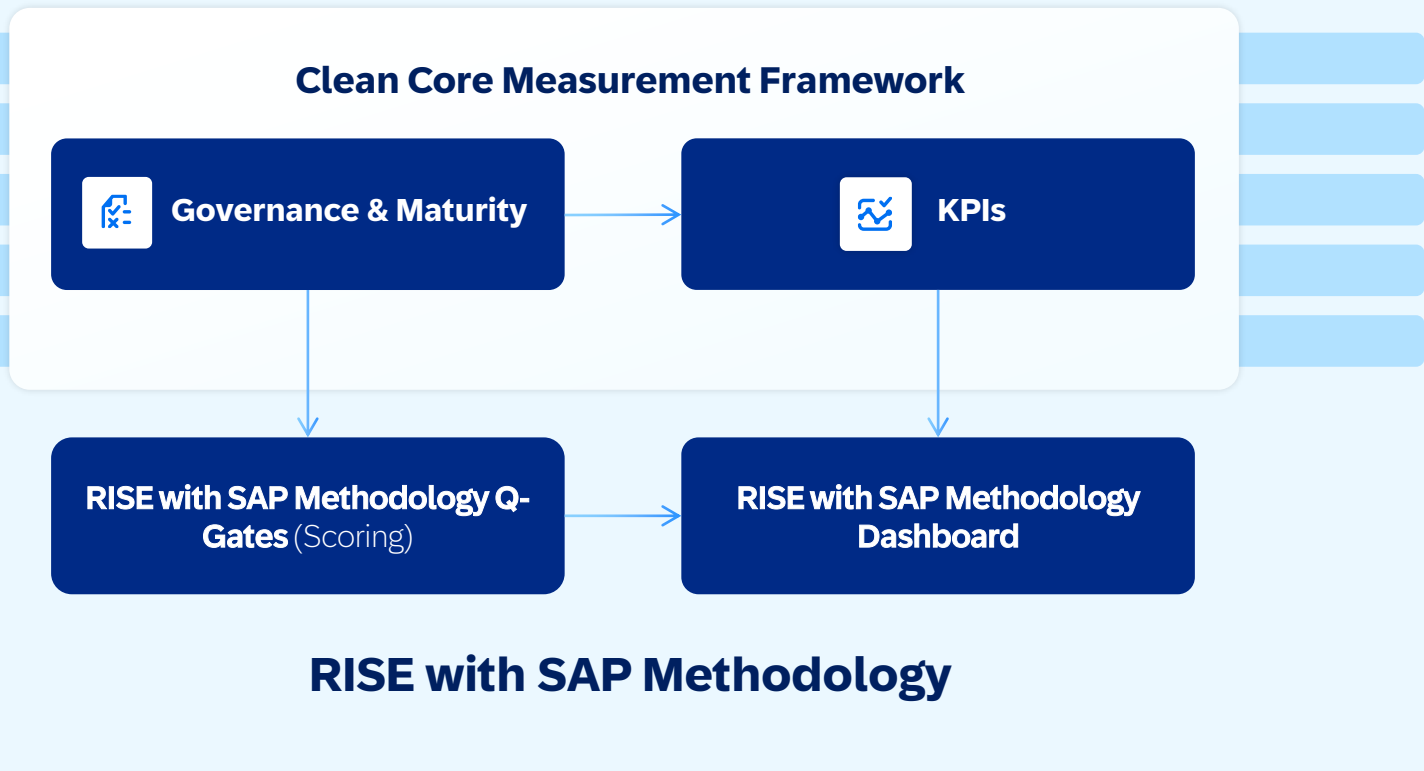
- Process
- Extensibility
- Data
- Integration
- Operations



KPIs

Measure clean-core adherence during a project and business as usual. Derived from analysing system data or data points collected.

Clean Core Measurement Framework enables Customers to successfully move toward **clean core** by assessing their **governance and maturity**, define concrete **actions** and continuously track their success using **clean core KPIs**.



Governance and Operational Practices Extensibility Maturity - Results

LAB PREVIEW

Extensibility Score	CURRENT SCORE	TARGET SCORE
	<0-5>	<0-5>

Governance Maturity (0,00)	Practices	Importance	Current Score	Target Score
	Maturity of Governance Process (for Extensions)	High	<0-5>	<0-5>
	Maturity of Extension Architecture Guidelines	High	<0-5>	<0-5>
	Maturity of Development Guidelines	High	<0-5>	<0-5>
	Maturity of Developer Skills & Enablement	High	<0-5>	<0-5>
	Maturity of Measurements & KPIs	High	<0-5>	<0-5>
	Maturity of Housekeeping Practice	Medium	<0-5>	<0-5>
	Maturity of scoping process for custom code adoption (conversion only)	Medium	<0-5>	<0-5>

System Setup (0,00)	Practices	Importance	Current Score	Target Score
	Setup of Automated code checks	High	<0-5>	<0-5>
	Setup of exemption process	High	<0-5>	<0-5>
	Setup for Three-Tier-Model	Medium	<0-5>	<0-5>
	Setup of Usage Data Collection	Medium	<0-5>	<0-5>
	Setup of SAP BTP Account for Clean Core Extensibility	Low	<0-5>	<0-5>

EXT-GOV-02: Extension Architecture Guidelines

LAB PREVIEW

Practice	Maturity of Extension Architecture Guidelines	IMPORTANCE	CURRENT SCORE	TARGET
		HIGH	<0-5>	<0-5>
Rationale	In order to evaluate the best technical architecture, you ideally have architecture guidelines in place that recommend "cleaner" options over "uncleaner" ones and considers the choice between On-Stack or Side-by-Side extensions in your technical environment.			
Practice Summary	This practice ensures consistent and high-quality extension architecture by establishing comprehensive guidelines, favoring cleaner options (like Tier 1 extensions), and basing extension domain choices on documented, objective criteria. It links guidance to governance processes, enforces the use of released APIs, and mandates thorough documentation and control over architectural decisions and exceptions.			
Evaluation Criteria			Current Score	Target Score
	Is there a guidance implemented that ensures consistent e.g. based on the SAP Application Extension Methodology?		<0-5>	<0-5>
	Is the 3 Tier Extension Model for SAP S/4HANA considered and is the guidance aiming for Tier 1 extensions whenever possible?		<0-5>	<0-5>
	Is the extension domain (On-Stack or Side-by-Side) chosen based on your extension use case? Is there an objective guidance in place to choose the best fitting extension domain?		<0-5>	<0-5>
	Is your guidance on architectural level guiding towards released APIs (local or remote)? Is there a well-defined exception process in place? (e.g. mandatory influence request to SAP in case of missing APIs)		<0-5>	<0-5>
	Is your governance approval process linked to your guidelines? (e.g. choosing a modification needs approval from central architecture board; classic user exit only from local architect)		<0-5>	<0-5>
	Is there a clear documentation process for your decisions on extension architecture established?		<0-5>	<0-5>
	Is there a definition in place on how exemptions in development will be approved and documented upfront? (e.g. allowing package exemptions in ATC only in exceptional cases)		<0-5>	<0-5>

Measuring Clean Core – the SAP way

Manage your gaps and features in SAP Cloud Application Lifecycle Management ([CALM](#)) – e.g. with “clean” tags

Learn about improvements (now and upcoming), especially on:

- [RISE with SAP Methodology Dashboards](#) in [CALM](#)
- [ABAP Test Cockpit \(ATC\)](#)

RISE-Dashboard will include more KPIs in future (Still 2025) – KPI definition ongoing.

- Share of Tier 1 objects
- Share of objects leveraging “classic APIs” (see new ATC check)
- Share of objects with usage of “noAPI” (see new ATC Check)
- Object Score (weighted per findings of the categories above)
- Share of unused Code

Clean core governance preparation
Activate tag display

To display the assigned tags, you need to activate the tag display and save it as your personal or public view. You need to do this for the following overview screens:

- Process Traceability
- Requirements Traceability, Requirements
- User Story Traceability, Tasks
- Feature Traceability

Procedure

- Click on the icon Column Setting activate the checkbox Tags
- Save it as your personal or public

Analytics: Check correctness of tagging
Example solution process traceability: Top down check by pop-up

Use Case: As a project manager I'd like to check, if an as Clean rated solution process is correctly tagged

Procedure

- Filter for processes with tag Clean
- Click on the assigned requirements, user stories and features (planned)
- Click on the total in the right upper corner of the displayed pop-up
- One by one check, if the requirements, user stories and features (planned) are tagged correctly
- In case a standard requirement has not-clean user stories or features assigned:
 - Align with responsible, if the not-clean assignments shall be part of this requirement.
 - Change requirement tag to Not-clean

Clean core governance preparation

Activate tag display

To display the assigned tags, you need to activate the tag display and save it as your personal or public view

You need to do this for the following overview screens

- Process Traceability
- Requirements Traceability, Requirements
- User Story Traceability, Tasks
- Feature Traceability

Procedure

- Click on the icon Column Settings and activate the checkbox Tags
- Save it as your personal or public view

The screenshot displays the 'Solution Process Traceability' interface. A 'Save View' dialog is open, showing the view name 'Solution Process Traceability with tags' and the 'Set as Default' checkbox checked. Below the dialog, a table lists solution processes with their status, requirements, and project tasks. To the right, the 'Columns' panel is open, showing a list of columns with checkboxes. The 'Tags' checkbox is checked, and the 'Column Settings' icon is highlighted in the top right corner of the main interface.

Title	Solution Process Status	Requirements	Stories	Project Tasks
Account Finance	Design	2	3	
CC Customer Accounts Receivable (J59) Finance	Design	1	1	1
Accounts Payable (J60) Finance	Design	1		1

Analytics: Check correctness of tagging

Example solution process traceability: Top down check by pop-up

Use Case: As a **project manager**
I'd like to check, if an as Clean rated
solution process is correctly tagged

Procedure

- Filter for processes with tag Clean
- Click on the assigned requirements, user stories and features (planned)
- Click on the total in the right upper corner of the displayed pop-up
- One by one check, if the **requirements**, **user stories** and **features** (planned) are tagged correctly
- In case a standard requirement has not-clean user stories or features assigned:
 - Align with responsible, if the not-clean assignments shall be part of this requirement
 - Change requirement tag to Not-clean

The screenshot illustrates the SAP Solution Process Traceability interface. The main window, titled "Solution Process Traceability", shows a list of solution processes with columns for Title, Solution Process Status, Requirements, User Stories, Features, and Test Preparation. A search bar and filters are at the top. A pop-up window, titled "Requirement Traceability", is open, showing details for a selected requirement. It includes a table of requirements and a table of user stories. Arrows indicate the flow from the main list to the pop-up and then to the specific items being checked.

Title	Solution Process Status	Requirements	User Stories	Features	Test Preparation
Accounting and Financial Close (J58) Finance	Design	2	2	2	
CC Customer Accounts Receivable (J59) Finance	Design	1	1	1	
Accounts Payable (J60) Finance	Design	1	1		
Accounts Receivable (J59) Finance	Design				

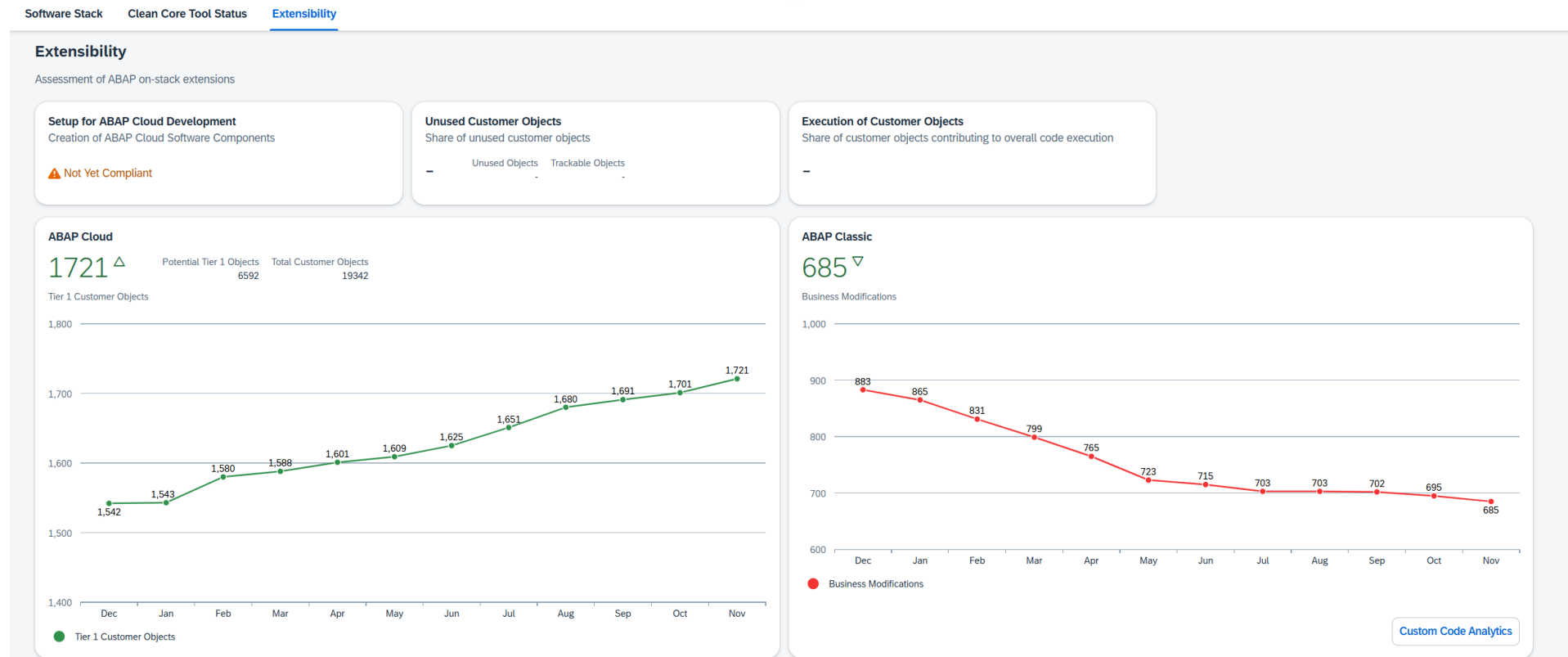
Title	Requirement Status	Features	User Stories	Project Tests	Test Preparation	Test Execution
Accounting and Financial Close Process must provide a comprehensive and flexible general ledger (General Ledger, Chart of Accounts, etc.)	In Realization					
CC Customer Accounts Receivable (J58) Main Process - Establish a new Customer Activity in parallel to Entering Invoice without Sales Order	In Realization					
Reconciliation of various accounts (Debit, Credit) and general ledger accounts must be possible	In Realization					
Requirement for Accounts Payable (J60) - Approval Workflow is required with multiple steps, where this is required	In Realization					
Requirement for Accounts Payable (J60) - Approval Workflow is required with multiple steps, where this is required	In Realization					

Title	Feature Status	Requirement Status	Release	Deployment Status	User Stories
Required Customizing for multiple level Approval Workflow for Accounts Payable	In Implementation				
Accounts Receivable (J58) Main Process - Create automated Check to evaluate Custom Fields entries	In Implementation				
Accounts Receivable (J58) Main Process - Create automated Check to evaluate Custom Fields entries	In Implementation				
Required Customizing for used Chart of Accounts	In Implementation				
Accounting and Financial Close Process must provide a comprehensive and flexible general ledger (General Ledger, Chart of Accounts, etc.)	In Realization				
Accounts Receivable (J58) Main Process - Establish a new Customer Activity in parallel to Entering Invoice without Sales Order	In Realization				
Accounts Receivable (J58) Main Process - Create automated Check to evaluate Custom Fields entries	In Realization				
Required Customizing for used Chart of Accounts	In Implementation				

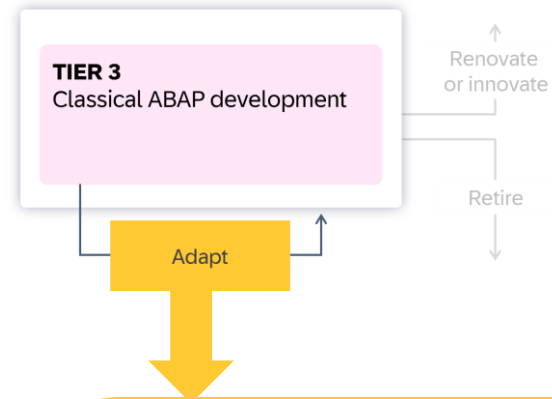
Title	User Story Status	Requirement Status	Feature Status	Sub-tasks	Test Preparation	Test Execution
CC Customer Accounts Receivable (J58) Main Process - Check if the Custom Fields have been filled in	In Progress					
Accounts Receivable (J58) Main Process - Create automated Check to evaluate Custom Fields entries	In Progress					
Accounts Receivable (J58) Main Process - Create automated Check to evaluate Custom Fields entries	In Progress					
Required Customizing for used Chart of Accounts	In Progress					
Accounting and Financial Close Process must provide a comprehensive and flexible general ledger (General Ledger, Chart of Accounts, etc.)	In Realization					
Accounts Receivable (J58) Main Process - Establish a new Customer Activity in parallel to Entering Invoice without Sales Order	In Realization					
Accounts Receivable (J58) Main Process - Create automated Check to evaluate Custom Fields entries	In Realization					
Required Customizing for used Chart of Accounts	In Implementation					

System View – Tab Extensibility

The cards displayed under “Extensibility” give you an overview of your ABAP Cloud (**Customer Objects**) and ABAP Classic (**Business Modifications**) extensions.



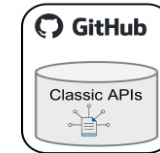
Enhanced guidance for custom development in tier 3



Recommended [classic APIs for tier 3](#) (e.g. CL_GUI_ALV_GRID) provided as JSON file on GitHub

New ATC check variant detecting the classic ABAP custom code which shall be adapted

- Report violation of standard security and quality rules
- Report on modifications and implicit and explicit enhancements
- Report use of critical statements, for example, SYSTEM-CALLs
- Enforce usage of SAP released and classic APIs instead of internal APIs
- Enforce Clean ABAP guidelines



Classic APIs on GitHub

CL_SALV_TABLE Classic API

Class (CLAS)

Application Component: BC-SRV-ALV
Software Component: SAP_BASIS
Main Object Name: CL_SALV_TABLE
Main Object Type: CLAS
Object Type: CLAS

BAPI_PR_CREATE Classic API

Function Module (FUNC)

Application Component: MM-PUR-REQ
Software Component: S4CORE
Main Object Name: 2105
Main Object Type: FUGR
Object Type: FUNC

Labels

Label

remote-enabled

transactional-consistent

New ATC Check based on usage guidelines for SAP APIs

New ATC check Usage of APIs <remote-enabled>

- Checks usages of SAP standard objects in custom code like interfaces, classes, function modules, CDS views, behavior definitions, DDIC database tables and DDIC database views, programs or their subroutines.
- Does not check usages SAP DDIC object types like data elements, domains, table types and structures
- Check behavior
 - **No finding in ATC**
 - ✓ Usage of released APIs
 - **Priority 3 (Info)**
 - ✓ Usage of classic APIs
 - **Priority 2 (Warning)**
 - ✓ Usage of internal Objects (not classified Objects)
 - ✓ SQL read access to SAP database table
 - ✓ SUBMIT statement on programs
 - **Priority 1 (Error)**
 - ✓ Usage of “no API” (with successor if available)
 - ✓ SQL write access to SAP database table
 - ✓ PERFORM statements on external subroutines

Example: Check a simple ABAP report with ALV using

ABAP_CLOUD_READINESS checks:

Findings: 14 Errors	
ABAP Language Version (Syntax) (5 Errors)	
Syntax error in restricted language scope (Open SQL)	SELECT
Syntax error in restricted language scope (Open SQL)	UPDATE
Syntax error in restricted language scope (dynpro)	PARAMETERS
Syntax error in restricted language scope (report)	REPORT
Syntax error in restricted language scope (report)	START-OF-SELECTION
Allowed Object Types in Cloud Development (1 Errors)	
Objects of type PROG are not allowed in ABAP Cloud Development	
Usage of Released APIs (8 Errors)	
Usage of API that will not be released.	BUT000
Usage of API that will not be released.	BUT000
Usage of API that will not be released.	BUT000
Usage of API that will not be released.	BUT000
Usage of not released ABAP Platform APIs.	BU_PARTNER
Usage of not released ABAP Platform APIs.	CL_SALV_TABLE
Usage of not released ABAP Platform APIs.	CL_SALV_TABLE
Usage of not released ABAP Platform APIs.	CX_SALV_MSG

New ATC check Usage of APIs

Findings: 1 Errors, 2 Warnings, 2 Infos	
Errors (1 Errors)	
Updating DDIC database tables or DDIC table views is not allowed (successor available)	BUT000
Warnings (2 Warnings)	
Reading from DDIC database tables or DDIC table views is not recommended (successor available)	BUT000
Usage of internal API	CX_SALV_MSG
Infos (2 Infos)	
Usage of classic API	CL_SALV_TABLE
Usage of classic API	CL_SALV_TABLE

How to measure clean core with ATC (ABAP_CLOUD_READINESS)

ZCL_DEMO

```
DATA: lv_column TYPE salv_de_column.  
[...]  
SELECT * FROM vbak [...] INTO TABLE lt_orders.  
[...]  
CALL FUNCTION 'JOB_OPEN' [...].  
[...]  
CALL FUNCTION '/SCWM/TO_POST' [...].
```

ABAP Test Cockpit (with 3 Tier Model)

Description	Check	Line...	Ref. Object Type	Ref. Object
Findings: 4 Errors				
Errors (4 Errors)				
Usage of not released ABAP Platform APIs.	Usage of Released APIs	3 (IF_C DTEL		SALV_DE_COLUMN
Usage of API that will not be released.	Usage of Released APIs	5 (IF_C TABL		VBAK
Usage of API that will not be released.	Usage of Released APIs	7 (IF_C FUNC		JOB_OPEN
Usage of not released application API.	Usage of Released APIs	9 (IF_C FUNC		/SCWM/TO_POST

Released Objects

VBAK Not To Be Released

Structure, Database Table (TABL)

Application Component: SD-SLS
Software Component: S4CORE
Main Object Name: VBAK
Main Object Type: TABL
Object Type: TABL

Successors

Object Key	Object Type	
I_SALESDOCUMENT	CDS_STOB	>



How to measure clean core with ATC

(3-Tier Check from SAP Note 3565942)


ZCL_DEMO

```
DATA: lv_column TYPE salv_de_column.  
[...]  
SELECT * FROM vbak [...] INTO TABLE lt_orders.  
[...]  
CALL FUNCTION 'JOB_OPEN' [...].  
[...]  
CALL FUNCTION '/SCWM/TO_POST' [...].
```

ABAP Test Cockpit (with 3 Tier Model)

Description	Check
Findings: 2 Warnings, 1 Infos	
Warnings (2 Warnings)	
Reading from DDIC database tables or DDIC table views is not recommended (successor available)	Usage of A
Usage of internal API	Usage of APIs 9 (I
Infos (1 Infos)	
Usage of classic API (successor available)	Usage of internal API
	Usage of APIs 7 (I

Released Objects



Not To Be Released

Database Table (TABL)

Component: SD-SLS
Component: S4CORE
Object Name: VBAK
Object Type: TABL
Object Type: TABL

Errors

Object Key	Object Type
SALESDOCUMENT	CDS_STOB

Classic API

FUNC)

Application Component: BC-CCM-BTC
Software Component: SAP_BASIS
Main Object Name: BTCH
Main Object Type: FUGR
Object Type: FUNC

Object Classifications

Application Component: BC-CCM-BTC
Software Component: SAP_BASIS
Main Object Name: BTCH
Main Object Type: FUGR
Object Type: FUNC

Measuring Clean Core – the {projectname} way

Governance, Governance, Governance!

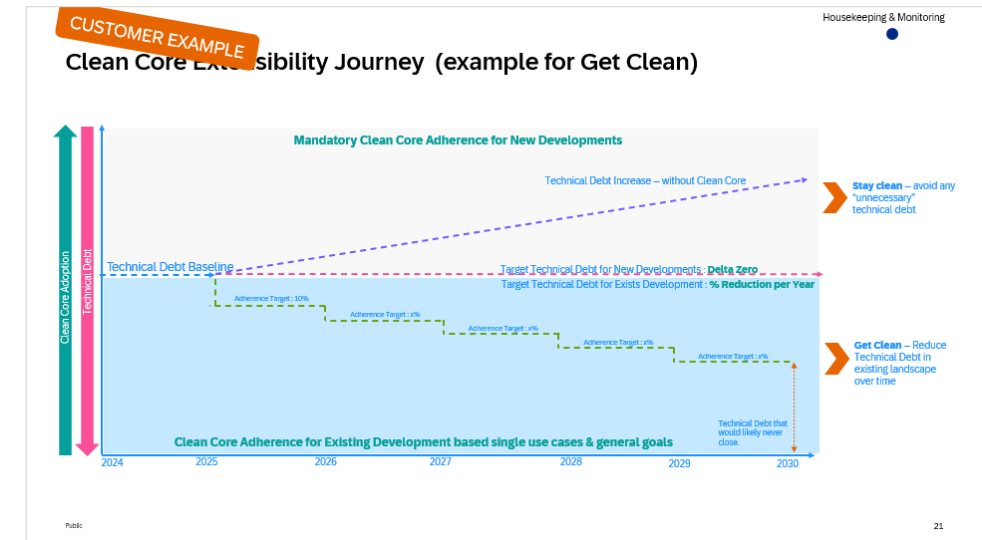
Identify reasonable KPIs (insightful, actionable) for your project – if needed: Build Dashboards on your own based on your project need (e.g. greenfield vs. brownfield has different needs)

Set (realistic) goals to reduce technical debt!

Manage your requirements in “your” tool (project example: Azure DevOps) with a connection to “clean core” and approvals

Leverage ATC Checks and measure ATC results regularly

If needed, leverage custom classification based on [Project Kernseife](#).



Open-source Project: Kernseife

Project Kernseife

- Project “Kernseife” is an open-source custom ABAP Test Cockpit (ATC) check
- Score the technical debt regarding clean core
- This check is an enhanced version of the 3-Tier Model ATC
- Create custom classification to make clean core a reality
- Align your Kernseife score with your strategic goals

Public

CUSTOMER EXAMPLE

Clean Core Extensibility Journey (example for Get Clean)





LinkedIn
Posts
about clean core



Talking about
actual clean
core projects
and experiences

Accidental Modifications



Restrict the “normal”
Developer via
S_DEVELOP

Example:

Services & Support / KBAs & Notes / Basis Components / ABAP Workbench, Java IDE and Infrastructure / Workbench Tools: Editors, Painter, Modeler (BC-DWB-TOO)

2309060 - The SSCR license key procedure is not supported in SAP S/4 HANA

SAP Note, Version: 13, Released On: 11.02.2022

Component: BC-DWB-TOO Category: Consulting Correction: 0 SAP Note/KBA Number: 63
Priority: Recommendations / Additional Info Release Status: Released to Customer Manual Activities: 0
Prerequisites: 0

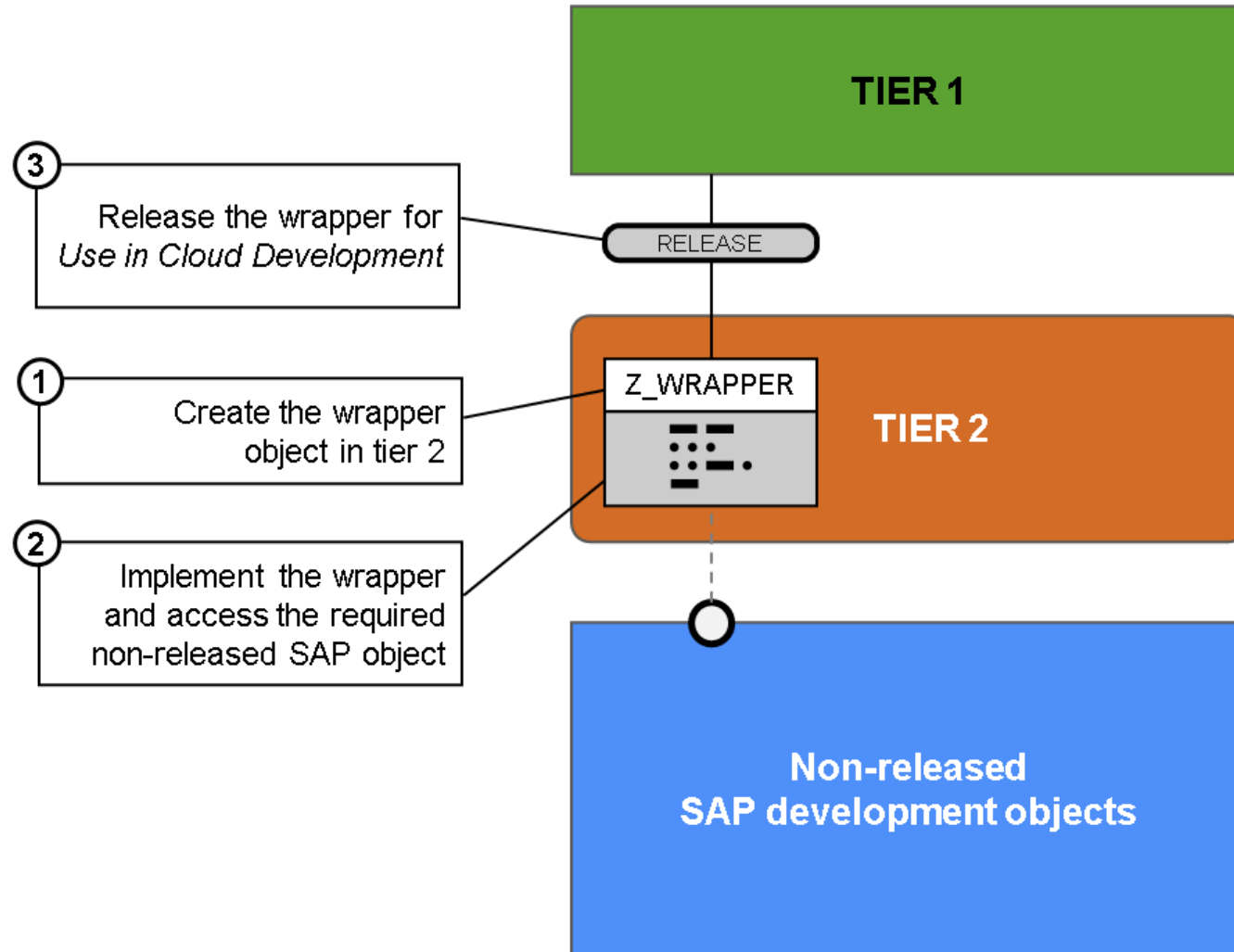
[Description](#) [Software Components](#) [This document is referenced by](#) [Attributes](#) [Available Languages](#)

Symptom

You have developer authorization and use ABAP development tools, but the SSCR license key request is not shown.

Authorizat.	Maintained			ABAP Workbench
DEVCLASS	Standard	63	\$*, T*, Y*, Z*	Package
OBJTYPE	Standard	63	FUGR, LDBA, MESS, MSAG, PROG, TRAN, UDMO, UENO	Object Type
OBJNAME	Maintained	63		Repository object name
P_GROUP	Maintained	63	*	ABAP Program Authorization Group
ACTVT	Standard	63	Add or Create, Change, Display, Delete	Activity

How much Stability is enough?



How much Stability is enough?

API Release: Compatibility of Released Objects

“Exemptions for Incompatible Changes

In some cases, incompatible changes can't be avoided, but due to their potential criticality, evaluation by an expert is required.”

Options:

- Reduce Check Priority
- Deactivate Check
- API Snapshots

<input type="checkbox"/>		Package		API Release: Compatibility of Released Objects	Object was changed incompatibly		05-27-2025	False Positive - finding does ... Interface		05-28-2025
<input type="checkbox"/>		Package		API Release: Compatibility of Released Objects	Object was changed incompatibly		05-27-2025	False Positive - finding does ... Interface		05-28-2025
<input type="checkbox"/>		Package		API Release: Compatibility of Released Objects	Object was changed incompatibly		05-22-2025	False Positive - finding does ... As some custom objects be...		05-23-2025
<input type="checkbox"/>		Package		API Release: Compatibility of Released Objects	Object was changed incompatibly		05-22-2025	False Positive - finding does ... As some custom objects be...		05-23-2025
<input type="checkbox"/>		Package		API Release: Compatibility of Released Objects	Object was changed incompatibly		09-24-2024	False Positive - finding does ... This object is created as par...		09-24-2024
<input type="checkbox"/>		Package		API Release: Compatibility of Released Objects	Object was changed incompatibly		09-24-2024	False Positive - finding does ... This object is created as par...		09-24-2024
<input type="checkbox"/>		Package		API Release: Compatibility of Released Objects	Object was changed incompatibly		09-24-2024	False Positive - finding does ... This object is created as par...		09-24-2024
<input type="checkbox"/>		Package		API Release: Compatibility of Released Objects	Object was changed incompatibly		09-24-2024	False Positive - finding does ... This object is created as par...		09-24-2024
<input type="checkbox"/>		Package		API Release: Compatibility of Released Objects	Object was changed incompatibly		09-24-2024	False Positive - finding does ... This object is created as par...		09-24-2024
<input type="checkbox"/>		Package		API Release: Compatibility of Released Objects	Object was changed incompatibly		09-24-2024	False Positive - finding does ... This object is created as par...		09-24-2024
<input type="checkbox"/>		Package		API Release: Compatibility of Released Objects	Object was changed incompatibly		09-24-2024	False Positive - finding does ... This object is created as par...		09-24-2024
<input type="checkbox"/>		Package		API Release: Compatibility of Released Objects	Object was changed incompatibly		09-24-2024	False Positive - finding does ... This object is created as par...		09-24-2024
<input type="checkbox"/>		Table		API Release: Compatibility of Released Objects	Object was changed incompatibly		05-22-2025	False Positive - finding does ... False positive, as Object ch...		05-23-2025
<input type="checkbox"/>		Table		API Release: Compatibility of Released Objects	Object was changed incompatibly		05-21-2025	False Positive - finding does ... False positive, as Object ch...		05-23-2025
<input type="checkbox"/>		Table		API Release: Compatibility of Released Objects	Object was changed incompatibly		05-19-2025	False Positive - finding does ... False positive, as Object ch... 12-31-9999		05-20-2025

To Wrap or not to Wrap?



```
Method: [colorful bar] active

13  "
14  "
15  "
16
17  CALL FUNCTION 'BAPI_OBJCL_CHANGE'
18  EXPORTING
19      objectkey      = objectkey
20      objecttable    = objecttable
21      classnum       = classnum
22      classtype      = classtype
23      status         = status
24      standardclass   = standardclass
25      changenumber    = changenumber
26      keydate        = keydate
27      no_default_values = no_default_values
28      keep_same_defaults = keep_same_defaults
29      objectkey_long  = objectkey_long
30  IMPORTING
31      classif_status  = classif_status
32  TABLES
33      allocvaluesnumnew = allocvaluesnumnew
34      allocvaluescharnew = allocvaluescharnew
35      allocvaluescurrnew = allocvaluescurrnew
36      return            = return.
37  if sy-subrc <> 0 ##FM_SUBRC_OK.
38      rv_subrc = sy-subrc.
39  endif.
40
41  ENDMETHOD.
```

Scope: \METHOD [colorful bar]



Use ACO_PROXY

SAP

SAP ACo Proxy Generation Utility

✓

Get Metadata

☒ locally

☐ by RFC

☐ by Fileupload

RFC Destination

Function Modules

Create

☒ Class

☐ File

Name of Proxyclass

Package

\$TMP

☐ Create Interface

☐ Create Factory Class

☐ Inactive

☐ Readonly

Options

☒ Pass Destination via Constructor

☐ by IF RFC DEST (Cloud only)

☒ Class-Based Exceptions

☐ Classic Exceptions

☐ Bapi Exceptions

☐ Do not Create Shadows of C1 Released Types

☐ Create Asynchronous Call

☐ Create Retrieve Result Method

☐ Add Keep Task Option

☐ bgRFC

☐ C1 Release

☐ Create private Methods

Additional Options for Variable Functions

☐ Generate Variable Function

Allowlist Generator

Internal Information

Or do it the Kernseife Way:

CLAS - CL_GUI_ALV_GRID
BC-SRV-ALV

General Information	Score
Application Component: BC-SRV-ALV	Rating Code: FW1
Software Component: SAP_BASIS	Adoption Effort: undefined
Release Level: Classic API	Comment: –
Successor Classification: None	

This can be achieved with Standard Checks, but with Kernseife you can define what is “classic” and what is not.

E.g. are you ok with usage of “only” Key-User released Objects?

This also then allows management of usage of objects which are hard to wrap, like interfaces.

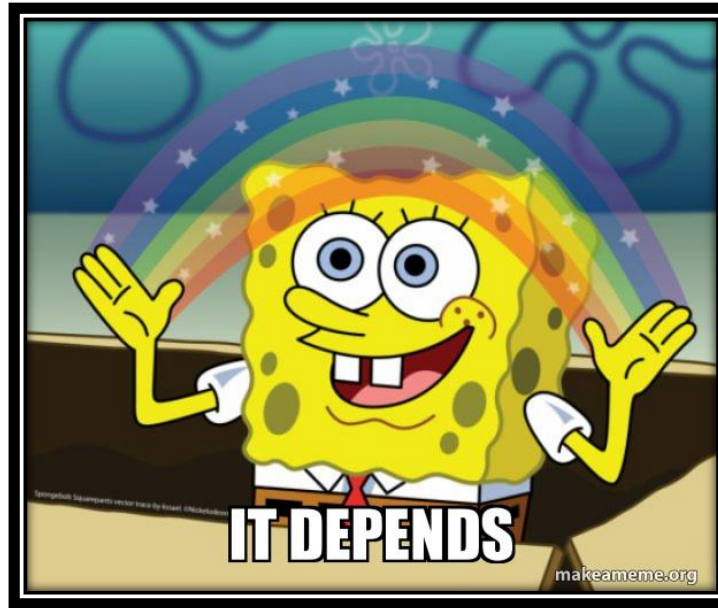
Should I use this SAP object?

Is there a
successor?

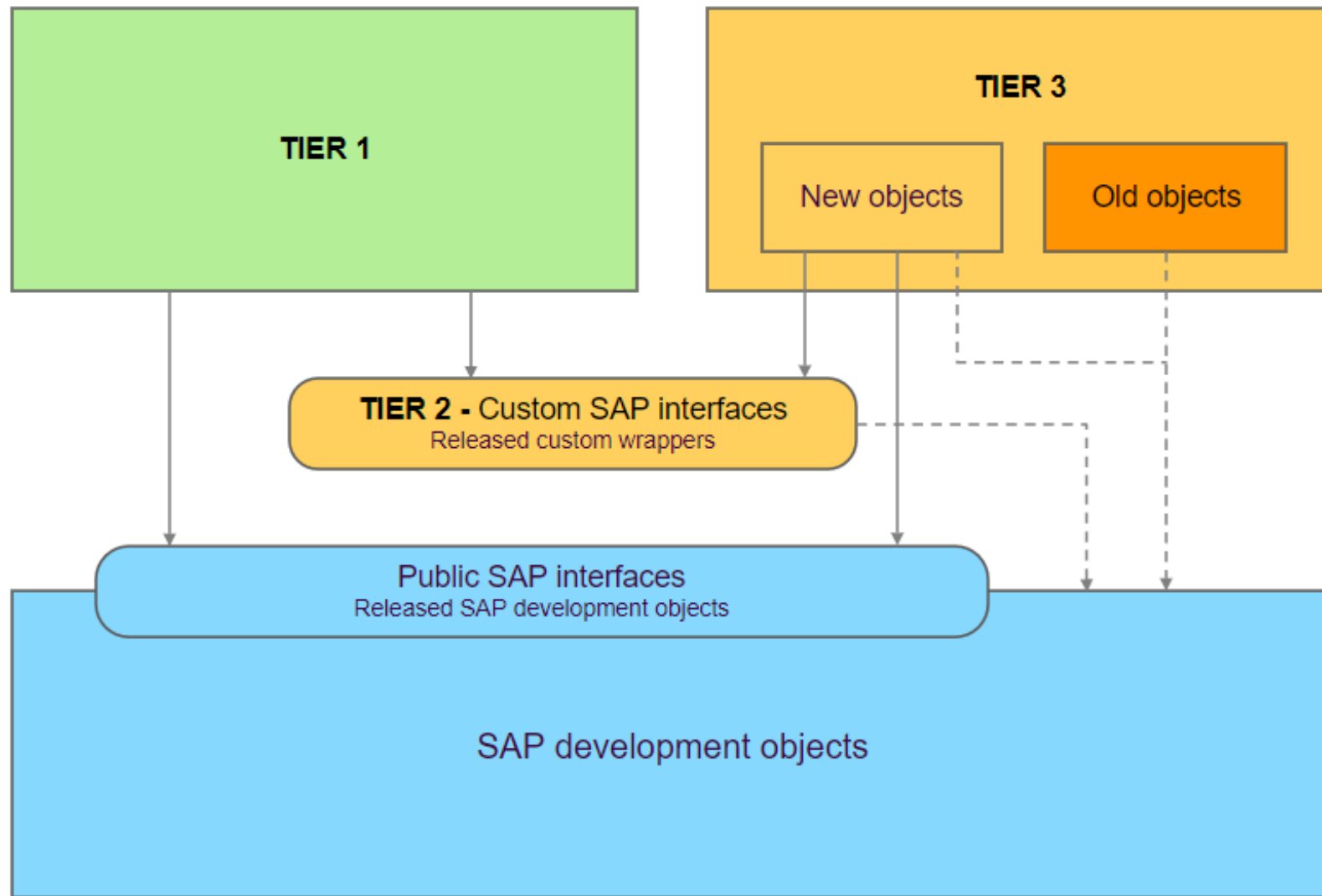
Was this
wrapped
already?

How “bad”
is it?

Solution

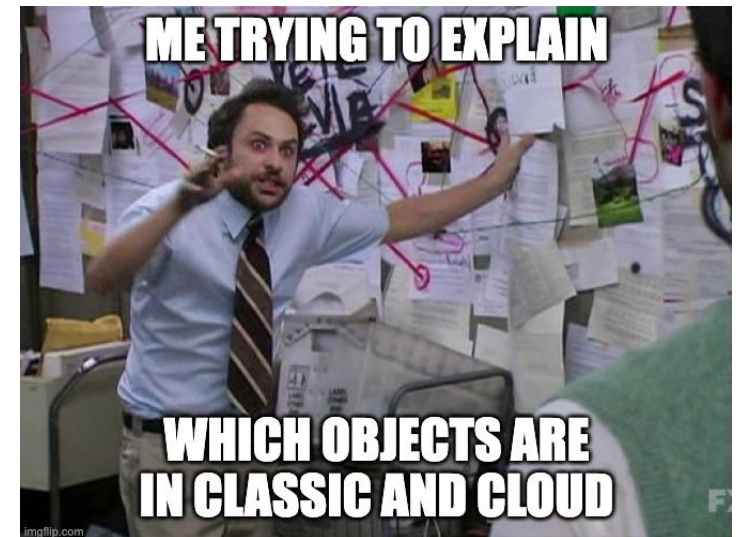


The reality is messy



LEGEND

- Custom objects, ABAP Cloud development model
- Custom objects, classic ABAP, ABAP Cloud rules enforced by ATC
- Custom objects, classic ABAP, ABAP Cloud rules not enforced by ATC
- SAP objects
- ABAP Cloud conform access
- ABAP Cloud non-conform access, controlled by ATC



How to measure clean core with ATC (Project Kernseife)

ZCL_DEMO

```
DATA: lv_column TYPE salv_de_column.  
[...]  
SELECT * FROM vbak [...] INTO TABLE lt_orders.  
[...]  
CALL FUNCTION 'JOB_OPEN' [...].  
[...]  
CALL FUNCTION '/SCWM/TO_POST' [...].
```

ABAP Test Cockpit (with Kernseife)

Description	Check	Ref. Object	Line...
Findings: 1 Errors, 5 Infos			
Errors (1 Errors)			
Reading from SAP database tables or table views (successor available)	Kernseife: Usage of APIs	VBAK	5 (IF_C
Infos (5 Infos)			
Framework - Released	Kernseife: Usage of APIs	IF_OO_ADT_CLASSRUN	1 (IF_C
DDIC - Simple	Kernseife: Usage of APIs	SALV_DE_COLUMN	3 (IF_C
Framework - Recommended (successor available)	Kernseife: Usage of APIs	JOB_OPEN	7 (IF_C
Framework - Released	Kernseife: Usage of APIs	IF_OO_ADT_CLASSRUN	8 7 (IF
Business Function - Recommended	Kernseife: Usage of APIs	/SCWM/TO_POST	9 (IF_C

Cloudification
Repository

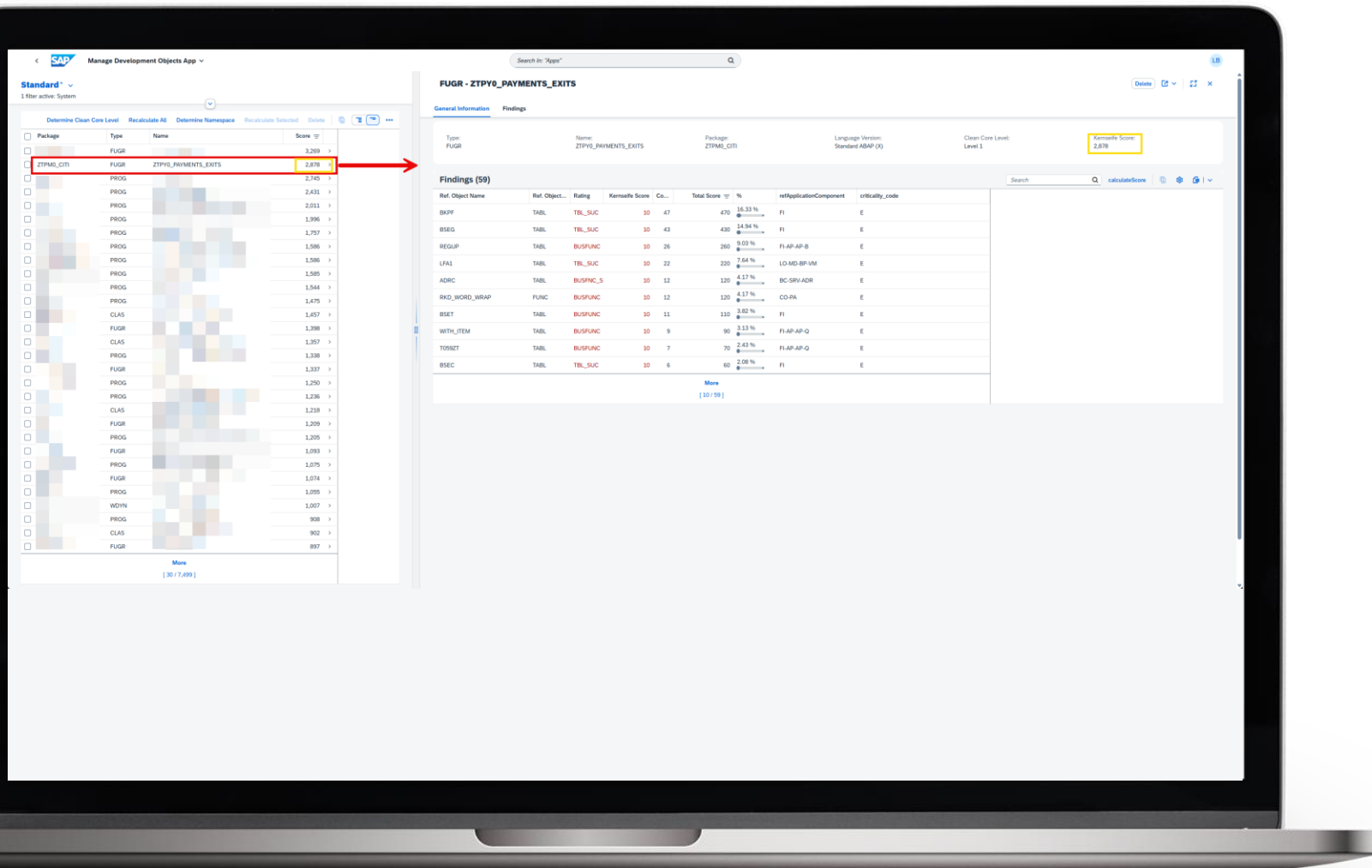
Released Objects

Kernseife Classification (custom)

Classifications (4)				Assign Framework	Assign Successor	Create	Delete
<input type="checkbox"/> Type	Name	Software Component	Rating Code	Release Level	Successor ...	Framework	
<input type="checkbox"/> TRANSP	VBAK	S4CORE	TBL	Must not be used	Standard		
<input type="checkbox"/> FUNC	/SCWM/TO_POST	S4CORE	BF1	Unkown	None		



Open-source Project: Kernseife



Project Kernseife



- Project “Kernseife” is an open-source custom ABAP Test Cockpit (ATC) check
- Score the technical debt regarding clean core
- This check is an enhanced version of the 3-Tier Model ATC
- Create custom classification to make clean core a reality
- Align your Kernseife score with your strategic goals

Agenda / Summary



Clean Core – Update 2025

What is new? What has changed?



Measuring Clean Core

Focusing on Governance, outlook on system measurability



Lessons Learned & Best Practices

In reality, clean core relies on experts & governance

Thank you.

Contact information:

Lukas Bretschneider & Daniel Huser

lukas.bretschneider@sap.com | daniel.huser@sap.com