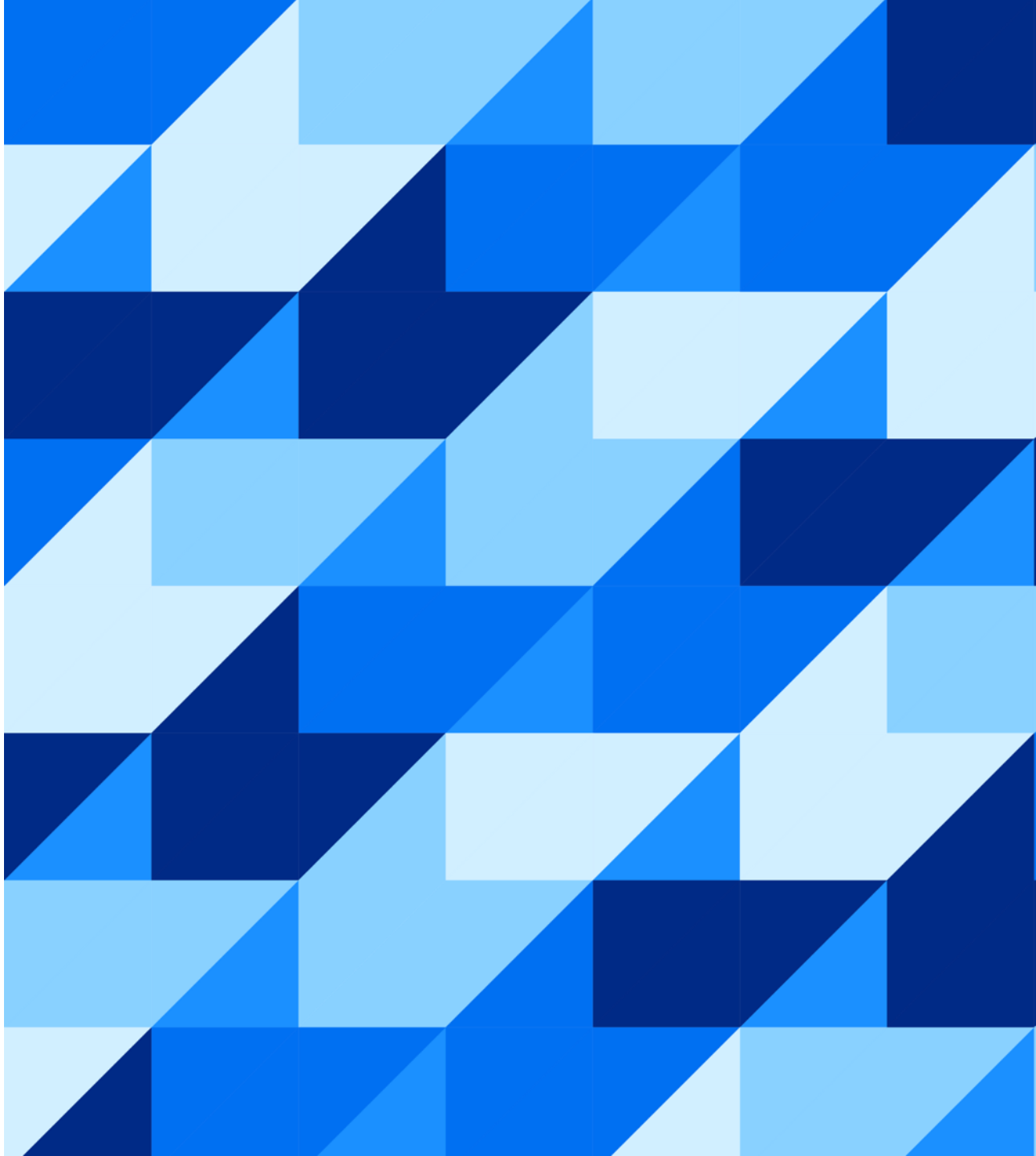


Clean Core & ABAP Cloud - Bericht aus der Praxis

Lukas Bretschneider & Lukas Käser, SAP
June 06, 2024

Public

Agenda

Extensibility Model

Governance Process

THE GOOD

Expectation Management

What are KPIs?

“clean core” developers?

APIfication process

THE BAD

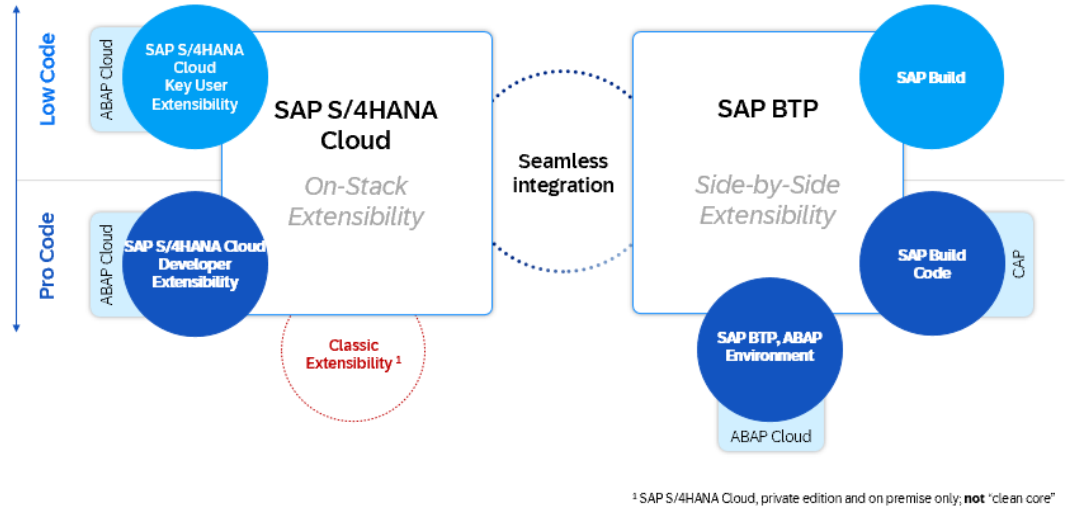
Legacy Code

and **THE UGLY**

THE GOOD...

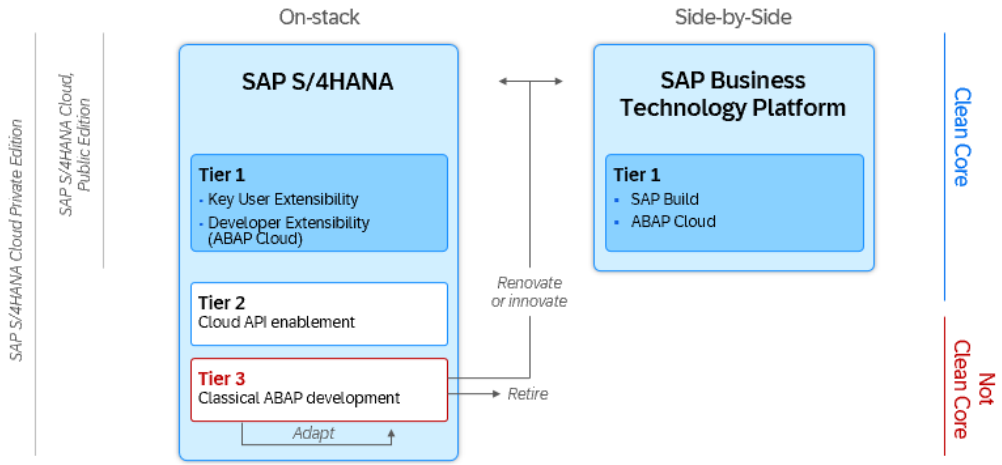
Extensibility Models – welcome to SAPPHIRE!

Extensibility Portfolio for SAP S/4HANA Cloud

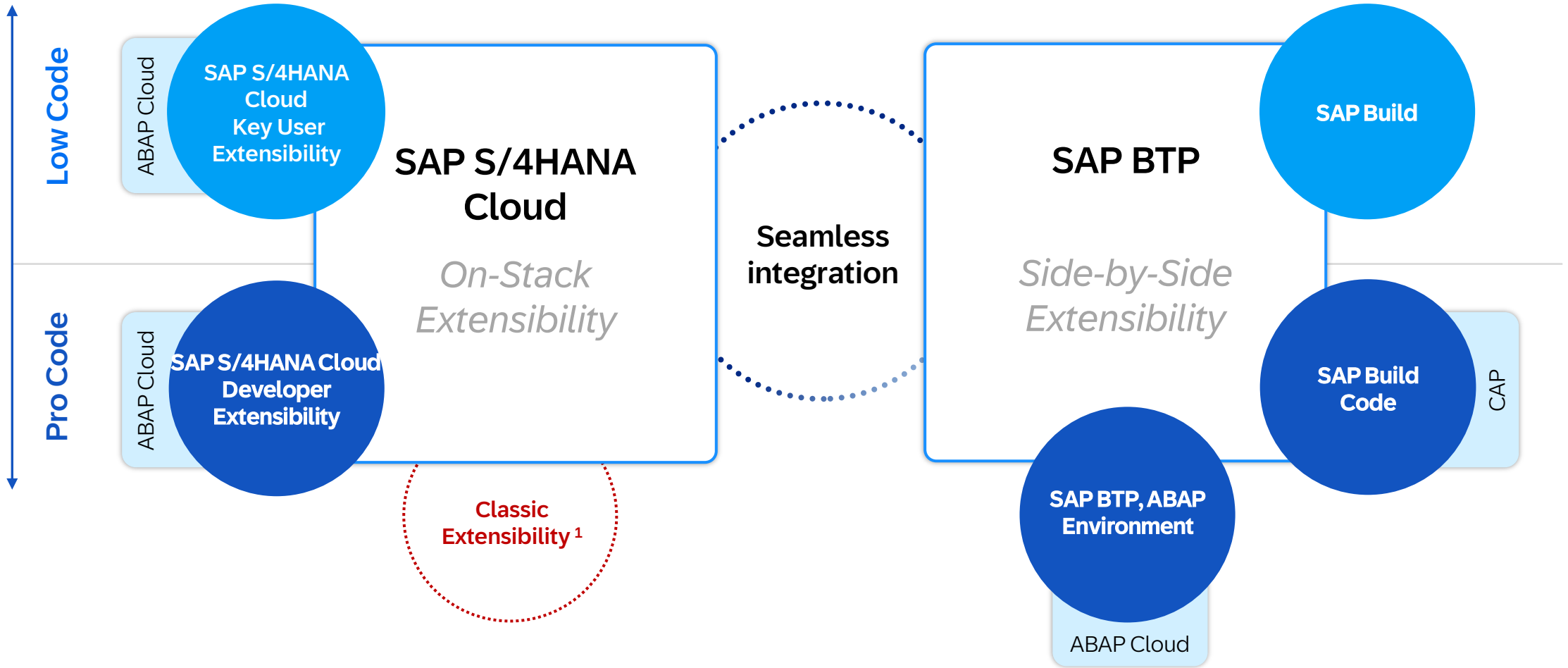


[1] SAP S/4HANA Cloud, private edition and on premise only; **not** "clean core"

Extensibility Model for SAP S/4HANA – Decoupling SAP and Custom Code is Key

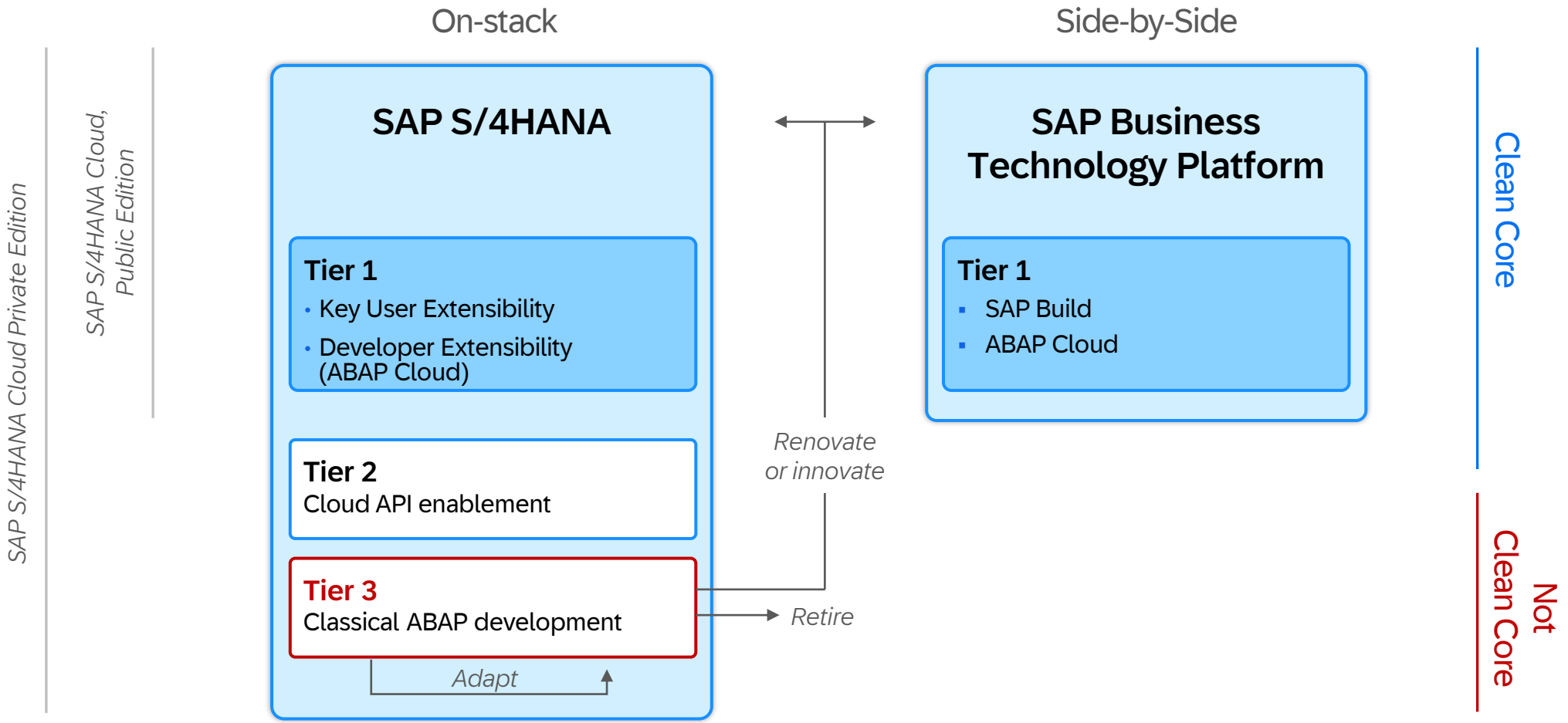


Extensibility Portfolio for SAP S/4HANA Cloud

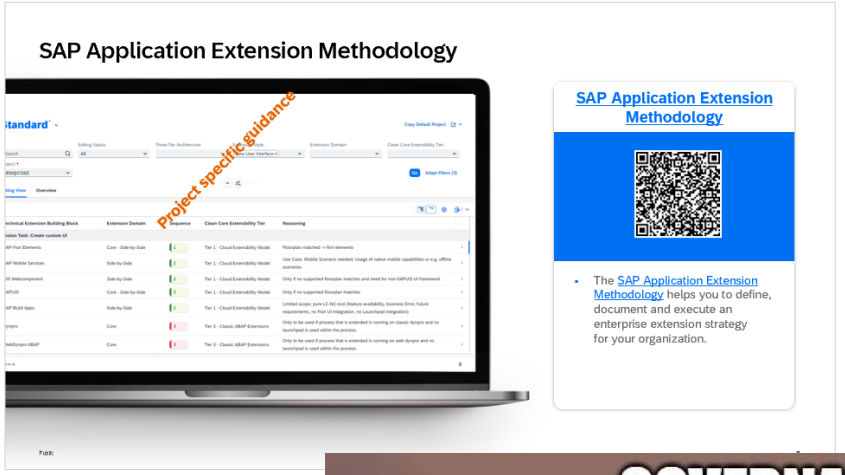


[1] SAP S/4HANA Cloud, private edition and on premise only; **not** "clean core"

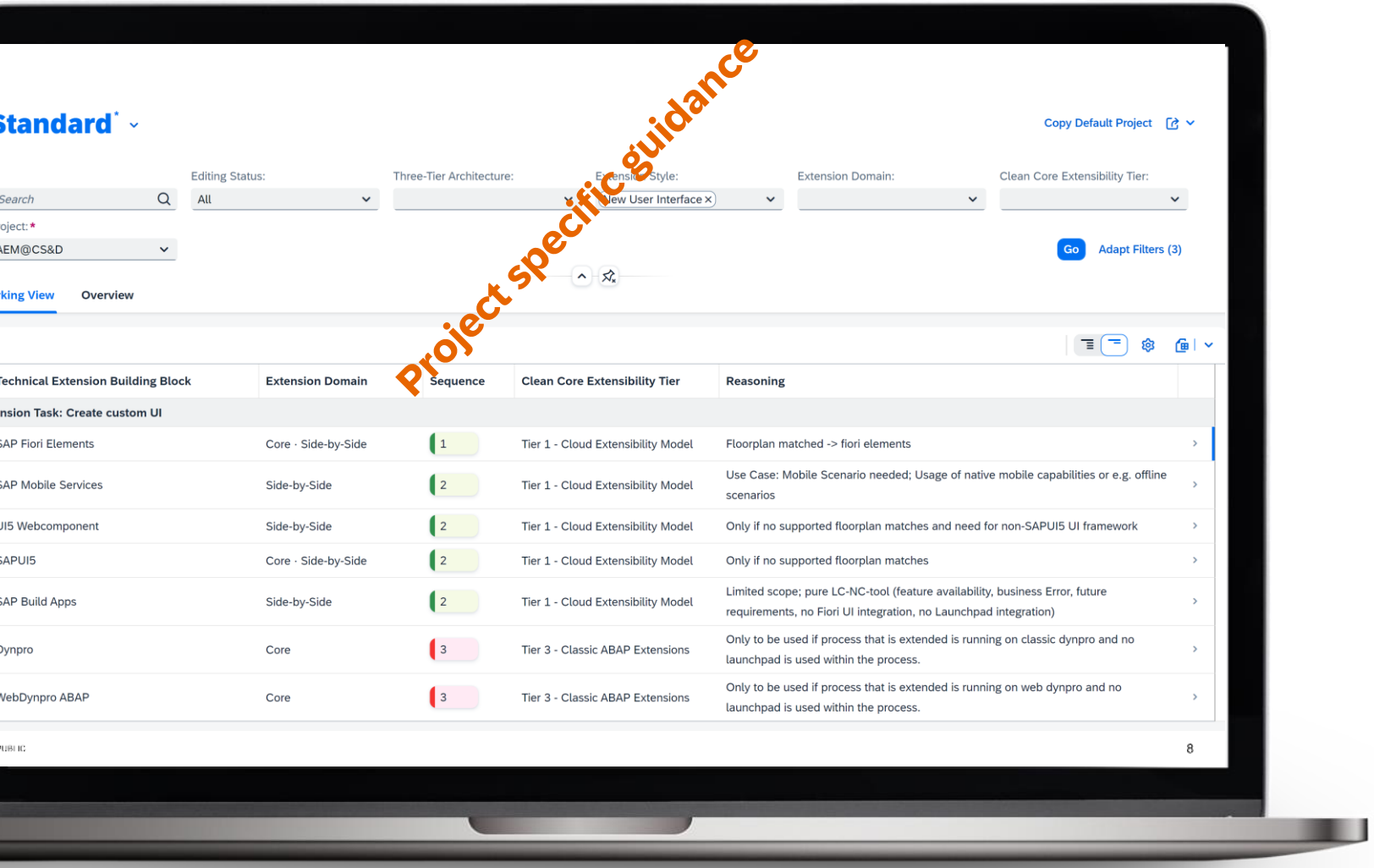
Extensibility Model for SAP S/4HANA – Decoupling SAP and Custom Code is Key



The details – governance and “when to choose what”?

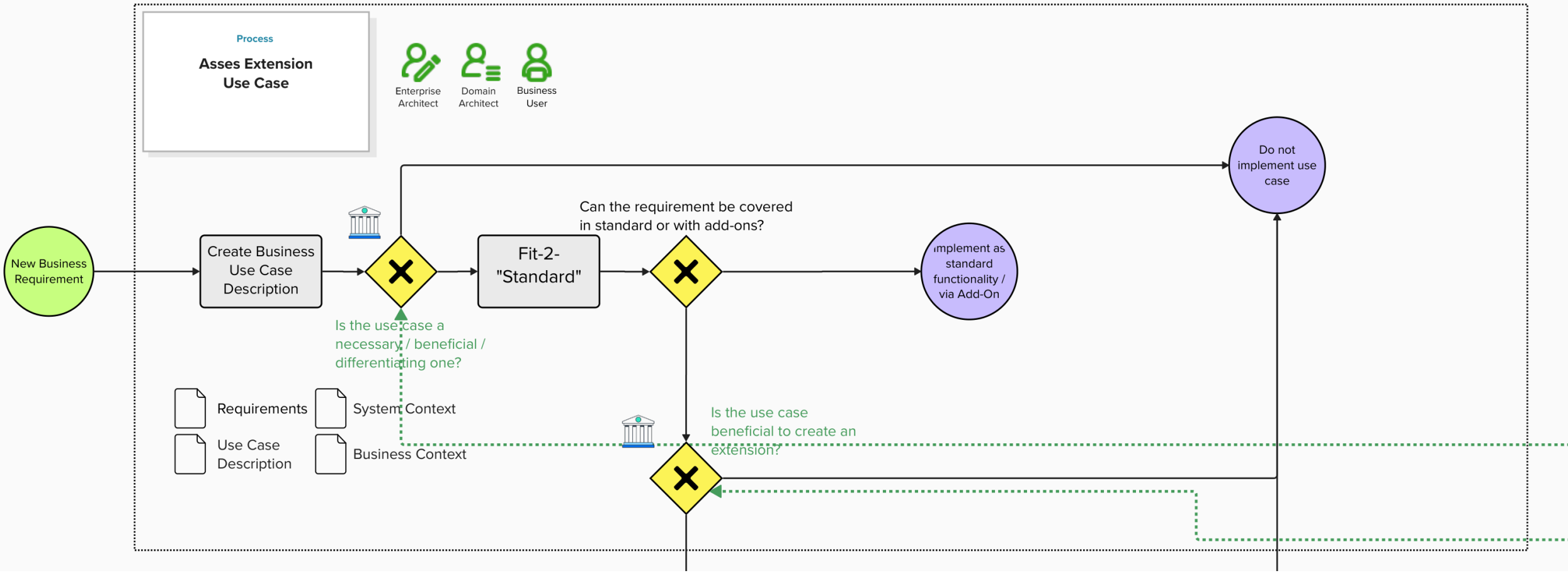


SAP Application Extension Methodology



SAP Application Extension Methodology

- The SAP Application Extension Methodology helps you to define, document and execute an enterprise extension strategy for your organization.

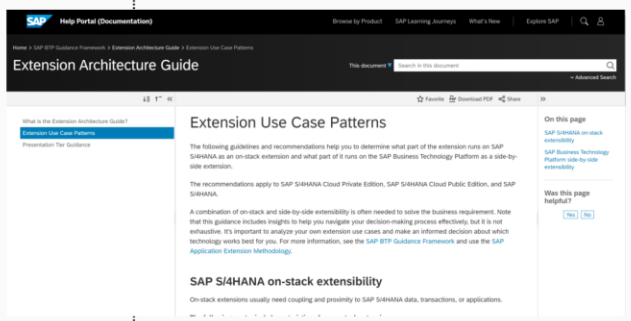

Process
Asses Extension Use Case
Enterprise Architect
Domain Architect
Business User
New Business Requirement
Create Business Use Case Description
Fit-2-"Standard"
Can the requirement be covered in standard or with add-ons?
Implement as standard functionality / via Add-On
Do not implement use case
Is the use case a necessary / beneficial / differentiating one?
Requirements
System Context
Use Case Description
Business Context
Is the use case beneficial to create an extension?

Process

Assess Extension Technology and Define Extension Target Solution

Map Requirements to Extension Tasks

Decide on Extension Domain (on-stack, side-by-side or mixed)

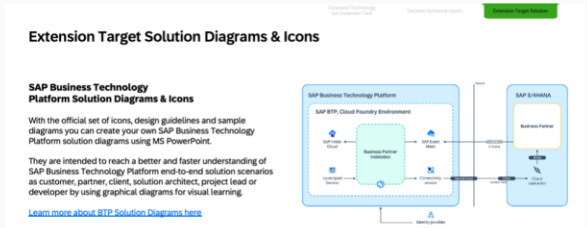


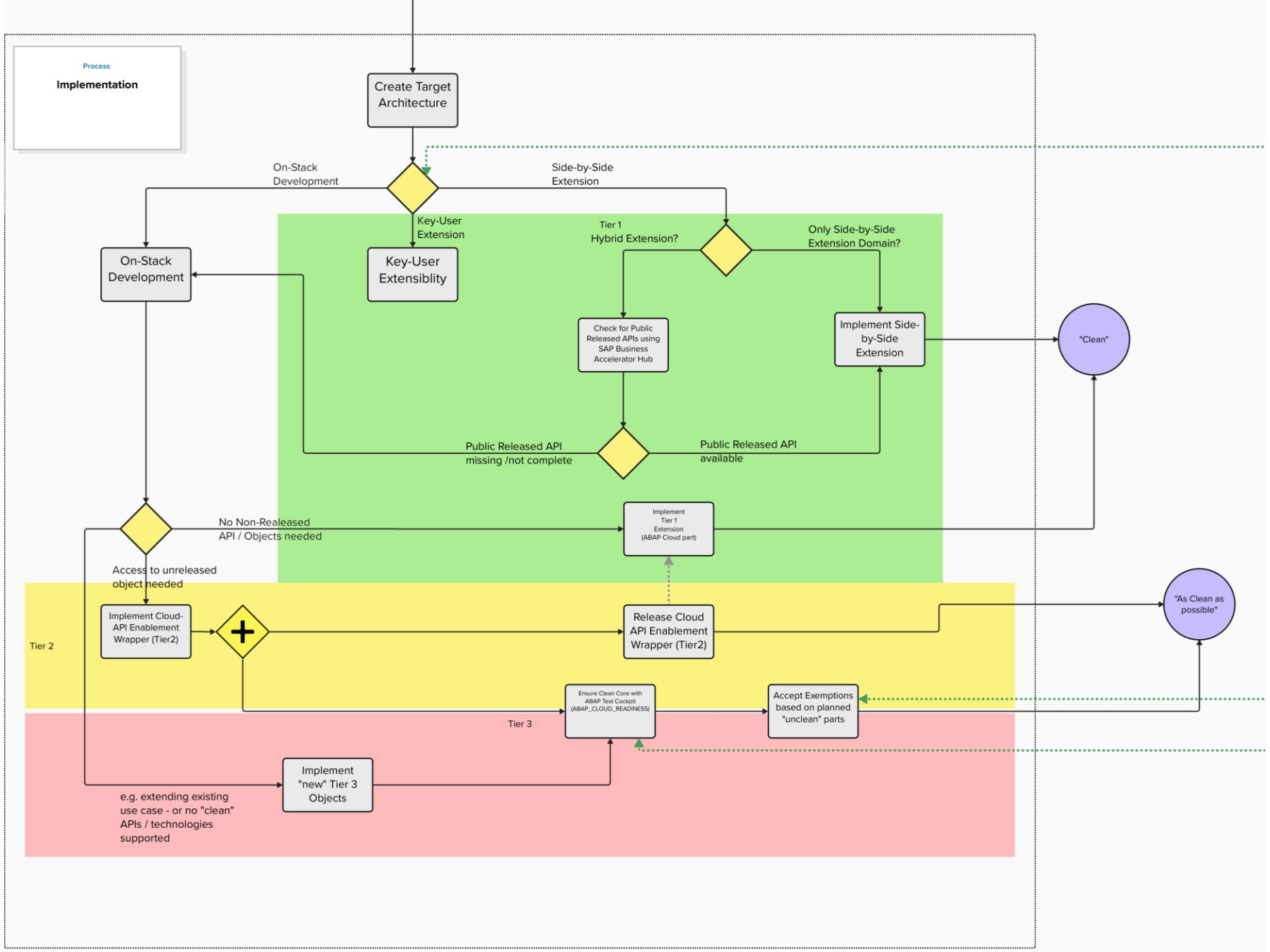
Map extension tasks to Technical Building Blocks

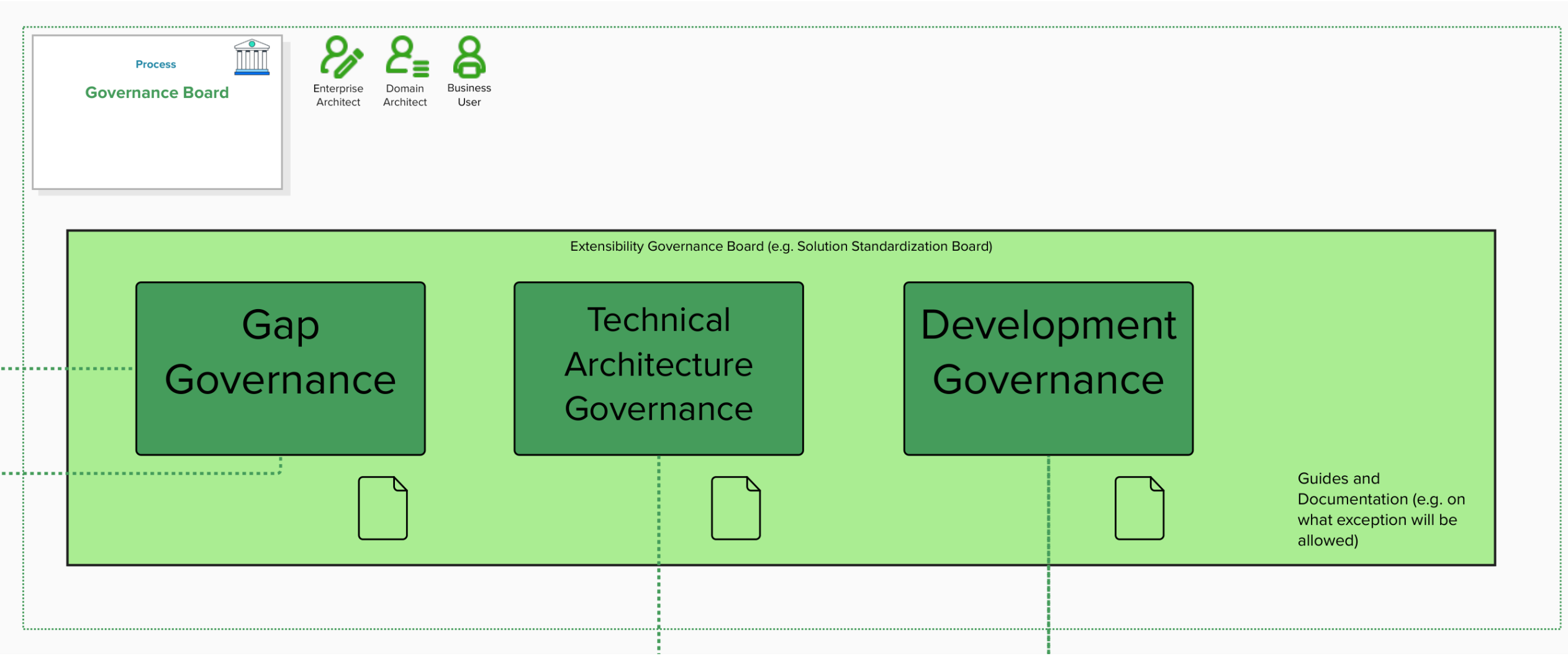
Technical Extension Building Block	Extension Domain	Sequence	Clean Core Extensibility Tier	Reasoning
Extension Task: Create custom UI				
SAP Fiori Elements	Core - Side-by-Side	1	Tier 1 - Cloud Extensibility Model	Floorplan matched -> fiori elements
SAPUI5	Core - Side-by-Side	2	Tier 1 - Cloud Extensibility Model	Only if no supported floorplan matches
UI5 Webcomponent	Side-by-Side	2	Tier 1 - Cloud Extensibility Model	Only if no supported floorplan matches and need for non-SAPUI5 UI framework
SAP Build Apps	Side-by-Side	2	Tier 1 - Cloud Extensibility Model	Limited scope: pure LC-NC-tool (feature availability, business Error, future requirements, no Fiori UI integration, no Launchpad integration)
SAP Mobile Services	Side-by-Side	3	Tier 1 - Cloud Extensibility Model	Use Case: Mobile Scenario needed: Usage of native mobile capabilities or e.g. offline scenarios
WebDynpro ABAP	Core	4	Tier 3 - Classic ABAP Extensions	Only to be used if process that is extended is running on web dynpro and no launchpad is used within the process.
Dynpro	Core	4	Tier 3 - Classic ABAP Extensions	Only to be used if process that is extended is running on classic dynpro and no launchpad is used within the process.

Is the value of the extension higher than the technical debt?

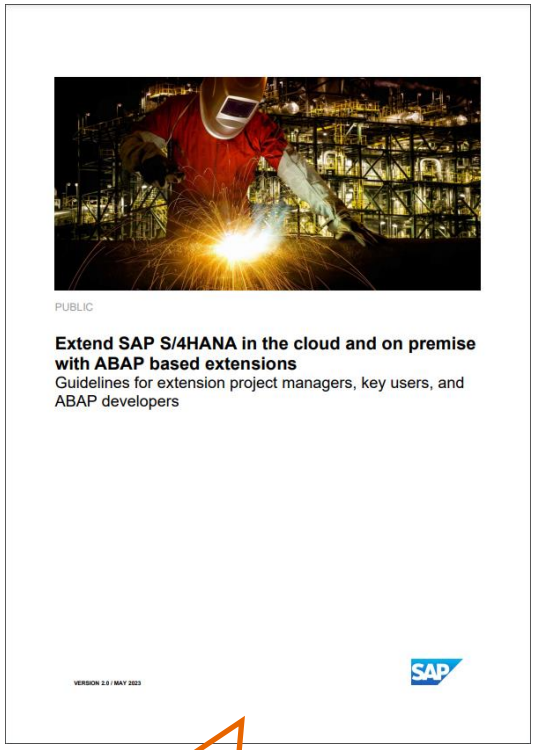
Create Target Architecture



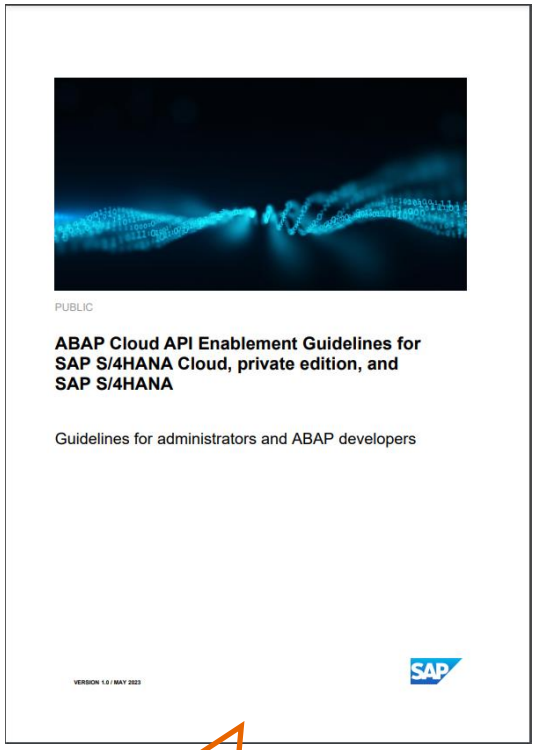

Process
Implementation
Create Target Architecture
On-Stack Development
Side-by-Side Extension
Key-User Extension
Tier 1 Hybrid Extension?
Only Side-by-Side Extension Domain?
On-Stack Development
Key-User Extensiblity
Check for Public Released APIs using SAP Business Accelerator Hub
Implement Side-by-Side Extension
"Clean"
Public Released API missing /not complete
Public Released API available
No Non-Realeased API / Objects needed
Implement Tier 1 Extension (ABAP Cloud part)
Access to unreleased object needed
Implement Cloud-API Enablement Wrapper (Tier2)
Release Cloud API Enablement Wrapper (Tier2)
"As Clean as possible"
Tier 2
Ensure Clean Core with ABAP Test Cockpit (ABAP_CLOUD_READINESS)
Accept Exemptions based on planned "unclean" parts
Tier 3
Implement "new" Tier 3 Objects
e.g. extending existing use case - or no "clean" APIs / technologies supported


Process
Governance Board
Enterprise Architect
Domain Architect
Business User
Extensibility Governance Board (e.g. Solution Standardization Board)
Gap Governance
Technical Architecture Governance
Development Governance
Guides and Documentation (e.g. on what exception will be allowed)

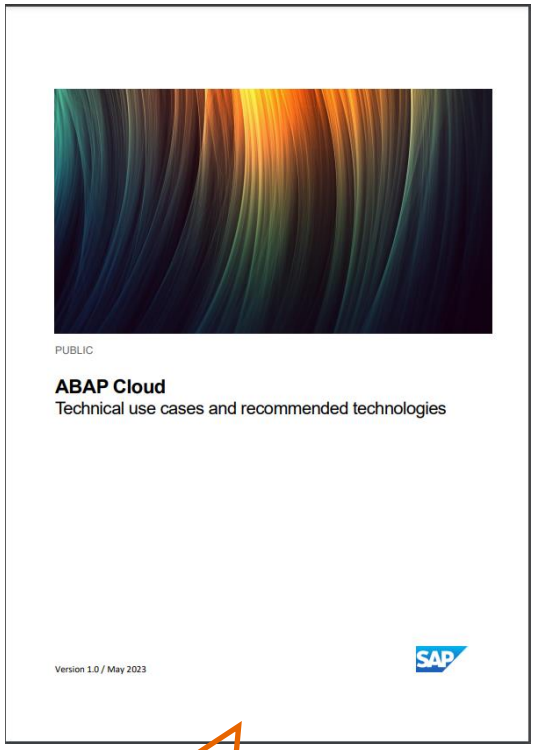
Important guides for Implementation



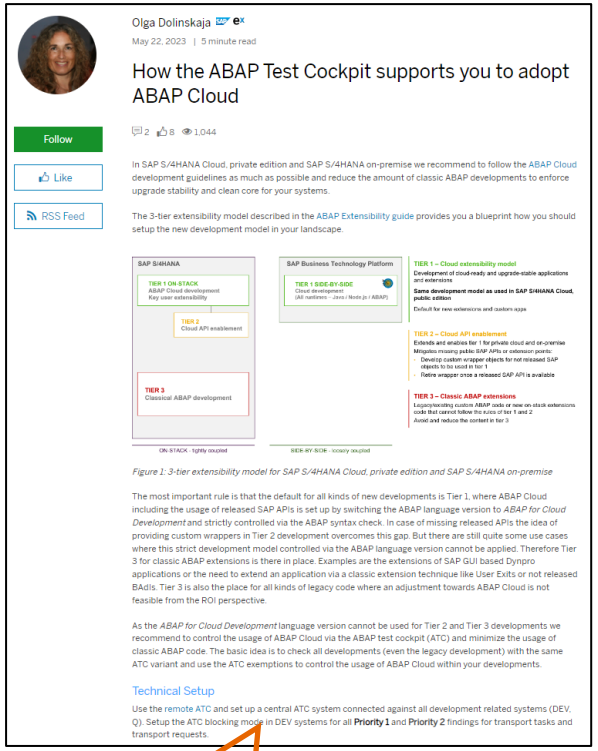
Overview, comparison of extensibility options, introduction to ABAP Cloud and 3-tier development model.



How to leverage tier 2 of the 3-tier development model



How the old, tier 3 / classic ABAP world maps to the new, tier 1 / ABAP Cloud world

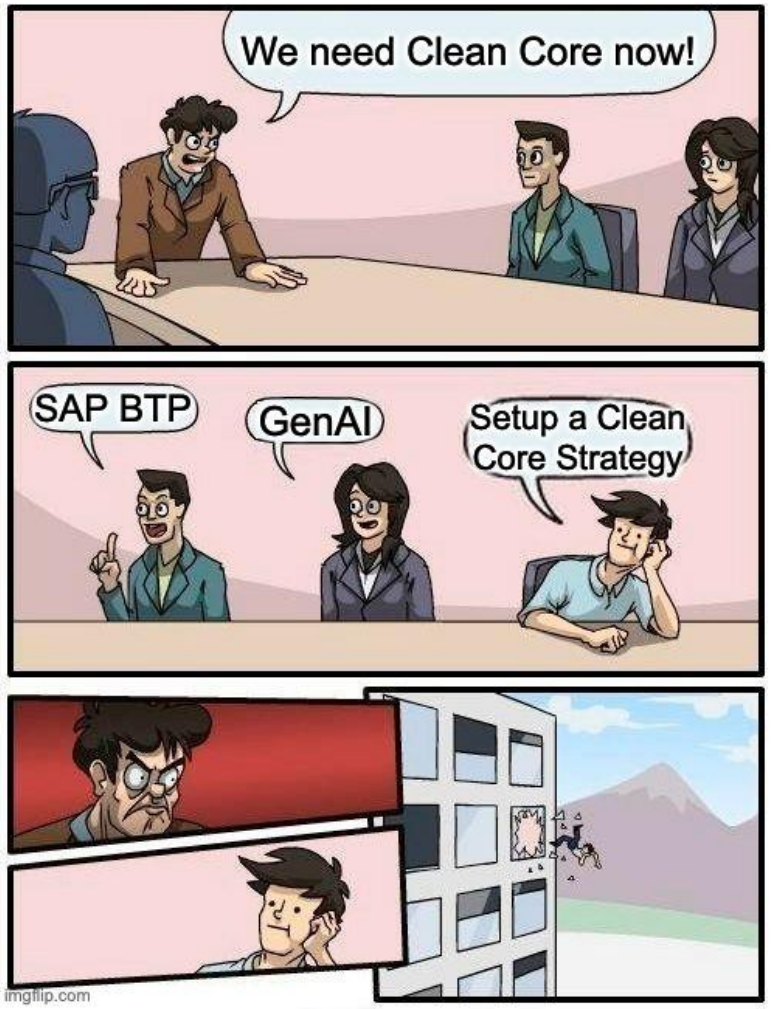


How to check compliance with the ABAP Cloud and 3-tier development model

Additional information and blog posts linked at https://community.sap.com/topics/s4hana-cloud-abap-environment

THE BAD...

The bad... expectation management



Clean Core is a journey

Establish a strategy and governance model that fosters clean core principles.

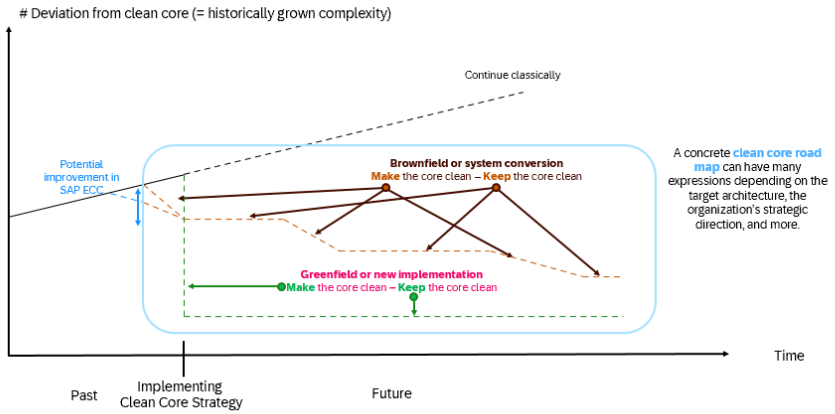
Greenfield

- Unique opportunity
- Follow Clean Core principles right from the beginning.
- Establish strong governance

Brownfield

- Establish Clean Core principles for every NEW development.
- Plan your journey towards a clean core and follow an iterative approach.
- Clean Core does **not** happen over night.

SAP S/4HANA transformations and clean core



Clean Core is a journey

Establish a strategy and governance model that fosters clean core principles.

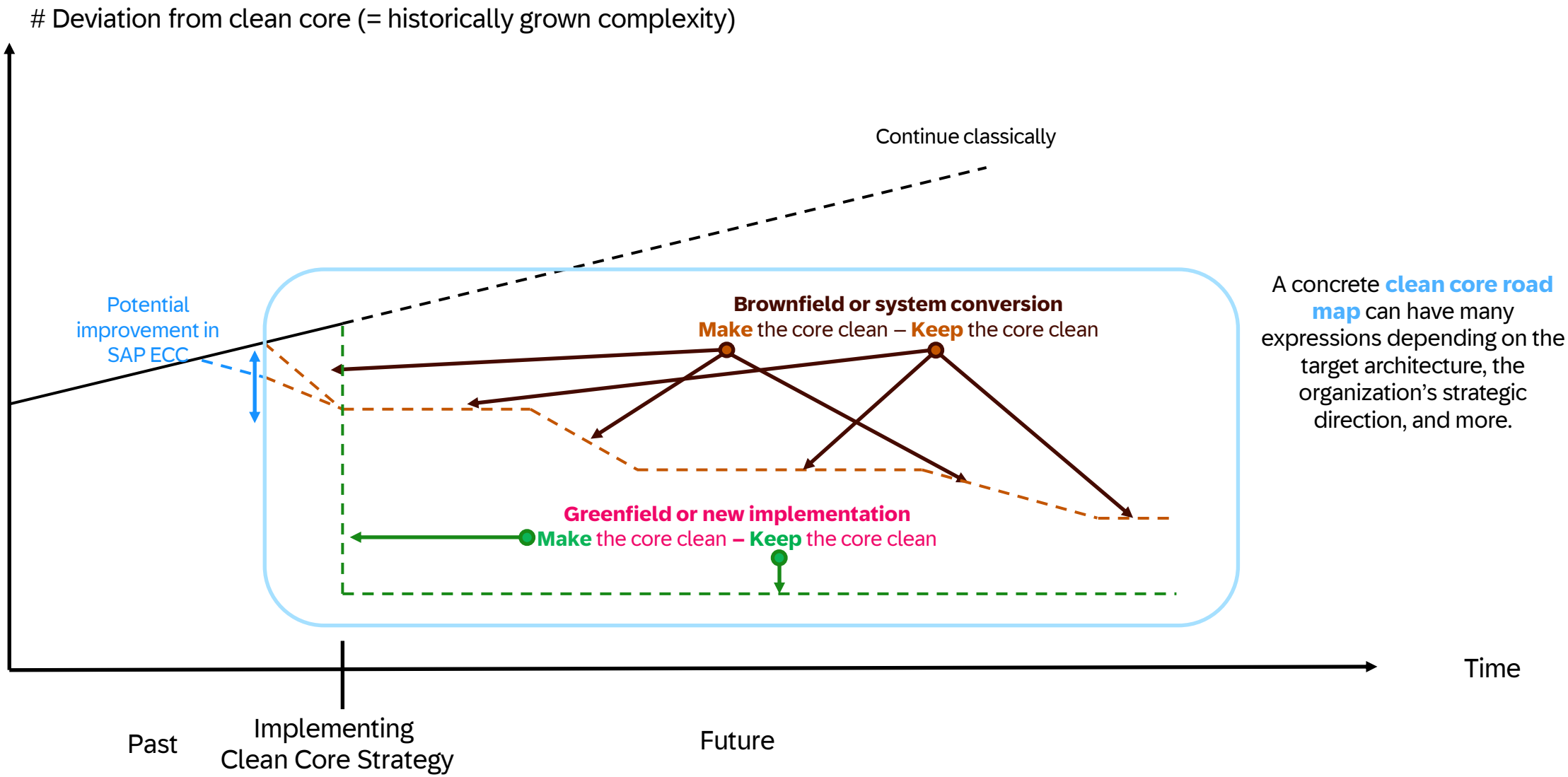
Greenfield

- Unique opportunity
- Follow Clean Core principles right from the beginning.
- Establish strong governance

Brownfield

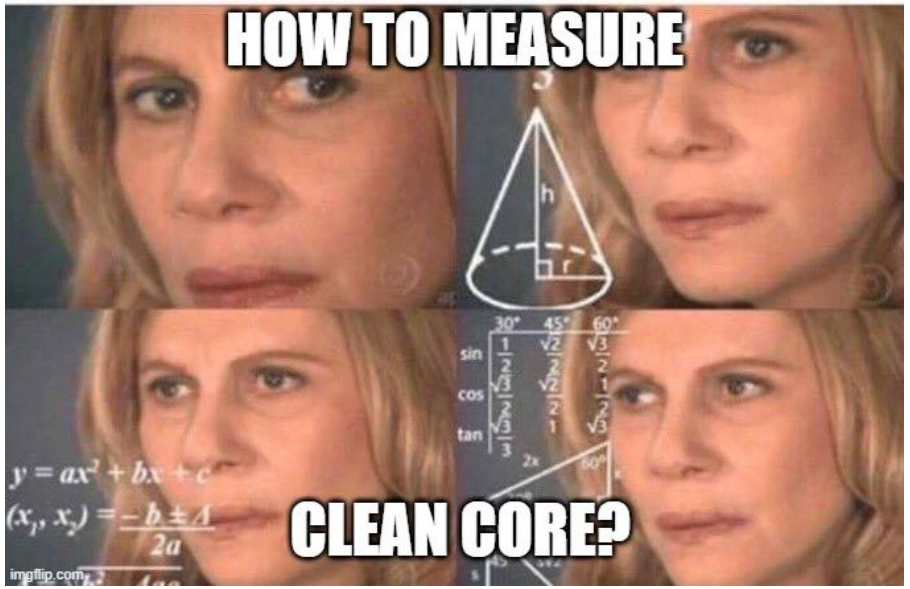
- Establish Clean Core principles for every NEW development.
- Plan your journey towards a clean core and follow an iterative approach.
- Clean Core does **not** happen over night.

SAP S/4HANA transformations and clean core

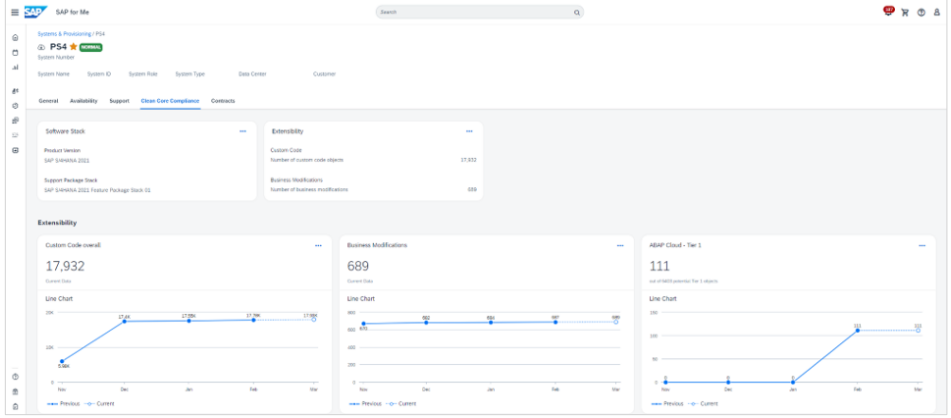


A concrete **clean core road map** can have many expressions depending on the target architecture, the organization's strategic direction, and more.

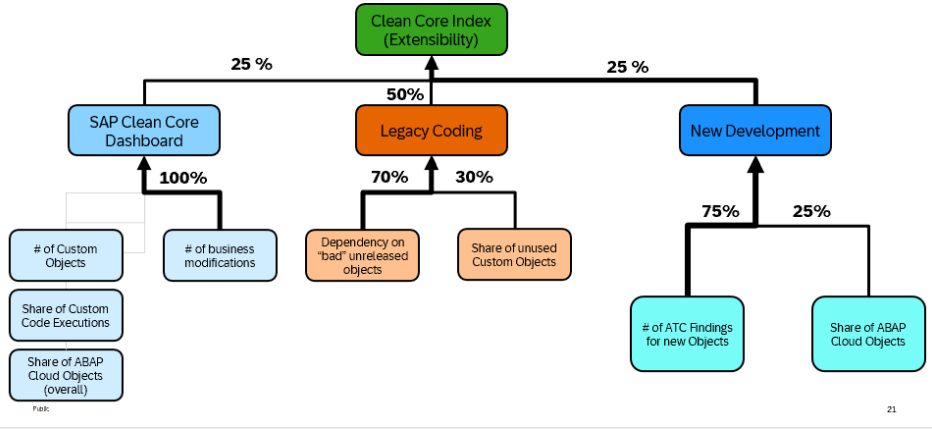
The bad... measurability



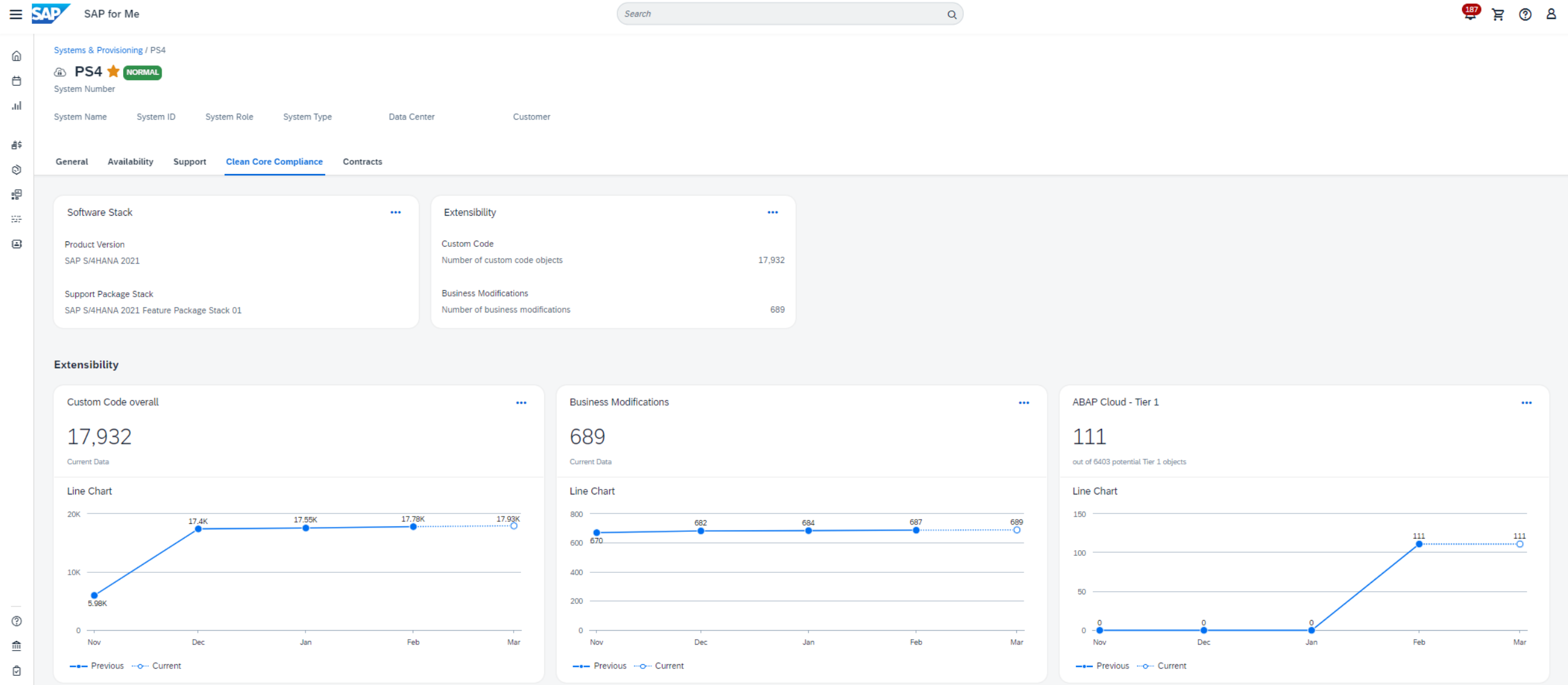
Clean Core Dashboard Example



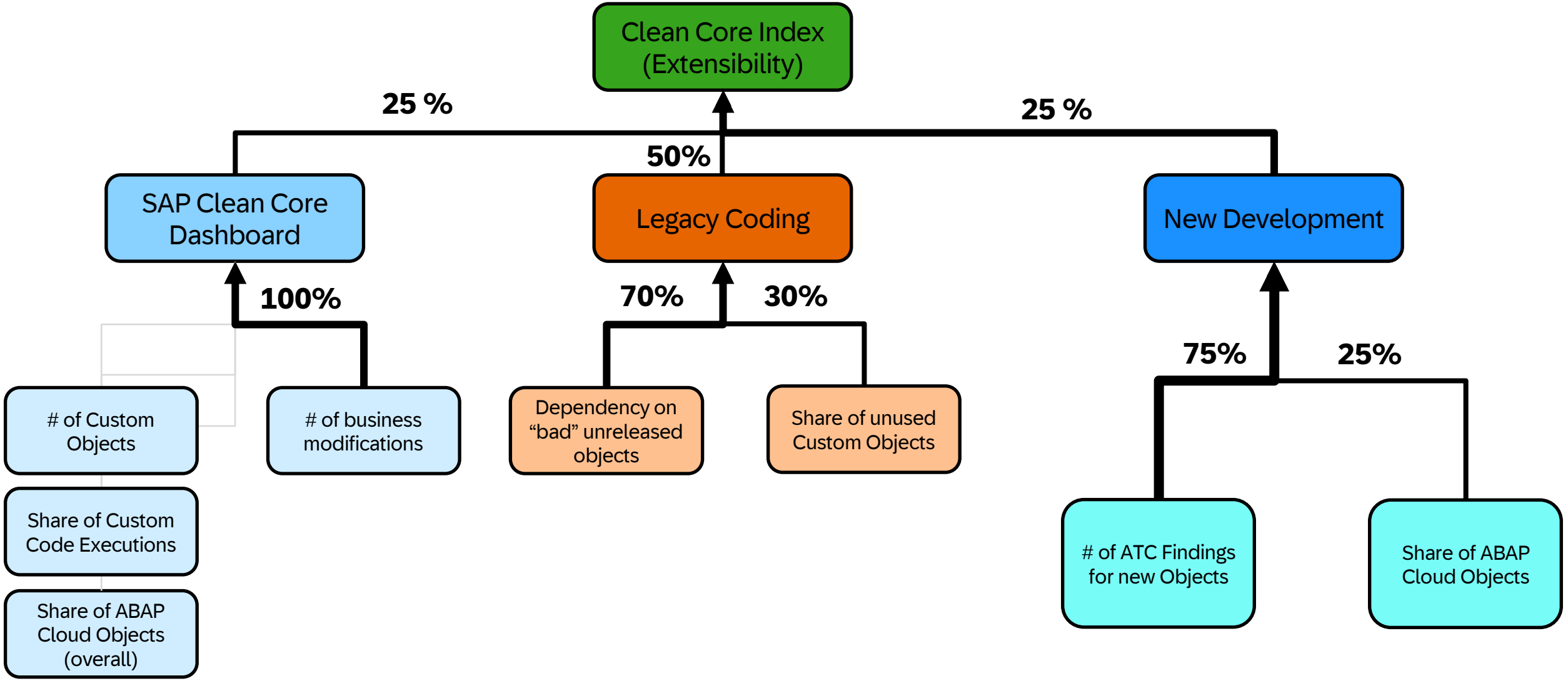
Draft of a "clean core" KPI for Extensibility



Clean Core Dashboard Example



Draft of a “clean core” KPI for Extensibility



The bad… “clean core developers”

2.2 → Decision Tree & priority of options

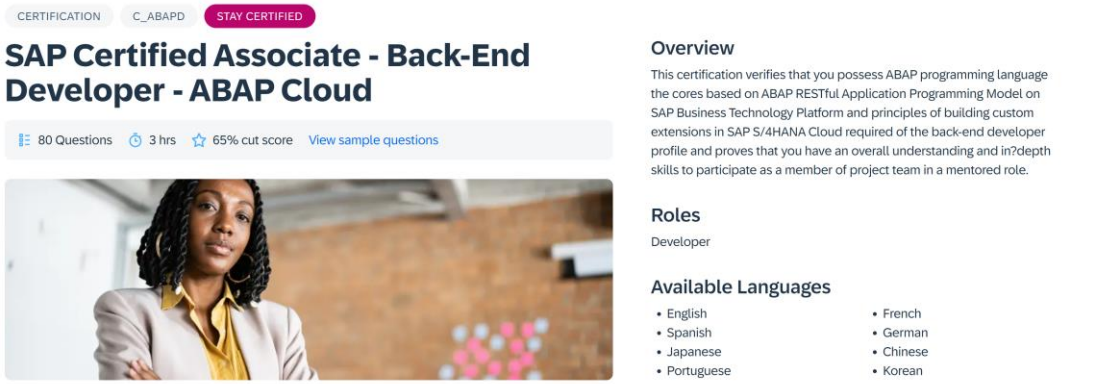
Extensibility decision tree supports in selection of the right approach and confirms exception for extensions inside the S/4 program. Prime aspect of the program is to **keep the CORE clean** by utilization of standard functionalities and avoid building custom objects as much as possible.

Sl No	Best Practices
1	Standard coding guidelines for ABAP to be followed while writing codes inside any enhancements
2	First option should be to use New BAdIs/Classic BAdIs
3	If BAdIs are not available, then only we will go with Customer Exits.
4	Implicit enhancements are the last option
5	If any standard tables are modified using APPEND or CI Includes, add the fields in the underlaying CDS views as well or follow the process to adjust the dependent structures.

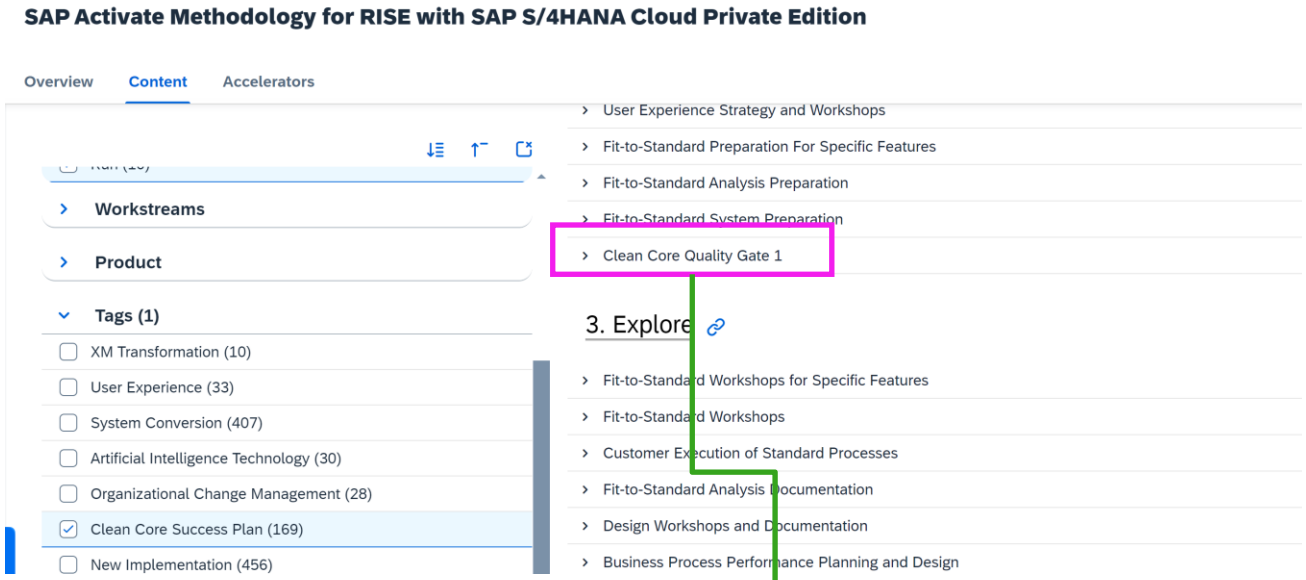
Overview after few months:

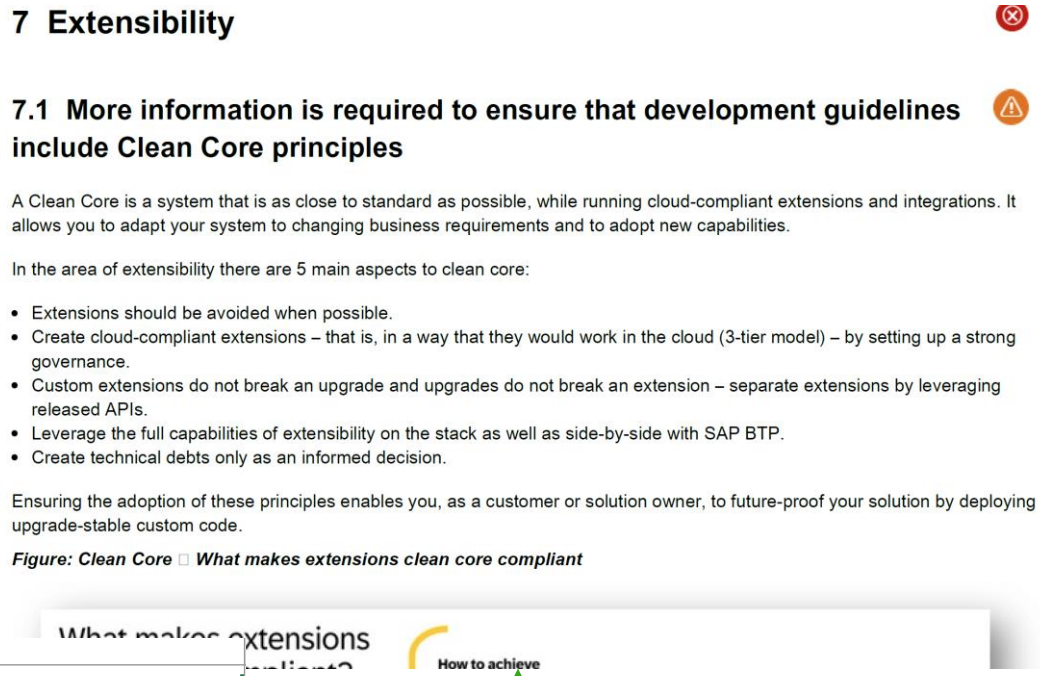
>5.000 Objects in ABAP Standard

<100 Objects in ABAP Cloud



How to ensure “clean core” in projects?





7 Extensibility

7.1 More information is required to ensure that development guidelines include Clean Core principles

A Clean Core is a system that is as close to standard as possible, while running cloud-compliant extensions and integrations. It allows you to adapt your system to changing business requirements and to adopt new capabilities.

In the area of extensibility there are 5 main aspects to clean core:

- Extensions should be avoided when possible.
- Create cloud-compliant extensions – that is, in a way that they would work in the cloud (3-tier model) – by setting up a strong governance.
- Custom extensions do not break an upgrade and upgrades do not break an extension – separate extensions by leveraging released APIs.
- Leverage the full capabilities of extensibility on the stack as well as side-by-side with SAP BTP.
- Create technical debts only as an informed decision.

Ensuring the adoption of these principles enables you, as a customer or solution owner, to future-proof your solution by deploying upgrade-stable custom code.

Figure: Clean Core □ What makes extensions clean core compliant

What makes extensions

How to achieve

Check ID	Title	Description	Answer
2-EX01	Extensibility: Does your governance process (SSB) ensure that extensions are only created if needed / valuable? (including business and technical validation)	For example, you have a process to define when an extension/modification/enhancement is business critical or when an extension/modification/enhancement is a "nice to	We are planning to setup such governance processes, but additional information would be useful
3-EX02	Extensibility: Do you plan to create development guidelines following the clean core principles?	For example, based on SAP Application Extension Methodology	We are NOT aware of the clean core principles
4-EX03	Extensibility: Are you aware of the SAP S/4HANA Extensibility framework and SAP BTP Extensibility paradigms?	For example, you are aware of the on-stack extensibility options in SAP S/4HANA and the side-by-	No, we are not aware

The bad… “APIfication process”

Explore Categories

Explore all the resources available from SAP and its partners

View All Categories

APIs

Find APIs to integrate and extend.

1592 OData APIs
1398 REST APIs
1068 SOAP APIs
20 Policy Templates
4 GraphQL APIs

Events

Explore all events in the SAP Business Accelerator Hub.

377 Event Objects

Graph

One connection to all business data.

Build extensions with a single OData v4 API exposing a consolidated data graph with thousands of connected entities supporting SAP S/4HANA, SAP SuccessFactors and SAP Sales Cloud.

Adapters

Discover prebuilt adapters and connectors from SAP and Partners for seamless integration

176 REST APIs
55 Adapters

Process Automation

Explore all live process content packages from SAP Build Process Automation and SAP Workflow Management.

100 Decisions
81 Visibility Scenarios
59 Processes
6 Business Rules

Business Add-Ins (BAdIs)

Explore Business Add-Ins (BAdIs) that can be used for developer extensibility.

903 Business Add-Ins

Business Object Interfaces

Explore business object interfaces that can be used for developer extensibility.

320 Business Object Interfaces

CDS Views

Explore CDS view content from SAP.

13835 CDS Views

SAP S/4HANA Cloud Private Edition for Extensibility and Integration (APIs)

API for confirming transfer order

Request ID: 322879, Category: Digital Supply Chain (TM, EWM)

Vote: Vote
Follow:

Status: **Not Planned**
Decision: Product strategy - will not be developed

Submitted on: May 15, 2024
Changed on: May 27, 2024

Author: Rene Hoelterling
Co-Author: Dennis Rohde
Coach: Tymur Raisov

DETAILS | ATTACHMENTS | COMMENTS | VOTES (2) | RELATED REQUESTS | PEOPLE | ACTIVITY LOG

Description:

What is the opportunity/problem the request will address?

Dear experts,

we need an odata-api for confirming transfer order analog to SAP Standard transaction LT11/12.

in SAP R3 the L_TO_CONFIRM function module solves the case.

What is the expected benefit?

Decision Details

Decision Maker:
Tymur Raisov
Decision Date:
2024-05-27
Decision Reason:
Product strategy - will not be developed
Comment (limited to 500 chars):
Hello

The bad… but how to avoid?

Manage the expectation between top management and developers but also between realistic goals and budget!

Based on this expectation, identify ways to measure the improvement!

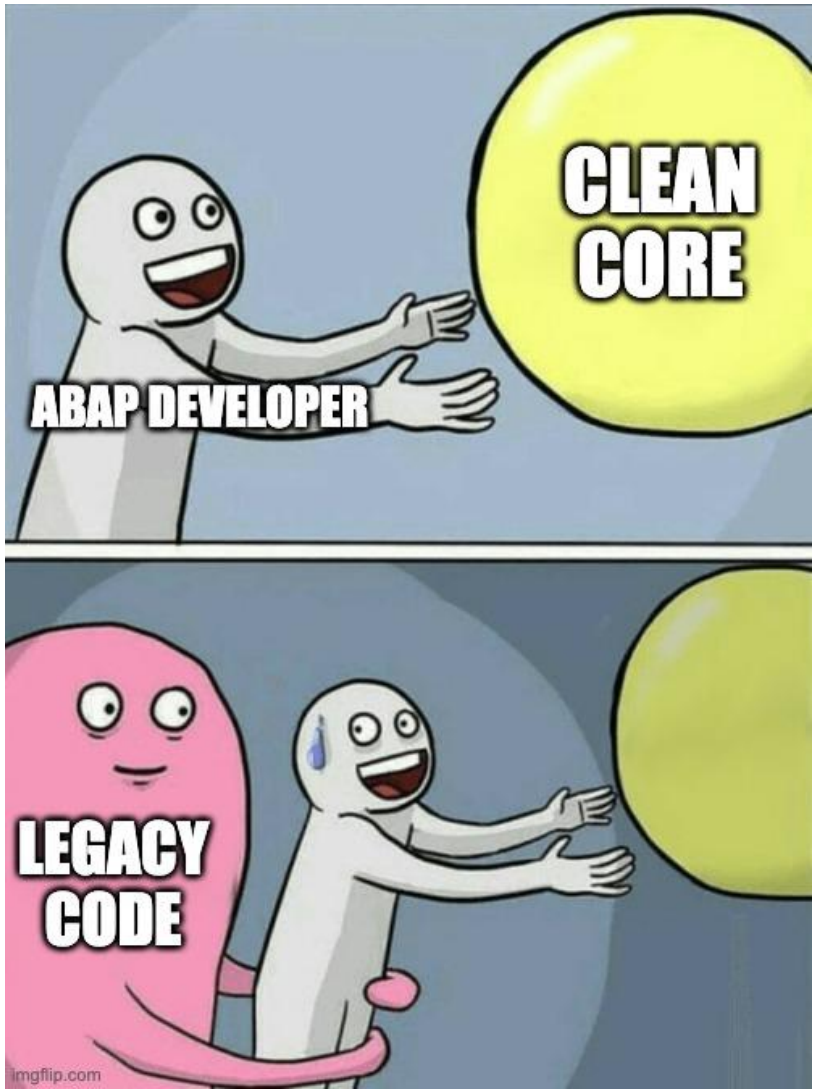
Ensure change management with involved architects and developers to ensure awareness for the “new way” of extensibility!

Leverage tools and templates available to live “efficient” processes for clean core development!

Identify the APIs you need – and mitigate and request if APIs are missing!

THE UGLY...

The ugly... legacy code!

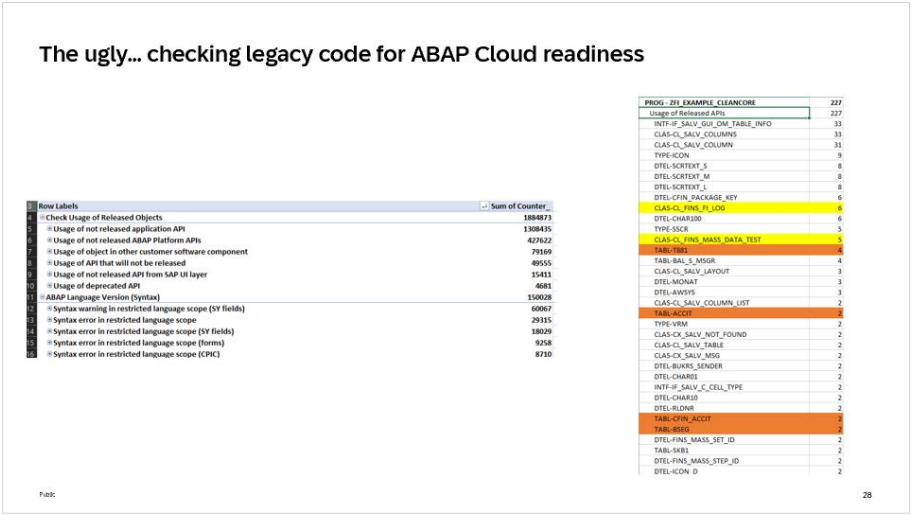


Which object is “expensive” when it comes to upgrades?

- Development efforts?
- Testing Efforts?
- Bugs and Tickets?

...collect any data available on this create transparency

Code analysis can be a valuable source, but not the only source of information!

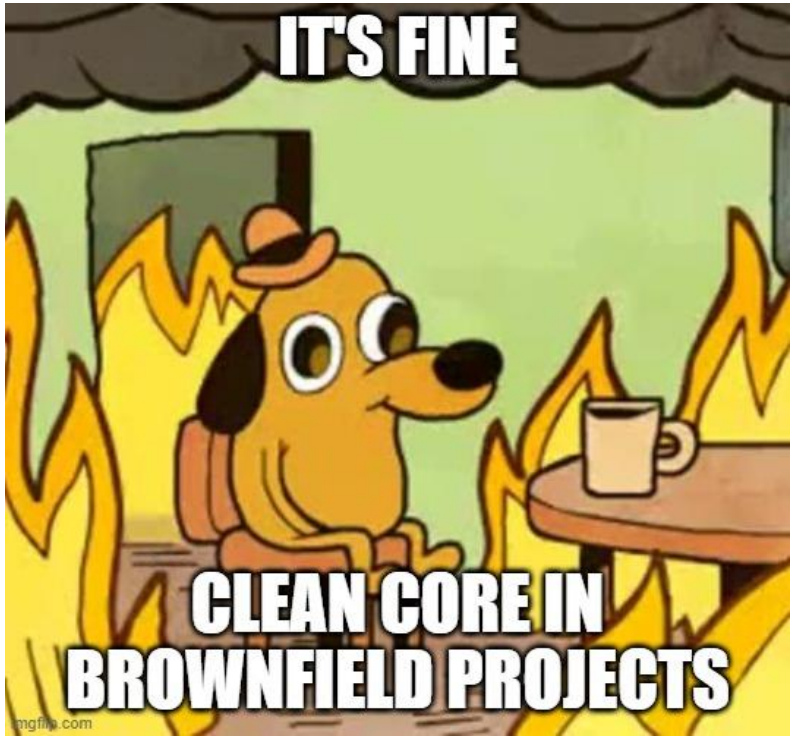


The ugly... checking legacy code for ABAP Cloud readiness

3	Row Labels	Sum of Counter_
4	⊟Check Usage of Released Objects	1884873
5	⊞Usage of not released application API	1308435
6	⊞Usage of not released ABAP Platform APIs	427622
7	⊞Usage of object in other customer software component	79169
8	⊞Usage of API that will not be released	49555
9	⊞Usage of not released API from SAP UI layer	15411
10	⊞Usage of deprecated API	4681
11	⊟ABAP Language Version (Syntax)	150028
12	⊞Syntax warning in restricted language scope (SY fields)	60067
13	⊞Syntax error in restricted language scope	29315
14	⊞Syntax error in restricted language scope (SY fields)	18029
15	⊞Syntax error in restricted language scope (forms)	9258
16	⊞Syntax error in restricted language scope (CPIC)	8710

PROG - ZFI_EXAMPLE_CLEANCORE	227
Usage of Released APIs	227
INTF-IF_SALV_GUI_OM_TABLE_INFO	33
CLAS-CL_SALV_COLUMNS	33
CLAS-CL_SALV_COLUMN	31
TYPE-ICON	9
DTEL-SCRTEXT_S	8
DTEL-SCRTEXT_M	8
DTEL-SCRTEXT_L	8
DTEL-CFIN_PACKAGE_KEY	6
CLAS-CL_FINS_FI_LOG	6
DTEL-CHAR100	6
TYPE-SSCR	5
CLAS-CL_FINS_MASS_DATA_TEST	5
TABL-T881	4
TABL-BAL_S_MSGR	4
CLAS-CL_SALV_LAYOUT	3
DTEL-MONAT	3
DTEL-AWSYS	3
CLAS-CL_SALV_COLUMN_LIST	2
TABL-ACCIT	2
TYPE-VRM	2
CLAS-CX_SALV_NOT_FOUND	2
CLAS-CL_SALV_TABLE	2
CLAS-CX_SALV_MSG	2
DTEL-BUKRS_SENDER	2
DTEL-CHAR01	2
INTF-IF_SALV_C_CELL_TYPE	2
DTEL-CHAR10	2
DTEL-RLDNR	2
TABL-CFIN_ACCIT	2
TABL-BSEG	2
DTEL-FINS_MASS_SET_ID	2
TABL-SKB1	2
DTEL-FINS_MASS_STEP_ID	2
DTEL-ICON_D	2

How to handle legacy code?



Decide on your own ”Clean Core” goals (e.g. Upgrade stability/ Cloud Readiness / Fiori coverage) and prioritize

- Clean legacy code when you have to touch it anyway
- Define “lighthouse” projects to create transparency
- Measure the “right” KPIs for your goal!

SUMMARY

Takeaways

There is no reason NOT to aim for the cleanest development possible*

Do not fall into “death by planning” antipattern – but “hope is not a strategy” as well!

Challenge your project team (architects, developers, business experts, management) if clean core was considered reasonably

There is a lot of useful stuff available – leverage it!

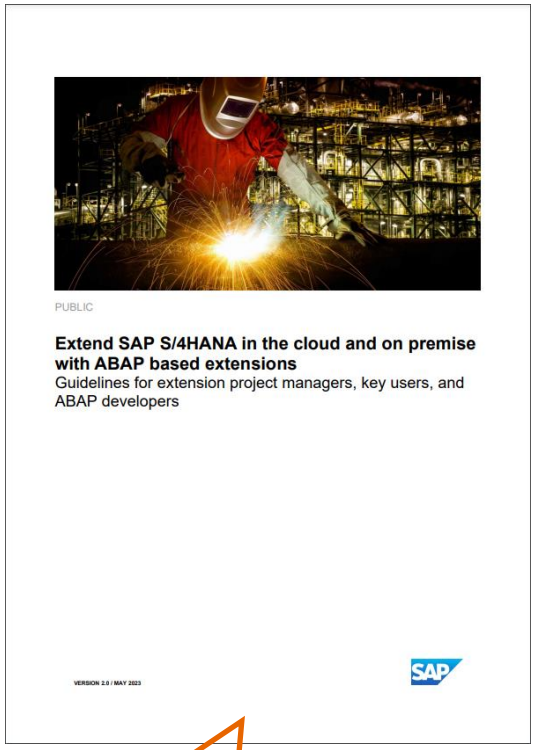
* especially as of SAP S/4HANA >2022

Thank you.

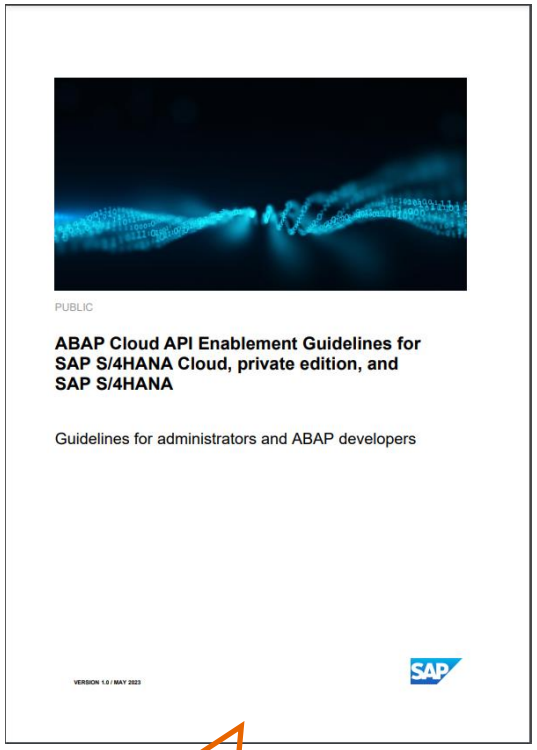
Contact information:

Lukas Bretschneider & Lukas Käser
lukas.bretschneider@sap.com / lukas.kaeser@sap.com

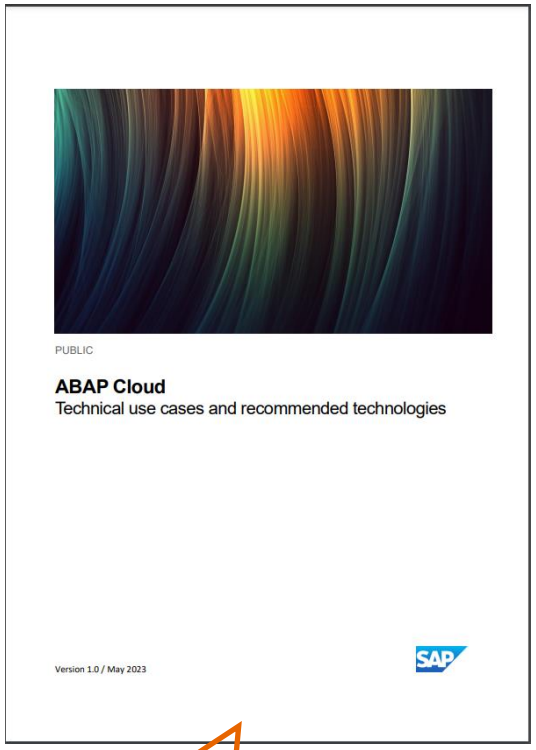
Important guides



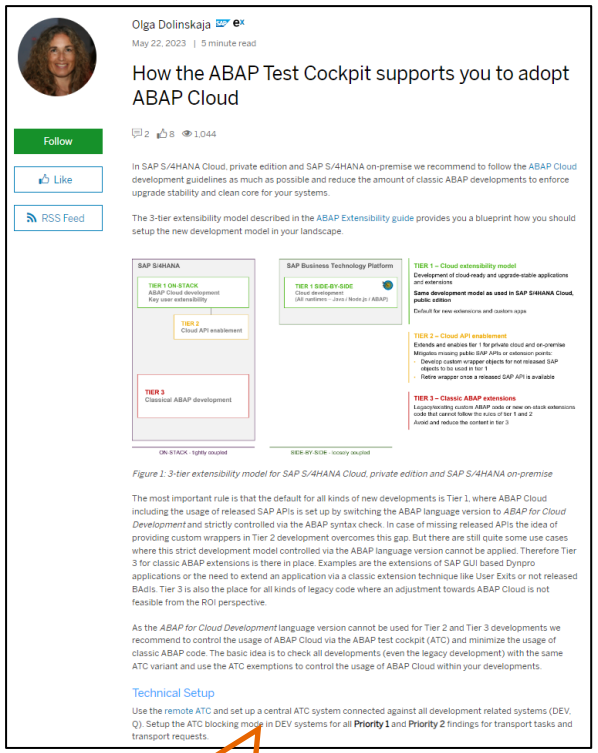
Overview, comparison of extensibility options, introduction to ABAP Cloud and 3-tier development model.



How to leverage tier 2 of the 3-tier development model



How the old, tier 3 / classic ABAP world maps to the new, tier 1 / ABAP Cloud world



How to check compliance with the ABAP Cloud and 3-tier development model

Additional information and blog posts linked at https://community.sap.com/topics/s4hana-cloud-abap-environment