

# Development ID Definition and Counting Rules

This document defines the use of Development IDs, how they are managed and counted in conjunction with SAP's Clean Core principles.

The Development ID is a centerpiece to manage, track and measure custom development requirements. Development IDs are lower level than WRICEFs to define and document the needed artefacts. Clean Core principles need to be applied to each Development ID object based on feasibility to keep the Total Cost of Ownership for custom developments manageable and explicit.

- [Development ID Definition](#)
  - [Examples](#)
- [SAP 3 Tier classification in Clean Core](#)
  - [SAP 3 Tier Technology Mapping](#)
- [Non-SAP SaaS Customization Tier classification](#)
  - [Salesforce](#)
  - [Icertis](#)
- [Clean Core Deviation tracking](#)
- [Development ID Counting Rules](#)

## Development ID Definition

"WRICEF" was the traditional way of categorizing business requirements that will be implemented with custom development, where each of the letters in the acronym stands for Workflow, Report, Interface, Conversion, Enhancement, and Form, respectively. Typically the letter of the acronym is used in combination with a running number. But in the several decades since this acronym entered into use, SAP development has evolved significantly. There are now many more types of developments which cannot meaningfully be combined into an "Enhancement" and treated equivalently; similarly it is sometimes not easy to assign a meaningful category to a development.

Instead of the traditional WRICEF approach, we will use Development IDs in combination with their Type.

The "Type of Development" will be introduced to categorize a development instead of the WRICEF. It is a more detailed classification that more accurately reflects modern SAP development types:

- Enhancement (incl. Conversion)
- Form (Output)
- HANA CDS View
- Integration Process (custom)
- Integration Process (standard)
- Metrics/KPI
- Mobile App
- Modification
- Program
- Report/Analytics
- System Interface
- User Interface
- Workflow (custom)
- Workflow (standard)
- Application Job (Custom)

The Development ID is the number assigned to developments. The Development ID will be created after the request has been approved. The [Development Request How-To](#) details the process in Jira.

The Development ID will be used in the functional design (FD), the development package name and needs to be assigned in all relevant L5 business process steps in Signavio.

Every development requires at least one Development ID. SAP Notes of type Modification and Consulting require an approved Development ID and a basic Functional Design document as well. Although SAP might provide the code, SAP does not support Consulting Notes or Modification Notes as part of its product support scope. Since the ongoing support and upgrade maintenance are at Syensqo's risk and cost, they are also considered custom developments from a governance perspective.

Conversions are not required to have a Development ID. The Master Data Register number will be used. A Development ID for conversions is only needed if a custom object, such as a load program, needs to be created.

## Examples

For example, a Development Request ID 1234 of type User Interface has been approved. The UI development requires a UI5 app and an oData service. That means 2 Development IDs will be created.

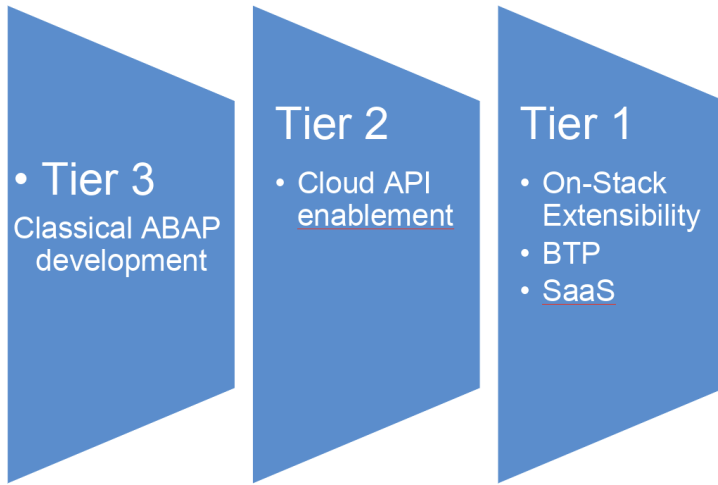
Sample: Jira/FD/Signavio : 1235 (Type User Interface) and 1236 (Type System Interface)

Sample: Package: Z[POD]\_1235 and Z[POD]\_1236

Sample: SaaS: 1237 (Type System Interface)

# SAP 3 Tier classification in Clean Core

Clean Core is SAP's concept to achieve modern, flexible and cloud compliant custom solutions. SAP came up with 3 tiers of custom developments:



Tier 3: Classical ABAP development in S/4HANA Core (deprecated legacy code/tools with exceptions where still required).

Tier 2: Cloud API enablement (custom wrapper to use unreleased APIs in Tier 1) or mandatory use of legacy tools, i.e. pricing routines, user exits, in S4HANA Core systems.

Tier 1: On-Stack (Key User or Developer Extensibility) or in BTP or SAP SaaS customizations.

ABAP Cloud for S/4HANA Core is the programming model for cloud compliant custom developments. Implementing with ABAP Cloud ensures that the code will be considered as compliant with Clean Core and is upgrade stable.

SAP BTP is the platform for SAP SaaS extensions or cross-system processes.

## SAP 3 Tier Technology Mapping

Based on the above definition of clean core the following table contains the main development objects and approaches per type of development.

Type of Development	Tier 1	Tier 2	Tier 3
Conversion (Load only)	Released standard APIs called from Syniti	<ul style="list-style-type: none"> <li>• Custom ABAP Load program*</li> <li>• Custom ABAP class or function*</li> </ul>	N/A
Conversion (Transform + Load)	N/A	<ul style="list-style-type: none"> <li>• Custom ABAP Load program*</li> <li>• Custom ABAP class or function*</li> </ul>	N/A
Enhancement	<ul style="list-style-type: none"> <li>• Released Badi implementation</li> <li>• Custom structures or fields included into clean core compliant / EEW append structures</li> <li>• ABAP Cloud compliant custom tables</li> <li>• Custom ABAP Cloud compliant code</li> </ul>	<ul style="list-style-type: none"> <li>• Custom Tier 2 wrapper</li> <li>• Unreleased Badi implementation*</li> <li>• Customer exit*</li> <li>• Routine exit*</li> <li>• Custom structures or fields included into CI append structures*</li> </ul>	<ul style="list-style-type: none"> <li>• Custom ABAP Program with user access</li> <li>• Custom Classic ABAP for use in background jobs</li> <li>• Custom structures or fields directly added to the main table</li> <li>• Unreleased custom tables</li> </ul>

Form (Output)	<ul style="list-style-type: none"> <li>• Custom Adobe Form with Fragments using a standard oData service</li> <li>• Custom Adobe Form with Fragments using a custom oData service</li> </ul>	<ul style="list-style-type: none"> <li>• Custom Classic Adobe Form*</li> <li>• Custom Adobe Form with Fragments using a custom Gateway service*</li> </ul>	<ul style="list-style-type: none"> <li>• Custom SAP Script</li> <li>• Custom Smartform</li> </ul>
HANA CDS View	Custom Released CDS entity	N/A	Custom CDS view
Integration Process (custom)	<ul style="list-style-type: none"> <li>• Custom released interfaces connecting via middleware</li> <li>• Custom iFlows</li> </ul>		N/A
Integration Process (standard)	<ul style="list-style-type: none"> <li>• Standard iFlows</li> </ul>	N/A	N/A
Mobile App	Custom Neptune app (on Open Edition)	Custom Neptune app (on SAP Edition)	N/A
Modification	N/A	N/A	<ul style="list-style-type: none"> <li>• Standard code modification</li> <li>• Implicit Enhancement</li> </ul>
Program	N/A	Custom Classic ABAP program to run as part of CLOCO or custom print program for Adobe Forms*	Custom Non CLOCO / print program
Report/Analytics	<ul style="list-style-type: none"> <li>• Custom Fiori Elements Report</li> <li>• Custom SAC Report or Dashboard (with Datasphere integration)</li> <li>• Custom SAP Analytics Cloud, add-in for Microsoft Excel Report</li> </ul>	<ul style="list-style-type: none"> <li>• Custom KPI report (Manage KPIs and Reports app)</li> <li>• Custom AFO Report*</li> <li>• Custom Analytical Query app based views</li> </ul>	<ul style="list-style-type: none"> <li>• Custom ABAP Report with custom transaction code</li> <li>• Custom Business Objects Report (Lumira)</li> </ul>
System Interface	<ul style="list-style-type: none"> <li>• Released APIs</li> <li>• Standard RFCs</li> <li>• Custom RAP based oData or REST API</li> <li>• Standard integration proxies and services in S/4HANA</li> <li>• Custom BTP CAP based services</li> </ul>	<ul style="list-style-type: none"> <li>• Custom Gateway service*</li> <li>• Custom web service proxy object in S/4HANA</li> </ul>	<ul style="list-style-type: none"> <li>• Custom ABAP Program with direct service calls</li> <li>• Custom RFC service</li> </ul>
User Interface	<ul style="list-style-type: none"> <li>• Custom Fiori UX</li> <li>• Custom Web App on BTP (React or other non SAP UI libraries)</li> <li>• UI Adaptation project</li> <li>• Key User extensibility (i.e. UI Adaption, Custom Fields and Logic app)</li> <li>• Screen Personas for GUI simplification</li> </ul>	<ul style="list-style-type: none"> <li>• Standard Dynpro enhancement</li> <li>• Standard Web Dynpro enhancement</li> </ul>	<ul style="list-style-type: none"> <li>• Custom ABAP Dynpro</li> <li>• Custom Web Dynpro</li> </ul>
Workflow (custom)	<ul style="list-style-type: none"> <li>• Custom SAP Build Process Automation</li> </ul>	<ul style="list-style-type: none"> <li>▪ Custom Business Workflow</li> <li>▪ Custom Flexible Workflow</li> </ul>	N/A
Workflow (standard)	<ul style="list-style-type: none"> <li>• Standard Flexible Workflow</li> <li>• Standard Business Workflow</li> <li>• Standard SAP Build Process Automation</li> </ul>	N/A	N/A
Application Job	Custom class for Application Job	Custom class for Application Job calling a Tier 2 wrapper	N/A

- Actually Tier 3, but due to limited availability of Tier 1 alternatives it is considered Tier 2

## Non-SAP SaaS Customization Tier classification

For Non-SAP custom development it is also important to stay within the means of provided tools and technologies applying best practices to achieve and maintain a clean core.

The customizations are approached similar to SAP developments and categorized in Tiers if possible.

### Salesforce

The two key technologies that are used to enhance the Salesforce standard today are Apex and Lightning Web Components (LWC).

Apex and LWC are code-based tools. For UI development, LWC-based low-code tools like Gridmate and Omnistudio are preferred before developing a pro-code UI in LWC. Apex is the Salesforce programming language used for custom logic and interfaces.

LWC based apps use the modern and preferred framework. For workflows, Flow is the preferred no-code tool. Visualforce, Workflow Rules and Process Builder **should not be used** anymore.

Either one of these tools needs to be used with best practices in mind. More details are captured in the [Development Guidelines](#). The Guideline is kept up to date. **By applying the Salesforce Development Guidelines only tier 1 and tier 2 classification applies.**

## Icertis

Icertis as a SaaS system does not allow access to DB and code. Only enhancements allowed are via the provided framework and UI driven.

Only standard APIs can be used.

In case a custom development is needed Icertis will implement and support it.

**Only the Tier 1 classification is applicable.**

## Clean Core Deviation tracking

All Tier 2 and 3 Development IDs are logged in the [Clean Core Deviation Tracker](#). During upgrades, these items are revisited to see whether a Clean Core-compliant replacement is available.

The [Clean Core Deviation Tracker](#) records the following information:

S No.	Pod / Functional Area	L3 Process	L4 Process	System	Development Type	Development Id	Requirement title	Priority	Gap Classification	Extension Tier Classification	Technical Details
-------	-----------------------	------------	------------	--------	------------------	----------------	-------------------	----------	--------------------	-------------------------------	-------------------

## Development ID Counting Rules

Every development requires at least one Development ID. SAP considers all custom code as a core change. All development IDs will be counted. The type of it is irrelevant with the following exceptions:

- Standard interfaces or workflows require a development ID and FD to describe the business requirement and to cover the needed technical configuration. But these will not be counted.
- SaaS system customizations require a Development ID and FD for System Interfaces to describe the business requirement and to cover the needed technical configuration. But these will not be counted.

There is no defined maximum number of Development IDs. Instead, KPIs will be used to measure the degree of adherence to SAP's Clean Core approach:

- Average Clean Core Tier Score: Lower value is better. Should be 2 or less. For example 1.4.
  - Formula:  $(\text{No. of Tier 1 objects} \times 1 + \text{No. of Tier 2 objects} \times 2 + \text{No. of Tier 3 objects} \times 3) / \text{Total Number of Objects}$
- Current percentage of Tier 1 Clean Core Developments: Higher percentage is better. For example 56%.
- Current percentage of Tier 2 Clean Core Developments: This should be lower than the percentage for Tier 1.
- Current percentage of Tier 3 Clean Core Developments: Lowest of the three percentages. For example 14%.

Jira will show the approved Development IDs per Pod, L3/4 Process, Development Type, System, Clean Core Tier and by Release.

## Workflow history

This view shows the 5 most recent entries. The complete workflow log is available from the 'Document Activity' menu item.

Apr 22, 2026	Actor	Type	Activity	Version
--------------	-------	------	----------	---------

Approved	WENNINGER-ext, Sascha	State	changed state to <a href="#">Approved</a> at 10:43 am	v76
Edited following Approval	WENNINGER-ext, Sascha	State	gave <i>Minor change</i> approval at 10:43 am	
	WEINERT-ext, Patrick	State	changed state to <a href="#">Edited following Approval</a> at 10:36 am	v76
Approved	WEINERT-ext, Patrick	Edit	updated the page at 10:36 am	
			<i>Changed the Tiering for Workflows</i>	
<b>Nov 11, 2025</b>				
	WENNINGER-ext, Sascha	State	changed state to <a href="#">Approved</a> at 8:53 am	v75