



User Guide | PUBLIC
2026-10-07

SAP Application Extension Methodology

Content

- 1 SAP Application Extension Methodology Overview. 3**
- 2 The Three Phases of the Application Extension Methodology. 4**
 - 2.1 Phase 1: Assess Extension Use Case. 6
 - Step 1: System Context. 7
 - Step 2: Business Context & Requirements. 8
 - Step 3: Application Extension Use Case. 9
 - 2.2 Phase 2: Assess Extension Technology. 11
 - Step 1: Extension Styles. 12
 - Step 2: Extension Task. 13
 - Step 3: Extension Technology Mapping. 14
 - 2.3 Phase 3: Define Extension Target Solution. 16
 - Step 1: Extension Technology per Extension Task. 17
 - Step 2: Decision Guidance Assets. 18
 - Step 3: Extension Target Solution. 19
- 3 Version History. 21**

1 SAP Application Extension Methodology Overview

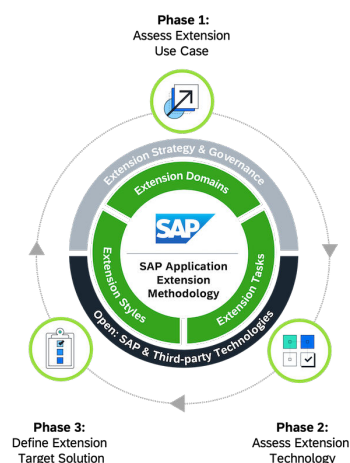
We want to support Enterprise Architects in assessing their extension use cases and defining an extension target solution in a structured and formalized way.

The SAP Application Extension Methodology provides a structured, technology-agnostic approach for customers and partners to define an organization-specific extension strategy.

If you face complex challenges like 'clean core', 'cloud readiness', and 'future proof' concepts, you need to come up with a custom-tailored extension strategy for your organization.

This is where the SAP Application Extension Methodology can support and guide you.

Following this methodology will ensure that everyone in your project uses the same terminology and quickly arrives at a common understanding of the business use case and future solution. You'll have an overview of possible technical extension building blocks and you'll be able to make an informed decision about your organization's future extension architecture. You'll be able to define an extension framework, to provide guidance for the internal audience of architects and developers.



- [Phase 1: Assess Extension Use Case \[page 6\]](#)
- [Phase 2: Assess Extension Technology \[page 11\]](#)
- [Phase 3: Define Extension Target Solution \[page 16\]](#)

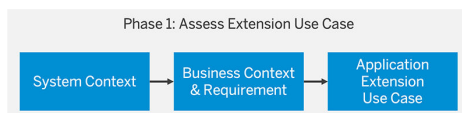
The SAP Application Extension Methodology is a three-phase cycle that is open to SAP and third-party technologies.

You can follow this methodology either for a specific use case (start with Phase 1) or use it to define your extension strategy and governance (start with Phase 2).

2 The Three Phases of the Application Extension Methodology

The Application Extension Methodology consists of three phases that will guide you through the whole process.

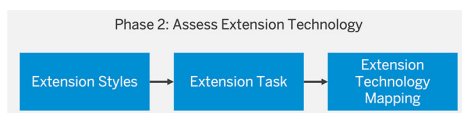
Phase 1: Assess Extension Use Case



- [Step 1: System Context \[page 7\]](#)
- [Step 2: Business Context & Requirements \[page 8\]](#)
- [Step 3: Application Extension Use Case \[page 9\]](#)

Firstly, you assess your extension use case. In this phase, you lay the foundation for any extension application use case based on the business context and requirements in a defined area. You'll understand your system context and create your extension application use case descriptions.

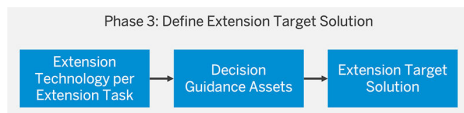
Phase 2: Assess Extension Technology



- [Step 1: Extension Styles \[page 12\]](#)
- [Step 2: Extension Task \[page 13\]](#)
- [Step 3: Extension Technology Mapping \[page 14\]](#)

In this phase, the methodology introduces different terms and concepts like extension styles, extension tasks, and extension domains. You'll get an overview about various extension technologies that we call technical extension building blocks. These building blocks, in combination with the extension tasks, will help you translate business requirements into technical requirements.

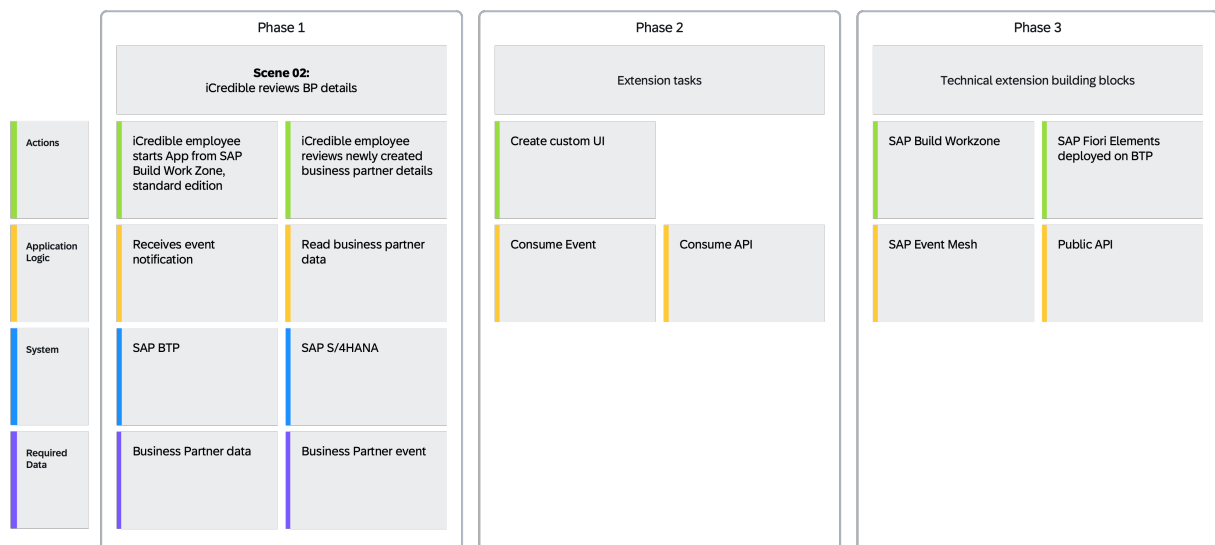
Phase 3: Define Extension Target Solution



- [Step 1: Extension Technology per Extension Task \[page 17\]](#)
- [Step 2: Decision Guidance Assets \[page 18\]](#)
- [Step 3: Extension Target Solution \[page 19\]](#)

Based on the overall requirements and the technology mapping between extension tasks and building blocks in the previous phase, you can make an informed decision on the technical extension building blocks needed to create your target solution. With the help of various decision guidance assets, such as guidance papers and SAP Discovery Center Missions, you can further enhance your target solution.

By following this methodology, you'll create several deliverables and artifacts. Here's a sample solution of a use-case scenario.



Example of a filled out Extension Task Mapping Template

To summarize: In phase 1, you shape the use case. In phase 2, you use the use case to decide on the extension task. In phase 3, you map the extension tasks to the technical extension building blocks and define the target solution.

Afterwards, it will be easy to create an extension target solution diagram.

Related Information

[Phase 1: Assess Extension Use Case \[page 6\]](#)

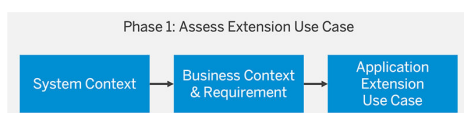
[Phase 2: Assess Extension Technology \[page 11\]](#)

[Phase 3: Define Extension Target Solution \[page 16\]](#)

2.1 Phase 1: Assess Extension Use Case

Assess extension use case of the SAP Application Extension Methodology and lay the foundation for any extension application use case based on the business context.

Description



- [Step 1: System Context \[page 7\]](#)
- [Step 2: Business Context & Requirements \[page 8\]](#)
- [Step 3: Application Extension Use Case \[page 9\]](#)

You'll understand your system context and business requirements, and you'll create descriptions for your extension application use case.

It's important to establish a common understanding of the business challenge you want to solve. You can use the SAP Application Extension Methodology templates or use your own templates to capture and share your results within your team.

There are many ways to achieve a common understanding. For example, you can follow the approach outlined by the SAP Application Extension Methodology, conduct a Design Thinking Workshop, or use advisory and execution services for intelligent custom code management.

Ideally, a 'make-or-buy' decision has been made before you start with the SAP Application Extension Methodology. It's recommended to make this decision as soon as possible after Phase 1.

Goal

The goal is to reach a common understanding about the business challenge and capture the results in a detailed Extension Application Use Case Description.

Deliverables

In this phase, you create three deliverables: System Context, Business Context & Requirements, Extension Application Use Case Description.

Personas

It is recommended to involve the following personas in this phase:

- Enterprise Architect
- Domain Architect
- Business User

Related Information

[Step 1: System Context \[page 7\]](#)

[Step 2: Business Context & Requirements \[page 8\]](#)

[Step 3: Application Extension Use Case \[page 9\]](#)

2.1.1 Step 1: System Context

Get a baseline architecture assessment of all relevant systems and components within the extension project scope and use case.

Context

You describe the systems currently in use 'as-is' and potential future 'to-be' (if you already know upcoming adjustments) as part of the template. You want to identify which systems are available and can be reused for designing the to-be scenario solution. If you're already aware of additional systems, you can also mention this at this step.

List of Solutions							
ID	Solution	Version	SID (host)	Owner	Solution Lifecycle	Deployment	
						As-is	To-Be

System Context Template

For more information, download the [SAP Application Extension Methodology template](#). 

Procedure

1. List all current and future systems that are relevant for the use case.
2. Think about the owner. Who or which department is responsible for this system?
3. Think about the solution lifecycle. What is the strategic direction of this system? Will there be a system conversion in future?

2.1.2 Step 2: Business Context & Requirements

Get a brief description of the overall business requirements and their scope for one or more use cases.

Context

Sometimes the business context can lead to valuable and important insights, which could potentially steer the to-be scenario solution in a certain direction. For example, think about a use case where your persona can only access a mobile device. Such crucial business context and requirement insights can have huge impacts on choosing a suitable technical extension building block later in phase 3. To document your findings, you can use the Business Context & Requirement template.

Business Context & Requirement for each Use-Case	
Title/ID	
Business Context Brief description of business requirement and its scope	
Roles and responsibilities List all the roles and their responsibilities	
Acceptance criteria Describe the acceptance criteria	

Business Context & Requirement Template

For more information, download the [SAP Application Extension Methodology template](#). 

Procedure

1. Describe briefly the business requirement and its scope.
2. List all roles and their responsibilities relevant for the specific use case.
3. Define acceptance criteria.

2.1.3 Step 3: Application Extension Use Case

This is the starting point to translate the business requirements into technical requirements.

Context

To have a common understanding of the business challenge at hand, you can use the Application Extension Use Case Description template.

In this step, you define a to-be scenario solution to solve your business requirement.

The template is divided into four rows: Actions, Application Logic, System, and Required Data. For each row, we provide some leading questions. Such a template is best worked on as a team in a design thinking approach or similar fashion.

Because the creation of a use case description can be a lot of work, you want start with the most valuable use case. You'll reuse and build on top of this use case description in the next phases of the methodology. Each row will help you match a fitting extension task in Phase 2 and a fitting technical building block in Phase 3.

	Extension Name	Business Context	Contributor and Role	Volumetrics	...	Scene description			
Actions	- Description of visualization and interaction - What actions does the user perform in which User Interface?								
Application Logic	- What existing process needs to be extended? - How does the extension work together with existing process application logic? - What application logic needs to happen within this extension?								
System	- Description of existing systems. - Needs to be mapped to system context. - How does the extension work together with existing systems?								
Required Data	- Description of required data for the extension. - Which data is from the core system? - Where is the data for the extension created and stored? - Is there a need for a data persistence in the core solution?								

Application Extension Use Case Description Template

For more information, download the [SAP Application Extension Methodology template](#). 

Procedure

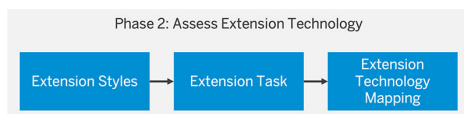
1. Target Scenario: Create a storyboard showing how the user interacts with the target solution. Once the storyboard is created, place the different scenes one after the other horizontally.
2. Actions: Under each scene, write the actions performed by each persona in the system. Indicate the persona name for each action. You want to describe the interactions of the user with the user interface.
3. Application Logic: To realize the user actions, indicate the technical requirements that need to be met. Add technical capabilities that are required at a specific step or action, like specific BTP services. What existing process needs to be extended? How does the extension work with existing process application logic? What application logic needs to happen within this extension?
4. System: Describe in which system the actions and application logic are happening. The systems can be mapped to the system context. How does the extension work with existing systems?
5. Required Data: Describe the required data for the extension. Which data is from the core system? Where is the data for the extension created and stored? Is there a need for data persistence?

2.2 Phase 2: Assess Extension Technology

Assess Extension Technology, the methodology introduces certain terminology and concepts. This allows you to establish a common understanding for a proposed solution.

Description

In this phase, the methodology introduces different terms like extension styles, extension tasks, and extension domains. You'll get an overview of various extension technologies, which are called technical extension building blocks. By understanding these concepts and applying them, it becomes easy to translate the business requirement from Phase 1 (extension application use case description) into technical requirements. In the beginning, this translation continues to be technology-agnostic by using the extension styles and extension tasks. This enables you to think freely about a future solution without thinking about technical limitations, thus freeing up the space for innovative architectural solutions. For example, providing the freedom to explore new technologies on the SAP Business Technology Platform.



- [Step 1: Extension Styles \[page 12\]](#)
- [Step 2: Extension Task \[page 13\]](#)
- [Step 3: Extension Technology Mapping \[page 14\]](#)

If you want to use the SAP Application Extension Methodology primarily to create an organization-wide extension strategy and governance processes, phase 2 and phase 3 will provide you with all necessary concepts to achieve that.

Goal

The goal of this phase is to understand extension styles and extension tasks, in order to map them to your extension use case description. In addition, you'll learn about the available technical extension building blocks from SAP.

Deliverables

You'll create two deliverables: A list of extension tasks and a list of possible technical extension building blocks.

Personas

It's recommended to involve the following personas in this phase:

- Enterprise Architect
- Domain Architect

Related Information

[Step 1: Extension Styles \[page 12\]](#)

[Step 2: Extension Task \[page 13\]](#)

[Step 3: Extension Technology Mapping \[page 14\]](#)

2.2.1 Step 1: Extension Styles

Learn about Extension Styles and how you can use them.

The first new terminology is extension styles. These styles help frame the overall technology for building extensions and structuring extension tasks. Extension styles can be categorized into the Three-Tier Architecture model: Presentation Tier, Application Tier, and Data Tier.

Extension Styles

Three-Tier Architecture	Extension Style	
Presentation Tier	User Interface Extension	Adapt the standard user interface of the core solution by adding, removing, and changing UI elements like labels, buttons, and input fields.
	New User Interface	Create a custom user interface in addition to or in replacement of a standard UI for specific usability needs.
	Form and E-Mail	Adapt existing or create new forms and e-mail templates.
Application Tier	Business Logic Extension	Realize business requirements by implementing additional business logic. Add, exchange, or rewire a process by creating additional process steps.
Data Tier	Data Model Extension	Store additional information by extending the entities of an existing data model, or define a new data model.

2.2.2 Step 2: Extension Task

For every extension style, there are one or more extension tasks. All extension tasks are technology-agnostic and solve a specific part of your use case.

You start to translate your business requirement into more of a technical requirement through extension tasks, by mapping each requirement from the use case description (phase 1) to one or many extension tasks. If you face a business requirement that can't be mapped to an extension task, feel free to adapt the methodology to your needs. However, try to stick to the above-defined extension styles and to the 3-Tier-Architecture model when adding new extension tasks.

At this step, you're still technology-agnostic, thus empowering you to ignore any possible technical limitations. This enables you to think freely without worrying about technical limitations. At step 3, you'll start to think about technology.

Extension Task for Presentation Tier

ID	Extension Style	Extension Task
P01	User Interface Extension	Adapt standard UI: add/hide/create/re-name/re-arrange field/label/headlines
P02	New User Interface	Create custom UI
P03	New User Interface	Create central entry point
P04	Form	Adapt custom form templates based on standard process
P05	Form	Create custom form
P06	E-Mail	Adapt e-mail template based on standard process
P07	E-Mail	Create custom e-mail

Extension Task for Application Tier

ID	Extension Style	Extension Task
A01	Business Logic Extension	Add custom field to UI service
A02	Business Logic Extension	Add custom field to API
A03	Business Logic Extension	Adapt standard business process with custom logic (for example pre-fill/validate field, within LUW)
A04	Business Logic Extension	Create application logic
A05	Business Logic Extension	Create API for UI
A06	Business Logic Extension	Create API for integration
A07	Business Logic Extension	Consume API
A08	Business Logic Extension	Choose event broker technology
A09	Business Logic Extension	Adapt standard events (derived event)
A10	Business Logic Extension	Create custom event
A11	Business Logic Extension	Consume event

ID	Extension Style	Extension Task
A12	Business Logic Extension	Create workflow
A13	Business Logic Extension	Create rules
A14	Business Logic Extension	Create and schedule a background job

Extension Task for Data Tier

ID	Extension Style	Extension Task
D01	Data Model Extension	Add custom field to existing database table
D02	Data Model Extension	Create persistence

Extension Task Mapping Template

The Extension Task Mapping Template helps you connect the extension use case description with the extension task.



For more information, download the [SAP Application Extension Methodology template](#). 📄

2.2.3 Step 3: Extension Technology Mapping

The extension technology mapping outlines the relationship between extension tasks and technical extension building blocks.

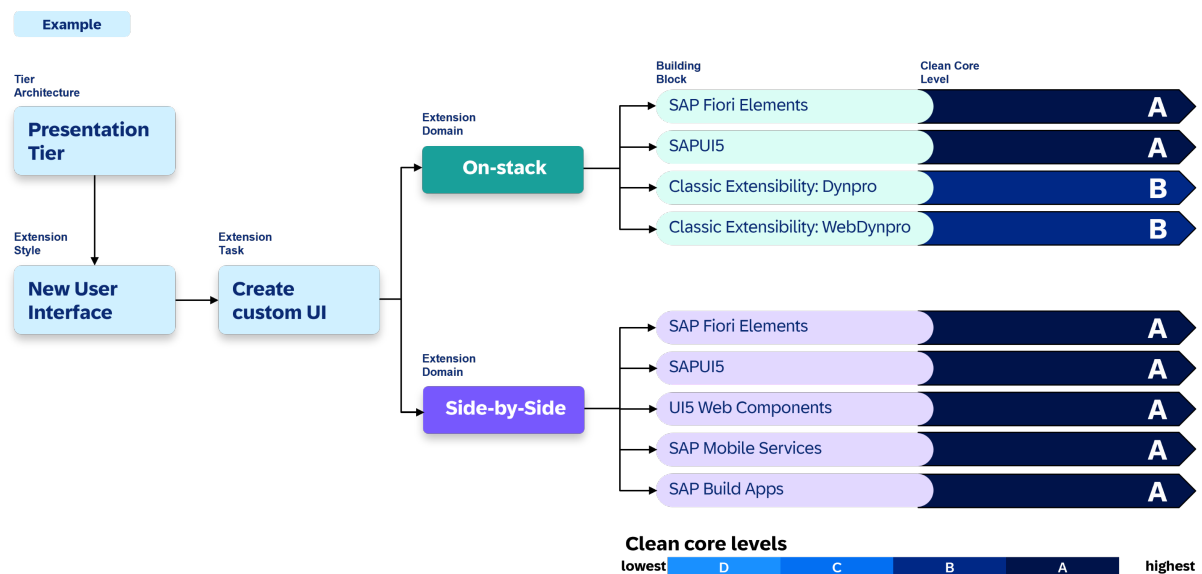
A technical extension building block can be categorized into extension domains such as the core solution extension domain and the side-by-side extension domain.

A very common example for an extension task is to add a custom field to a standard user interface. Let us assume you are in the context of SAP S/4HANA on-premise. To solve this extension task, you have various

technical extension building blocks as options. Useful technical extension building blocks are 'Custom Fields App' in combination with 'Adapting SAP Fiori UIs at Runtime'.

Another common extension task is to create a custom user interface. Many different building blocks are available, ranging from SAPUI5 freestyle UIs (with different deployment options) to low-code/no-code UIs and analytical UIs. As part of the extension technology mapping, you can explore different options that solve these and other extension tasks.

The extension technology mapping includes the clean core levels for the technical extension building blocks. However, the source of truth for clean core levels is SAP Note [3578329](#).



For more information, download the [SAP Application Extension Methodology Extension Technology Mapping](#).

Extension Domain

An Extension Domain describes the “location” of the Technical Extension Building Blocks and explains where it “runs”.

1. On-Stack Extension Domain: Running on the same technology stack as the core solution like SAP S/4HANA Cloud
2. Side-by-Side Extension Domain: Using extensibility capabilities outside the core solution based on standard APIs/events like SAP Business Technology Platform
3. Hybrid Extension Domain: Using a combination of on-stack and side-by-side extension domain.

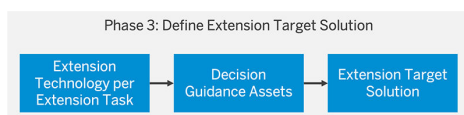
2.3 Phase 3: Define Extension Target Solution

In Phase 2, you decided which extension tasks fit best with your business requirements. Now, in Phase 3, you'll make an informed decision about which technical extension building blocks are suitable for each extension task, thus solving your business use case.

Description

Based on the overall requirements and the technology mapping in the previous phase, you can choose which technical extension building blocks you need to create your target solution. With the help of various decision guidance assets, such as white papers and SAP Discovery Center Missions, you can further enhance your target solution.

By the end of this phase, you'll be able to create a target solution (for example, in the form of a diagram). This will be the preparation for your implementation.



- [Step 1: Extension Technology per Extension Task \[page 17\]](#)
- [Step 2: Decision Guidance Assets \[page 18\]](#)
- [Step 3: Extension Target Solution \[page 19\]](#)

Goal

The goal of this phase is to create an extension target solution.

Deliverables

This phase involves four key tasks: Document the guidance per extension task, decide on the technical extension building blocks, provide the reasoning behind these decisions, and create the extension target solution.

Personas

It's recommended to involve the following personas in this phase:

- Enterprise Architect
- Domain Architect

Related Information

[Step 1: Extension Technology per Extension Task \[page 17\]](#)

[Step 2: Decision Guidance Assets \[page 18\]](#)

[Step 3: Extension Target Solution \[page 19\]](#)

2.3.1 Step 1: Extension Technology per Extension Task

Use the extension task guidance template to analyze which technical extension building blocks are suitable for each extension task.

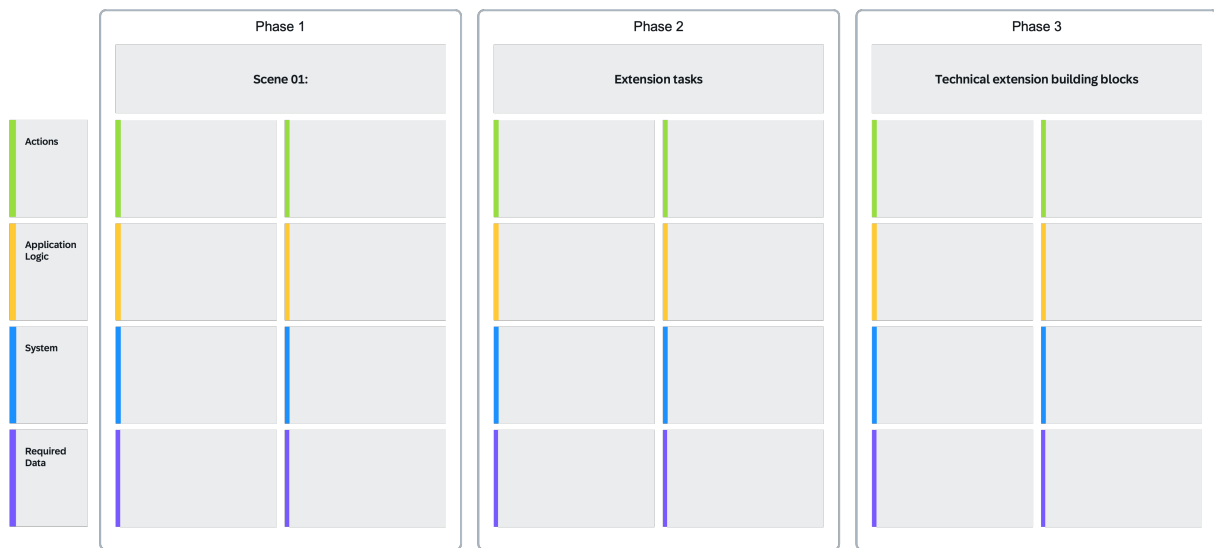
Context

To do a thorough analysis, use the Extension Task Guidance Template and the Extension Technology per Extension Task Mapping Template to document your organization-specific guidance before you make any decision on the technical extension building blocks.

ID	Extension Style	Extension Task	Technical Extension Building Block	Guidance	Color Code
P02	New User Interface	Create custom UI	SAP Fiori Elements	<to be filled out by project team>	
			SAPUI5	<to be filled out by project team>	
			UI5 Web Components	<to be filled out by project team>	
			SAP Mobile Services	<to be filled out by project team>	
			SAP Build Apps	<to be filled out by project team>	
			Dynpro	<to be filled out by project team>	
			WebDynpro	<to be filled out by project team>	

Color Code	Description
	<to be filled out by project team>
	<to be filled out by project team>
	<to be filled out by project team>

Extension Task Guidance Template



Extension Technology per Extension Task Mapping Template

For more information, download the [SAP Application Extension Methodology template](#). 

Procedure

1. Create a new table for each relevant extension task (results from Phase 2).
2. List all technical extension building blocks that are important to discuss.
3. Discuss each building block and establish guidance about when to use each building block.
4. Define and use color coding.

2.3.2 Step 2: Decision Guidance Assets

Familiarize yourself with the following decision guidance assets. These assets can help you make an informed decision. This list is not exhaustive, but you can consider it as a starting point for your learning journey.

SAP Business Technology Platform

Type	Name
SAP Help Portal	Extension Architecture Guide
	SAP BTP Developer's Guide
	SAP BTP Environment options
SAP Discovery Center	SAP Discovery Center Missions 
	SAP BTP Use Cases
	SAP BTP Adoption Framework 

Type	Name
Guidance Paper	Extend SAP S/4HANA in the cloud and on premise with ABAP based extensions
	ABAP Cloud - Technical Use Cases and Recommended Technologies
	Clean core extensibility for SAP S/4HANA Cloud
Blog	SAP S/4HANA Extensibility: All You Need to Know
SAP Help Portal	Extend and Integrate Your SAP S/4HANA

2.3.3 Step 3: Extension Target Solution

Create your extension target solution by reusing the official SAP Business Technology Platform diagrams & icons.

Context

Write down the reasoning and explain why a certain technical building block was chosen. To document your decision, you can use the Extension Target Solution Reasoning template.

With the official set of icons, design guidelines, and sample diagrams, you can create your own SAP Business Technology Platform solution diagrams using Microsoft PowerPoint, draw.io or Lucidchart. These diagrams are intended to help you better understand SAP Business Technology Platform end-to-end solution scenarios as customer, partner, client, solution architect, project lead or developer by using graphical diagrams for visual learning.

For more information, see the [BTP Solution Diagram Repository](#).

ID	Extension Style	Extension Task	Technical Extension Building Block	Reasoning
<id>	<extension style>	<Which extension task has been chosen>	<Which technical extension building block has been chosen>	<to be filled out by project team>

Extension Target Solution Reasoning template

For more information, download the [SAP Application Extension Methodology template](#).

Procedure

1. Fill out which extension tasks have been chosen.
2. Fill out which technical extension building blocks have been chosen.
3. Provide your reasoning and justify your decision.

3 Version History

SAP Application Extension Methodology version history

Version	Release Month
v2.4	August 2025
v2.3	March 2025
v2.2	December 2023
v2.1	November 2022
v2.0	March 2022
v1.0	June 2021

Latest Changes

Title	Description	Version
Update: SAP Application Extension Methodology Extension Technology Mapping	Update on the Extension Technology Mapping including Clean Core Levels	2.4
Update: Extension Tasks	Update on the Extension Tasks	2.3
Update: SAP Application Extension Methodology Extension Technology Mapping	Update on the Extension Technology Mapping like event-drive architecture	2.3
Update: BTP Solution diagrams & icons	The BTP Solution diagram and icons are not open source	2.3
Update: Extension Styles	Update on the Extension Styles	2.2
Update: Extension Tasks	Update on the Extension Tasks	2.2
Update: SAP Application Extension Methodology Extension Technology Mapping	Update on the Extension Technology Mapping	2.2
Update: Application Extension Use Case Description Template	Update on the Application Extension Use Case Description template.	2.1
New: Application Use Case Scene Template	For Phase 1, we introduce a new template to provide a scene overview about the application extension use case.	2.1
New: Extension Task Mapping Template	For Phase 2, we introduce a new template to document your extension task mapping to your use case.	2.1

Title	Description	Version
New: Extension Technology per Extension Task Mapping Template	For Phase 3, we introduce a new template to document your extension technology per extension task mapping.	2.1
New: Extension Task Guidance Template	For Phase 3, we introduce a new template to document your organization-specific guidance.	2.1
New: Extension Target Solution Reasoning Template	For Phase 3, we introduce a new template to document why certain technical building blocks were chosen.	2.1
Update: SAP Application Extension Methodology Extension Technology Mapping	Update on technical extension building blocks.	2.1

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon  : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
 - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
 - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon  : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

© 2026 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.

