SAP S/4HANA Manifesto & Custom Code Whitepaper

June 2020

CUSTOMER

THE BEST RUN SAP

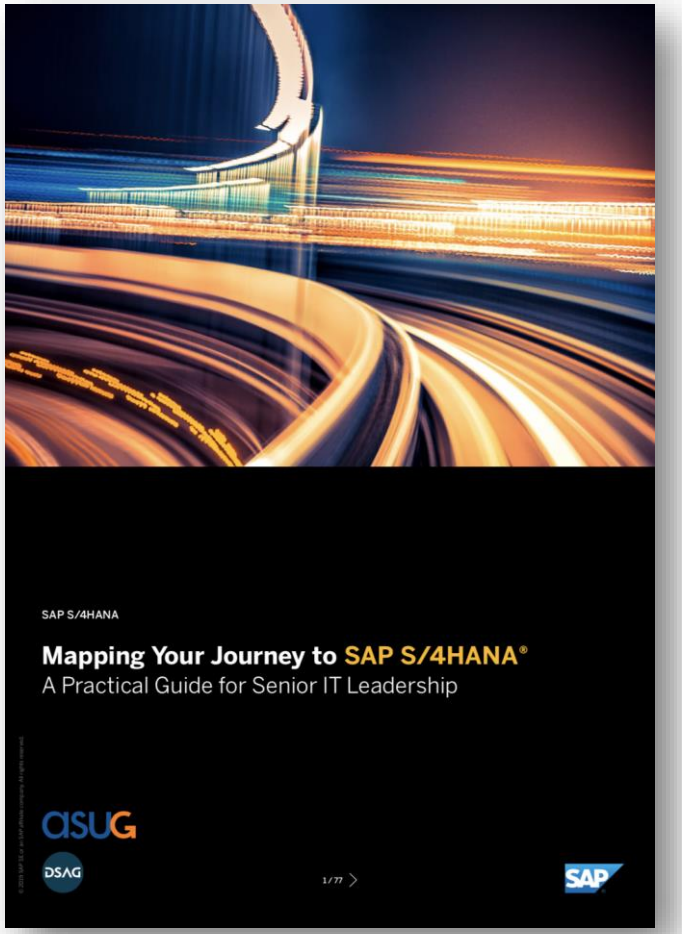
Disclaimer

The information in this presentation is confidential and proprietary to SAP and may not be disclosed without the permission of SAP. Except for your obligation to protect confidential information, this presentation is not subject to your license agreement or any other service or subscription agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or any related document, or to develop or release any functionality mentioned therein.

This presentation, or any related document and SAP's strategy and possible future developments, products and or platforms directions and functionality are all subject to change and may be changed by SAP at any time for any reason without notice. The information in this presentation is not a commitment, promise or legal obligation to deliver any material, code or functionality. This presentation is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. This presentation is for informational purposes and may not be incorporated into a contract. SAP assumes no responsibility for errors or omissions in this presentation, except if such damages were caused by SAP’s intentional or gross negligence.

All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.

Mapping your Journey to SAP S/4HANA –
A practical guide for Senior IT Leadership



"Thank you to SAP for putting together this must-read document for customers exploring the migration to SAP S/4HANA."

Geoff Scott and Chris Crone, ASUG

Click or scan

Why should you read it?

Succinct

Combined experience form over 5,000 projects we have seen so far

SAP + DSAG & ASUG

Written by 70+ subject matter experts from SAP

Regularly updated

Combines technical, architectural and project management view

Over 50,000 downloads in 12 months

What is it about? Whom is it for?

What decisions will ultimately shape our project?

What are key success factors for the project?

What tools shall we leverage?

Table of Contents

STRATEGIC CHOICES

Eight pivotal questions

Table of Contents

STRATEGIC CHOICES | Conversion vs. New Implementation

Figure 1: Key Considerations Influencing System Conversion Versus New Implementation

Do current business processes support your long-term strategy?
Strategic redesign of the business processes suggests a **new implementation**.

Can you adopt SAP® Best Practices packages or will you retain past customizations?
A move to standardization brings agility, suggesting a **new implementation**.

Is your move to SAP S/4HANA® driven by the business or IT?
IT-sponsored projects are typically **conversions** to SAP S/4HANA, which lay the foundation for incremental innovation projects driven by the business.

Can you convert from the SAP ERP application to SAP S/4HANA in a single step?
Single-step **conversion** is possible for SAP ERP 6.x (any enhancement package) single-stack, Unicode systems. Systems that don't fulfill these criteria have likely experienced little maintenance in the past years. If the system can't be converted in a single step, a new implementation is likely a better choice.

Do you require previous transactional data in the new system?
The requirement to retain **all** data in the system is a very strong indication for **conversion**. Alternatively, consider a **new implementation** while replatforming your current SAP ERP data on commodity hardware or leveraging data retention solutions.

Are landscape consolidation and process harmonization key value drivers?
Consider a **new implementation** and consolidate the required configuration and data into that new system.

Do you have a high or low number of interfaces to other systems (SAP and third-party)?
The higher the number of interfaces, the stronger the case is for **conversion**.

Can your company sustain a multiyear innovation plan with incremental innovations?
If incremental innovation is part of your company's philosophy, **a system conversion** followed by innovative projects will lead to the desired outcome. If you are uncertain whether a multiyear innovation plan can be sustained, **a new installation** is the only chance to harvest the full value.

PART ONE – STRATEGIC CHOICES

Do Current Business Processes Support Your Long-Term Strategy?

If your long-term strategy implies the need for business process redesign in the business areas considered key to strategic growth, or the ones expected to deliver substantial cost savings, this is a strong indication for a new implementation.

If your SAP ERP system today takes no advantage of best practices or relies on dated functionality (for instance, business areas instead of profit center accounting), a new implementation is a better choice. Likewise, if you run an oversized, overcomplicated, historically grown system, a new implementation is a more attractive option.

Can You Adopt SAP Best Practices Packages or Will You Retain Past Customizations?

Do you plan to make extensive use of SAP Best Practices packages and SAP Model Company services? If so, a new implementation is a better choice.

By contrast, if you see your custom enhancements and modifications as a major asset supporting your company's unique way of operating and intend to preserve them, a conversion is a more attractive option for you.

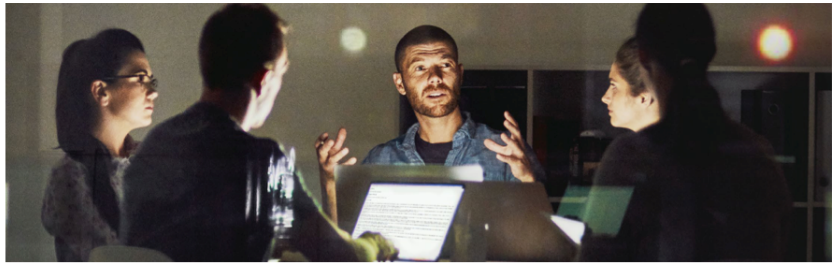
Is Your Move to SAP S/4HANA Driven by the Business or by IT?

It's virtually impossible to start a business transformation out of an IT project. IT-sponsored projects are typically system conversions that lay the foundation for later innovation projects driven by the business.

Can You Convert from the SAP ERP application to SAP S/4HANA in a Single Step?

Technically, single-step conversion is possible for SAP ERP 6.0 (any enhancement package) single-stack, Unicode systems; database and OS-level restrictions may apply. Systems that don't fulfill these criteria have likely experienced little maintenance in the past years. In practice, systems with dated software release levels may require somewhat more effort than the ones recently updated.

If the system can't be converted technically in a single step, a new implementation is a better choice, because the combined cost of an upgrade to SAP ERP 6.0 or a Unicode upgrade followed by a conversion to SAP S/4HANA would be prohibitively high. Moreover, combining two upgrades in a single downtime will most probably exceed the maximum system outage your business can afford.



STRATEGIC CHOICES

- Shall we upgrade to Ehp8 first?
- Shall we go to ERP powered by SAP HANA first?
- Shall we do a process-by-process roll-outs?

Table of Contents

STRATEGIC CHOICES

- What are preparation projects?
- Mandatory vs. optional preparation projects

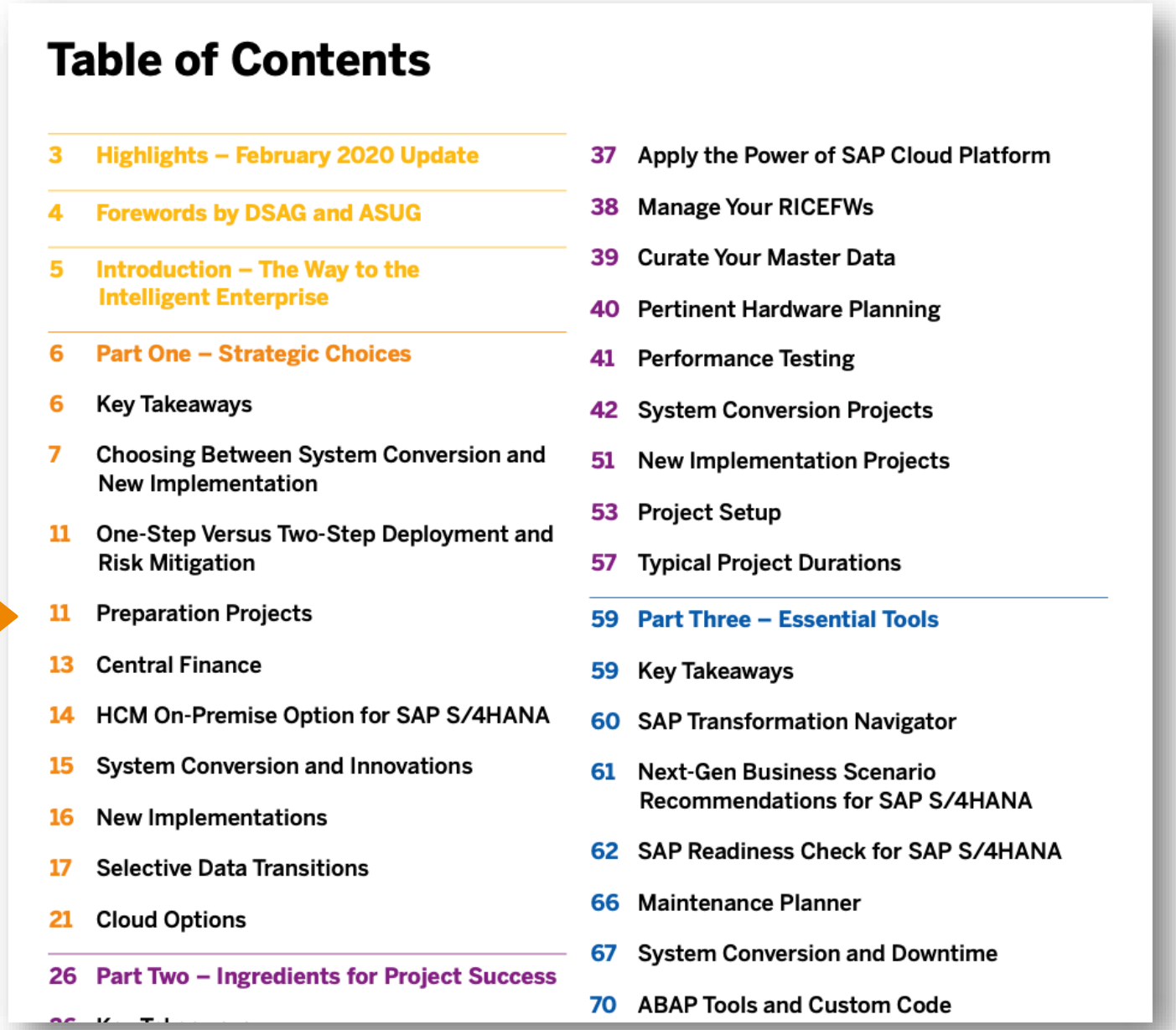


Table of Contents

STRATEGIC CHOICES

Is Central Finance good for us?

Table of Contents

STRATEGIC CHOICES

What happens to our HCM?

Table of Contents

STRATEGIC CHOICES

Don’t stop after conversion

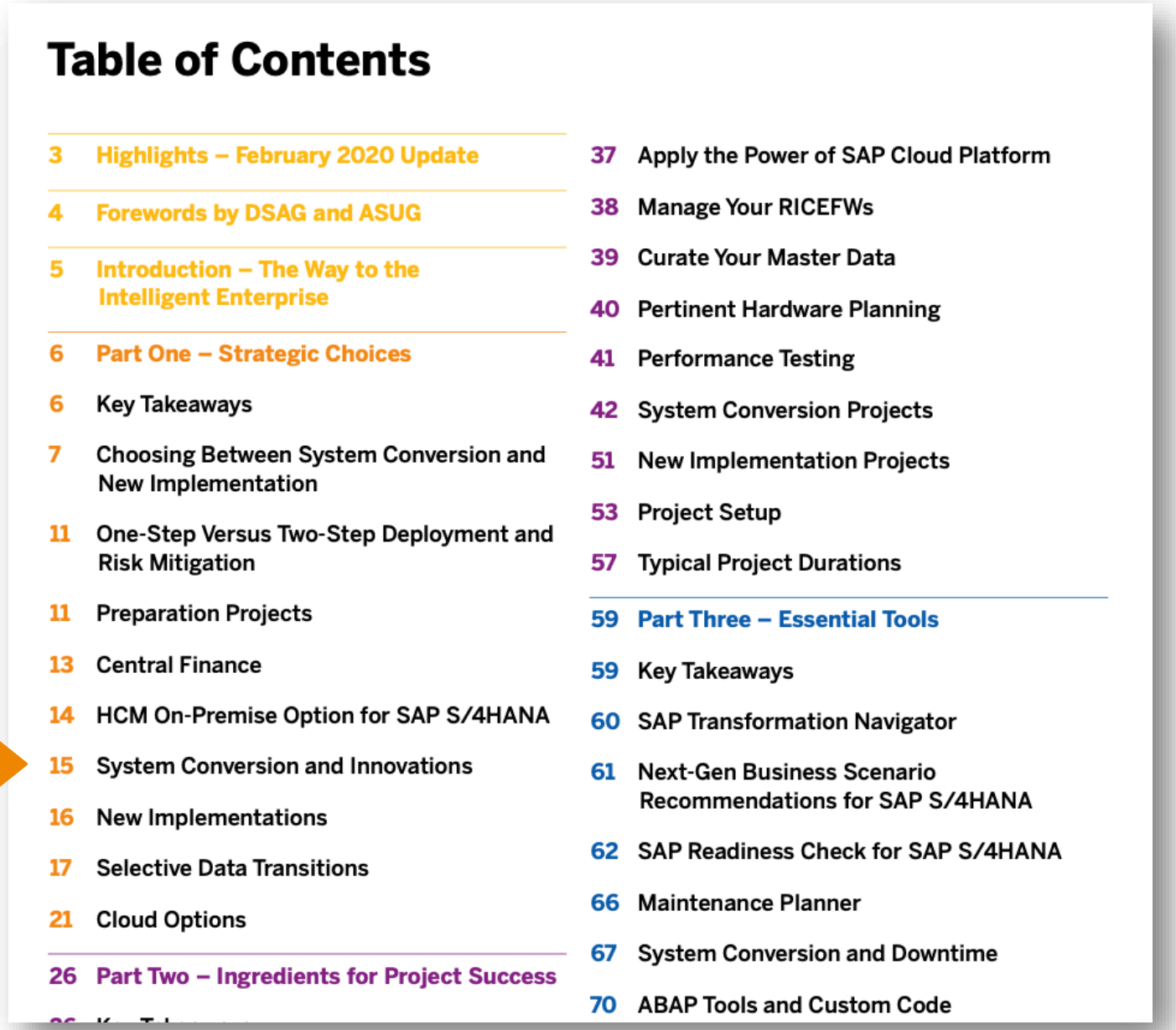


Table of Contents

STRATEGIC CHOICES

SAP Model Companies + Clean Core

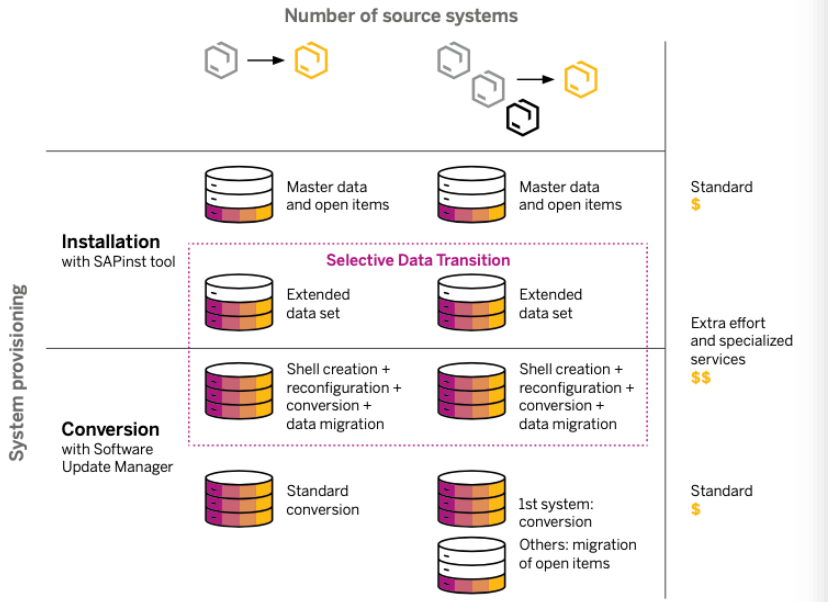
Table of Contents

STRATEGIC CHOICES

Table of Contents

STRATEGIC CHOICES | Selective Data Transitions

Figure 2: Selective Data Transitions with Complex Scenarios Framed with a Purple Dotted Line



Example 1: ERP Landscape Consolidation

One common situation is a consolidation of multiple SAP ERP systems into one. A standard transition path is either:

- Implementing the new system based on best practices followed by loading master data and open items from all source systems
- Converting one of the systems and loading open items from the others

If your requirement is to load historic data from **all** SAP ERP systems that are subject to consolidation, you will have to resort to a selective data transition and employ specialized tools and services.

This pattern sometimes appears in the context of mergers and acquisitions, when an organization needs to integrate the SAP ERP system of an acquired entity into an SAP S/4HANA system (see "Case Study" sidebar).

CASE STUDY: SUPPORTING M&A INTEGRATION

Preparing for future expansion and growth, a global player in the mill products and mining industries implemented SAP S/4HANA® in 2017 through a system conversion. About a year later, the company set out to integrate new factories acquired from a European firm that was running its business on an older release of the SAP® ERP application. Each factory was to receive its own company code in SAP S/4HANA ("company code split") and to assume the new global processes for both logistics and finance, including new accounting principles. To help ensure smooth integration and business continuity, the company wanted to retain access to the factories' historic data in the logistics applications, especially materials management and plant maintenance.

Given the combination of these requirements, the project team saw selective data transition as an attractive option and chose the SAP Digital Business Services organization as its implementation partner. The combined team leveraged SAP® Landscape Transformation software to migrate the master data as well as the last 12 months of selected transactional data at the table level, while loading the financial open items through regular postings.

The project took 10 months with three test cycles, followed by a productive migration.

Example 2: The Shell Approach

If the system is deemed to be in good shape, a project team may seek an approach on how to change only a part of the system configuration or functionality, while retaining the rest unaltered. Examples for such selective changes could be to restructure the chart of accounts or introduce the new general ledger (G/L) functionality with a ledger solution for parallel accounting.

Technically, such an approach includes these steps:

1. Performing a shell creation from the current SAP ERP
2. Performing corresponding customizing and configuration changes in this shell system for the simplification list items
3. Executing a standard system conversion of the shell system
4. Performing corresponding customizing and configuration changes in the SAP S/4HANA system to implement improvements and innovations
5. Loading the data

This approach is within the realms of standard as long as you load master data and open items using the SAP S/4HANA migration cockpit.

However, loading historic data into the new system entails an extra effort and use of specialized tools and services. Therefore, you should first evaluate if you can achieve the same outcome through a preparation project followed by a standard system conversion. The SAP Digital Business Services organization and SAP partners provide well-established services for such preparation projects, such as migration to the new G/L functionality, reorganization of the chart of accounts, or merge of controlling areas.

STRATEGIC CHOICES

Table of Contents

INGREDIENTS FOR PROJECT SUCCESS

ANY PROJECT

SPECIFIC FOR CONVERSIONS

SPECIFIC FOR NEW IMPLEMENTATIONS

PROJECT SETUP

DURATION

Table of Contents

INGREDIENTS FOR PROJECT SUCCESS | System Conversion Projects

- FINANCIAL DATA
- AVAILABLE TRANSITION OPTIONS IN FINANCE
- CONVERSION TEST CYCLES
- ADD-ONS
- KNOW YOUR SIMPLIFICATION ITEMS
- CUSTOM CODE : RETHINK, NOT JUST REWORK

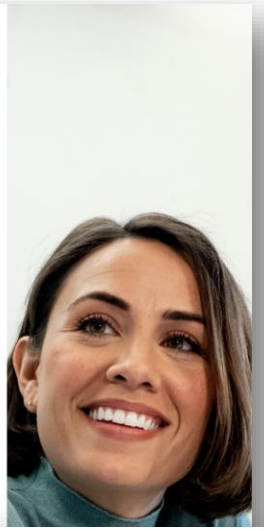
SYSTEM CONVERSION PROJECTS

This section covers the key elements of a successful system conversion, including how to handle your financial data, conversion test cycles, add-ons, simplification items, and custom code.

Take Care of Your Financial Data

There are two facts that many conversion projects are late to realize. First, financial line items increase runtime during the conversion. With more than 1 billion financial line items in the BSEG table (which stores accounting document information in SAP ERP), the project needs to employ either the downtime-optimized conversion option or the minimized downtime service from SAP Digital Business Services to complete the conversion within an acceptable system outage window.

Secondly, among hundreds of millions of records in the old finance (FI) data model that your system would have accumulated over decades, there may be some that are technically inconsistent. A frequent example is missing open items for an open-item managed account. You need to work out a plan together with your accountants for how to resolve these.



For any system with a significant FI
recommends running such an analy
sion project. SAP Note 2755360 gi
so. See KBA 2714344 for recomme
with the most common error messa

Consequently, archiving financial da
effects. On the one hand, it will short
possibly make the desired system ou
with the standard conversion option
team will likely have fewer technical i
the past fiscal years' data.

Conversion Test Cycles

Conversion test cycles are the backbone of a conversion project (see Figure 5). As a practical approach, SAP recommends that you follow the guidelines below to establish a sustainable project plan:

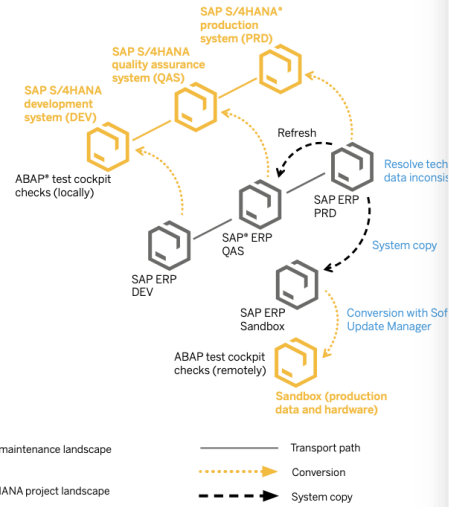
- Test your first conversion with a copy of the production ERP with the standard Software Update Manager conversion procedure and understand the individual phases, steps, and associated runtimes. Be prepared to see a long runtime of the first-pass conversion on a larger system.
- Using a copy of the current production system as a source system in the first conversion cycle is non-negotiable. Using a production-like hardware for the target SAP S/4HANA system in this cycle is highly recommended, especially to obtain the realistic execution times and make a reliable estimate of the expected business downtime. Conducting the first conversion cycle on a development system certainly helps your team to comprehend the technical procedure, but won't take them any further. Thus, let your team find potential problems early and track the resolution.
- Carefully execute all functional preparation steps already in the first sandbox cycle. Do not skip or short-cut activities impacting subsequent steps in the conversion procedure to get a full picture of the required tasks.
- Have a detailed project plan for each conversion cycle. Improve and refine it with each iteration.
- Create a conversion runbook. Use it to log all required functional and technical activities in a conversion cycle and the associated completion times.
- After the first conversion cycle, negotiate with business users what system outage window is acceptable and decide on the technology option: standard conversion or downtime-optimized conversion.
- Having chosen the option, plan for at least two additional conversion test cycles with production data and production hardware. One of these cycles should also include tests on the connected satellite systems to validate the integration.
- Once the new development (DEV) system has been b
the current
freeze"). Yo
manually ir
automated
- Create the
(that is, an
for the cut
ance syste
refine it. It
of producti

Available Transition Options in Finance with SA

In Use Today

General Ledger (G/L)	Parallel Accounting (Ledger Approach Vs. Accounts Approach)	Document Split	Transition Op
New G/L	Ledger approach	Yes	System conve
New G/L	Ledger approach	No	1. System co 2. Subsequer
Classic G/L	Accounts approach	No	1. System co 2. Subsequer **Currently not** approach **Options toda** a) Implement b) Perform system conversion and keep the accounts approach. Be aware that any new features and functions in parallel accounting will be based on parallel ledgers, not the accounts approach. Providing a transition path from the accounts approach to the ledger approach after a conversion is on SAP's road map.
Classic G/L	No parallel accounting	No	1. System conversion 2. Subsequent implementation of document split 3. Subsequent introduction of a new G/L

Figure 5: Conversion Cycles in a Three-System Landscape



Baseline Plan for System Conversion in a Typical Three-System Landscape

	Standard	Downtime Optimized
1st Sandbox	• Create or refresh the source sandbox system with a copy from the production system. • Perform standard conversion of the sandbox including finance conversion. • Analyze and evaluate possible technical data inconsistencies in finance. Resolve as many technical data inconsistencies in finance as possible and complete the conversion. • Resolve all critical technical data inconsistencies in finance in production system. • Connect the new sandbox to the development system. Use the new "Custom Code Migration" SAP Fiori® app to perform the scoping and prepare the deletion of unused custom code. Perform custom code analysis with the ABAP® test cockpit remotely on the development system to understand the impact and plan the necessary custom code adaptation.	
2nd Sandbox	**Optional for small systems** • Refresh the source sandbox with a copy from production • Repeat standard conversion, apply lessons learned for runtime optimization	• Refresh the source sandbox with a copy from production • Perform downtime-optimized conversion, reusing finance customizing from the 1st cycle
Development System (DEV)	• Perform standard conversion, use the prepared transport to delete unused code • Use the local instance of ABAP test cockpit to check the custom code and apply the ABAP quick fixes	
Quality Assurance System (QAS)	• Refresh QAS – unless it already contains a recent copy of production or there are data restrictions (for example, QAS requires a very specific set of test data) • Perform a standard conversion of the QAS	
Optimization (Sandbox)	**Optional for small systems** • Production hardware for the SAP HANA® business data platform • Conversion runtime optimization – if required	• Production SAP HANA hardware • Conversion runtime optimization and fine-tuning
Trigger Test, Load Verification, and Finance Online Conversion **Production System (PRD)**	N/A	• Execute downtime-optimized conversion on production system until the downtime phase • Test trigger creation and replay • Execute the downtime phase of the conversion on the production system copy
Dress Rehearsal	• Production SAP HANA hardware • Exact execution of the cutover plan on a PRD copy • Include satellite systems	• Production SAP HANA hardware • Exact execution of the cutover plan • Trigger and data replication test in PRD • Downtime tasks executed on PRD copy • Include satellite systems
PRD System	• Standard conversion and cutover	• Downtime-optimized conversion and cutover

ESSENTIAL TOOLS | Companions for your SAP S/4HANA journey

For System Conversion

SAP Readiness Check
Assess functional & technical impact,
understand the effort drivers, and plan mitigations

Check your data prior to conversion on SAP ERP
Customers & Vendors
Finance data

Custom code adaptation – ABAP Development Tools
SAP Code Inspector checks, Custom Code Migration app (also SaaS), Quick Fixes

Downtime-optimized conversion (doC) with Software Update Manager
for a reduced technical downtime

Other
SAP Solution Manager retrofit, ANST for automated SAP Notes detection, obsolete data handling,…

For New Implementation

SAP Cloud Platform Integration Advisor
with 10,000 new interfaces

SAP S/4HANA migration cockpit
more ready-to-use business objects
and object modeling capabilities

While SAP continues to improve these tools, customers should note that most of them are already available and ready for use.

Custom Extensions in SAP S/4HANA Implementations

A practical guide for Senior IT Leadership

"Valuable support and practical tips for SAP customers who are dealing with extensions in their SAP S/4HANA projects."

Ralf Peters, DSAG

Click or scan

What is it about?

Why do you have to deal with the new concepts?

How to structure work on custom code in a system conversion?

How to do it right if you decided to go for a new implementation?

What is the right design for new enhancement apps?

Table of Contents

PART TWO – CUSTOM CODE IN SYSTEM CONVERSIONS

REMOVING OBSOLETE CUSTOM CODE

Our statistics from the past 15 years show that 30% to 60% of custom code in an SAP ERP system is not executed. Another significant part can get replaced with SAP standard code upon system conversion. Thus, removing obsolete code significantly reduces the effort for custom code adaptation. At the same time, this is certainly a first step toward a clean digital core.

Building a business case for obsolete code removal initiatives was a challenge in the past, primarily because of the associated testing effort and the difficulty in measuring the business value or savings. Both these factors change in an SAP S/4HANA conversion. First, you can show immediate savings because you don't have to adopt the deleted code. Second, the safety concerns ("Will it still run smoothly after we delete what we ostensibly don't use?") are addressed by the comprehensive functional tests executed by your team as part of the conversion.

If you decide against deleting obsolete code, you will need to adopt it too. Otherwise, you will introduce the risk of system dumps and even data inconsistencies.

SAP has made the deletion procedure an integral part of the conversion and has enabled it through integration between the "Custom Code Migration" app in SAP S/4HANA and the Software

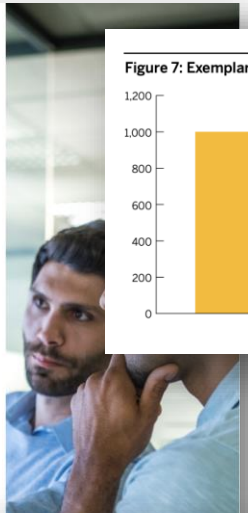
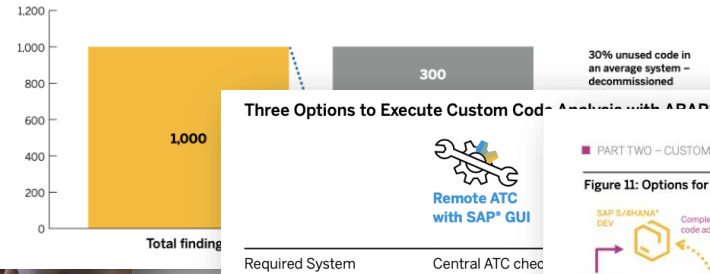


Figure 7: Exemplary Calculation for 1,000 Findings in Custom Code



Three Options to Execute Custom Code Analysis with ABAP® Test Cockpit (ATC)

	Remote ATC with SAP® GUI
Required System	Central ATC chec (SAP_BASIS 7.52
System Location	In customer land
Remote Connectivity	Through RFC
Analysis of SAP S/4HANA® Findings	✔
Simplification Information in ATC Results	✔
Results filtered by Scope and Quick Fix Availability	X
Definition of Custom Code Migration Scope Based on Usage Data	X
Removal Unused Code During System Conversion with Software Update Manager	X

Figure 8: Deletion of Unused Custom Code Integrated into C

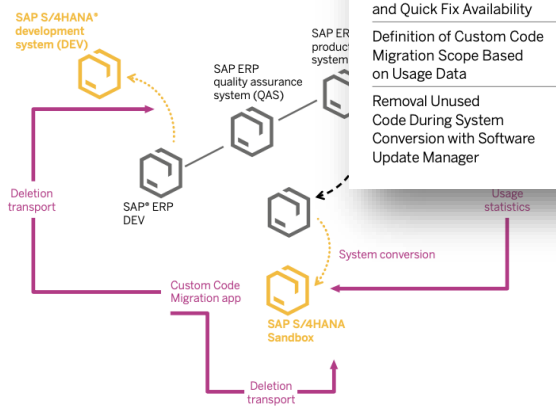
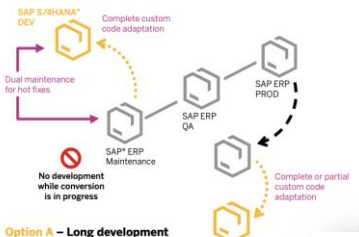
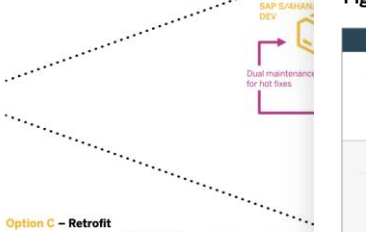


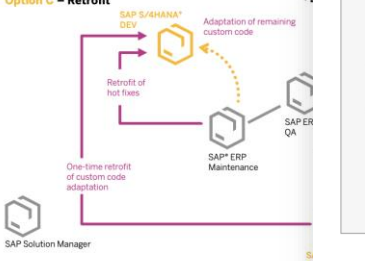
Figure 11: Options for Performing Custom Code Adaptations in a Sandb



Option A – Long development freeze, hardly an option for most businesses



Option C – Retrofit



Here is a piece of practical advice:

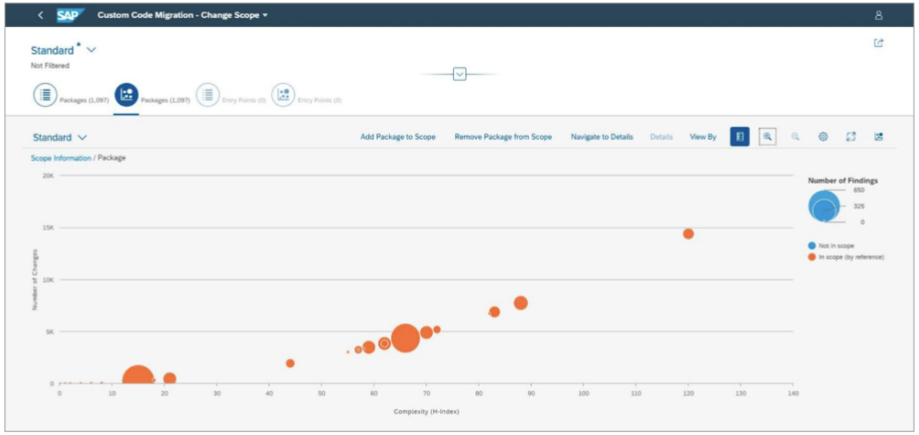
- Declare "zero modifications" as one of your project's goals.
- Treat modification analysis as a work package, not as a task, and schedule enough time for SPDD and SPAU during the sandbox conversion. Name solution architects responsible for making decisions on modification reimplementation.
- Analyze all modifications. Don't be intimidated by the numbers; the actual modifications are usually much fewer.
- Review all SAP code clones with the clone finder tool, implicit enhancements at begin, and class-method overwrite enhancements with transaction SE84 or SPAU_ENH. Treat them as "indirect modifications" and replace them with the SAP standard code in combination with explicit enhancements.
- Use the categorization defined in the table below when working on modifications.

In some cases, the current modifications can be replaced with the in-app extensibility mechanisms in SAP S/4HANA. However, modifications can hardly be replaced with side-by-side extensions.

To track your progress, use the following KPIs:

- Number of modifications reverted
- Number of clones eliminated
- Number of required modifications

Figure 15: Complexity Analysis in "Custom Code Migration" SAP Fiori® App



PART THREE – CUSTOM EXTENSIONS IN NEW IMPLEMENTATIONS

KEEP THE CORE CLEAN

A clean core paradigm requires changes in development processes and practices. Changing old habits may be hard at times. So besides updating your development guidelines, you should also establish a set of control mechanisms:

- Apply a zero-modification policy from the project's first day and use the "Modification Overview" (transaction code SE95).
- Make sure the new code is compatible with SAP S/4HANA. Make an ABAP Test Cockpit check mandatory for transport request release and use the S4HANA_READINESS check variant.
- Use white-listed APIs only; use SAP API Business Hub[5] to explore these.
- When approving development requests, check if your team has fully exploited the in-app extensibility options for UI adaptation, custom fields and tables, operational reporting, and analytics. Inserting a checklist with in-app and side-by-side extensibility options or a questionnaire into your development request template may turn out to be very useful.

Needless to say, a succes
requires your team to und
new technological option

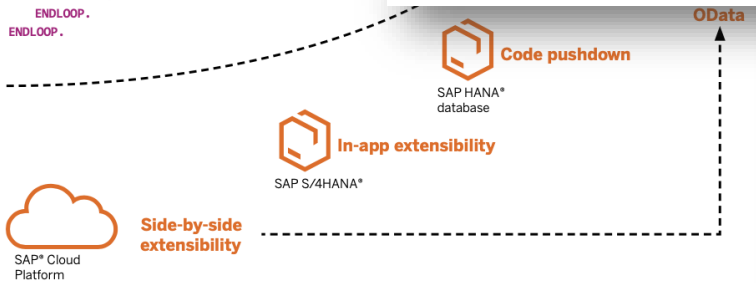
5. SAP API Business Hub is cur
Cloud deployment option. It is g
events for SAP S/4HANA.

SAP's rationale behind
the **clean digital core paradigm**
is simple: allow
customers to extend
their SAP S/4HANA

Figure 17: Custom Code Evolution at SAP

1995

```
SELECT * FROM Employee INTO TABLE it_empl WHERE orgun
LOOP AT it_empl.
    WRITE it_empl-id.
    WRITE it_empl-name.
    SELECT * FROM Addresse INTO TABLE it_addrs WHERE i
    LOOP AT it_addrs.
        IF it_addrs-type = 'HOMEADDR'.
          WRITE it_addrs-zipcode.
        ENDIF.
    ENDLOOP.
ENDLOOP.
```



Traditional RICEFW* Versus New Technologies

Traditional RICEFWs	New Technology
Reports (analytics)	• Real-time analytics and KPI tiles with SAP® Smart Business cockpits, and drill-down analysis with the Analysis Path Framework service in SAP S/4HANA* embedded analytics • Custom analytical application with multidimensional reporting in SAP S/4HANA • SAP Analytics Cloud solution • SAP Fiori® apps
Reports (automation)	• SAP Intelligent Robotic Process Automation services
Reports (applications)	• Custom applications on SAP Cloud P Extension Factory, decoupled from th the SAP Cloud Platform Enterprise M • Custom SAP Fiori apps deployed eith
Interfaces	• Extension of standard OData service services (CDS) views with the in-app • SAP Cloud Platform Integration Suite • SAP Application Interface Framework • Event brokering using SAP Cloud Pla
Conversion programs	• SAP S/4HANA migration cockpit for
Enhancements	• Custom business logic with the in-ap • SAP Cloud Platform Extension Factor
Workflows	• SAP S/4HANA flexible workflow • SAP Cloud Platform Workflow service
Forms	• SAP S/4HANA output management: as data source
Custom tables	• Custom business objects with a gene of SAP S/4HANA
Modifications	• You shouldn't have to make any. In-ap business requirements for UI adapta
User interface	• SAPUI5 and SAP Fiori user experienc
Performance	• Code pushdowns with the SAP HANA

*Reports, interfaces, conversions, extensions, forms, and workflows

AVOID LIFT AND SHIFT

Reusing the custom code from today's SAP ERP is often considered rather obvious. Many projects refer to such an approach as "lift and shift." However, before you endorse copying the old code into the new system, we would like you to reflect on the following facts.

First, the entire body of custom code that you copy over to the new system would have to undergo the same code adaptation procedure as in a system conversion. Complex code pieces are likely to also have quality issues and may need performance optimization too. Also, you may often end up copying much more code to your system than you originally intended because of the dependencies to other development packages. Needless to say, the associated effort grows linearly with the amount of code you copy over.

Second, and more important, many of your current implementations are likely to be outdated both technically and functionally (that is, they may rely on modifications, old-style enhancement techniques, or old business logic). Therefore, you may want to establish a procedure for code reuse requests. At a m
procedure should include the f

- Reconfirm the need for the
functionality with the busine
- Evaluate SAP and partner s
gate the SAP Store site for
and SAP App Center or Cer
Directory for partner solutio
ing replatforming options fo
code.
- Reevaluate if the same can b
in-app extensibility.
- Reevaluate if it qualifies for
extension on SAP Cloud Plat

That being said, executing an "i
ABAP transport requests from
SAP ERP DEV system is probab
you can do to your new implem

For more details, see the sectio
Key Architectural Patterns and
on page 48.

STARTING WITH SAP CLOUD PLATFORM

Take the following steps when getting started with SAP Cloud Platform.

Choose a Use Case
Asking your team or implementation partner to showcase how the cloud can solve the urgent challenges may seem very natu
sight. However, there is a trap: you ma
picking a use case that's not suitable f
cloud at all, make your implementatio
have a hard time building it, and dissat
business in the end.

When talking to customers' teams, we suggest starting with a use case that a to demonstrate the practical benefits cloud development. Many choose to s internal application to support their er day-to-day tasks. Building their first ap without external exposure will take so sure off your team. At the same time, t everyone in the company to collect ex with cloud applications and better ima different business departments can b the same.

Connect Back-End Systems with SAP Cloud Platform
Whichever use case you decide to star nect your SAP Cloud Platform accoun SAP S/4HANA landscape (on premise and your identity provider. This is esse side-by-side extensibility and will mak applications productive from day one.

Many companies have a copy of their system. Connecting your SAP Cloud P development subaccount to such a sy boost your team's productivity and he business buy-in when presenting prot early versions on real data.

Build for the Cloud
With the new technological options and programming models, there is no one-to-one migration of custom applications to the cloud. You should see it as an opportunity rather than a tedious rework.

Apply DevOps Practices
These tips can help you put DevOps to work in your development organization:

- Ask a technology expert from your team to study the DevOps methodology, take ownership of the pipeline including any extra tool you may want to plug in, and coach the development team(s).
- Choose and configure the code quality checks and security scans to match corresponding standards in your organization. Do not rely on default settings.
- Adhere to "build once and fail fast" principles as well as the mandatory participation and continuation of the scrum ceremonies
- Constantly improve the development process and automation tools. However, better does not mean more control. DevOps is not about strict rules and governance but about the efficiency of the development teams. Keep this in mind when conducting process reviews.
- Consider the DevOps metrics listed in the following table. These will help you to have a clear picture of your efficiency, code equality, and applications usage. Most pipeline tools offer APIs to retrieve the corresponding data points.
- Clearly, each organization needs to decide for itself on the target values for these KPIs. Consider the table on the next page titled "Aspect of Software Delivery Performance" when measuring the performance of your team.

DevOps Metrics

Metrics	
Lead time for changes[6]	Average time it takes from idea, feature request, or backlog item to a successful deployment in the productive system environment
Deployment frequency	Number of deployments for an application in a given time frame
Change failure rate	Percentage of changes that led to service degradation or issues
Mean time to recover (MTTR; also known as time to restore service)	How long it takes to recover from a failed deployment in production
Backlog burn-down	Number of backlog items submitted versus implemented per time frame
Number of bugs	Number of confirmed software issues
Number of security vulnerabilities	Number of security vulnerabilities detected through code checks and number of security vulnerabilities detected in production
Number of incidents	Number of incidents raised
Number of users	Number of users accessing the application in a given time frame

According to *Accelerate: State of DevOps 2018: Strategies for a New Economy*, a report by DevOps Research & Assessment (DORA), the highest performers manage to bring new code changes and code corrections into production within an hour, effectively delivering multiple code deployments a day.

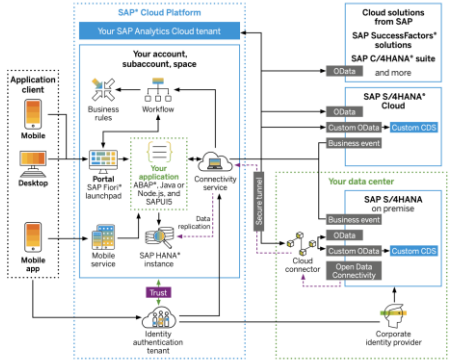
6. Notice the extended definition of the metric.

PART FOUR – KEY ARCHITECTURAL PATTERNS AND DECISION MATRIX

PATTERN #1: CLOUD APPLICATION

With this pattern, you create a new cloud application on SAP Cloud Platform and integrate it to SAP and third-party, on-premise, and cloud products based on standard and custom APIs (see Figure 18).

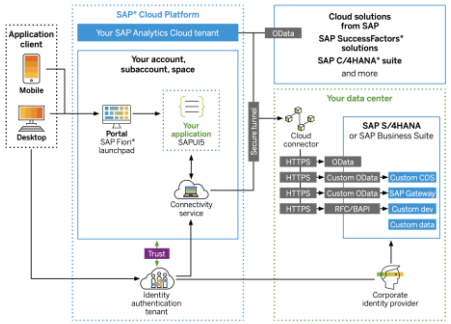
Figure 18: Cloud Application Pattern



PATTERN #2: HYBRID APPLICATION

With this pattern, you improve UX by exposing your application's user interface through SAP Cloud Platform while the application logic and data reside in your SAP S/4HANA on premise or on another back-end system (see Figure 19).

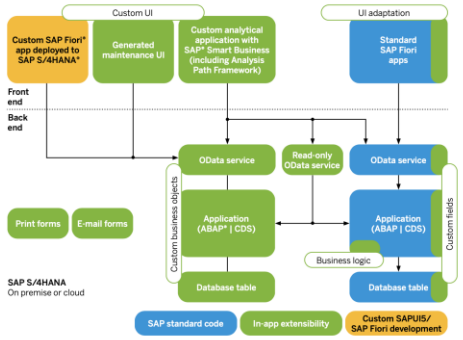
Figure 19: Hybrid Application Pattern



PATTERN #3: IN-APP EXTENSIBILITY

With this pattern, you consume the in-app extensibility capabilities of SAP S/4HANA to create custom extensions with key user tools and embedded analytics (see Figure 20).

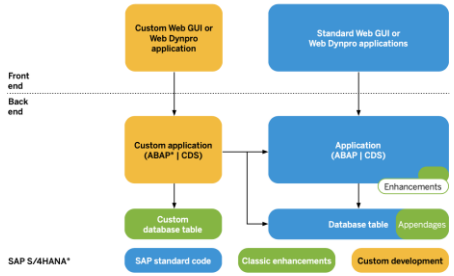
Figure 20: In-App Extensibility Pattern



PATTERN #4: CLASSICAL EXTENSION

With this pattern, you reuse the enhancement techniques and custom development capabilities from your SAP ERP Central Component system (see Figure 21).

Figure 21: Classical Extension Pattern



DECISION MATRIX

The purpose of the decision matrix in the table below is to help your architects to choose the best-fitting architecture pattern when evaluating new business requirements or deciding on the future of the existing custom developments in SAP ERP.

		Pattern			
	Requirements Toward Custom Functionality	Cloud application	Hybrid application	In-App extensibility	Classical extension
Users and UX	Involve consumers of the corporate products and services (B2C) (for example, service orders, master data self-services, catalogs, Web shops, mobile access)	+	+		
	Involve business partners (B2B) to enable direct collaboration (for example, order review and approval, service or good receipt, quality control, delivery checkpoints)	+	+		
	Involve employees (B2E) who otherwise have no access to the business solution (for example, outsourced workers, leased workers, mobile workers)	+	+		
	Adapt existing UIs based on the SAP Fiori® UX – Add, hide, move, or regroup fields on screen, add custom fields, change label texts			+	(+)[a]
	Improve UX by redesigning the UI for existing applications (for example, simplifying data-entry screens, dropping screens that are not required, auto-filling fields, and enabling speech-to-text, translation, and localization functionality)	(+)[b]	+		
	Open-source components and freestyle UI (non-SAPUI5/SAP Fiori)	+	+		
	Mobile native capabilities (for example, access to microphone, camera, GEO location, and so on)	+	+		
Data	Stand-alone application based on own data model with occasional consumption of standard data in SAP S/4HANA®	+		(+)[c]	
	Analytical application consuming standard and custom data residing in SAP S/4HANA		(+)[d]	+	
	Analytical application consuming data distributed across multiple SAP® and non-SAP systems (for example, data lake)	+			
	Transactional data consistency – Custom data is changed in a single database transaction with core data in the back end.		+	+	+
Features	Agility and independence on the back-end lifecycle	+			
	Reactive (event-based) process extensions and custom workflows	+			
	Use of SAP and third-party cloud services (for example, machine learning solutions from SAP, SAP Localization Hub services, tax services, Google Maps, and so on)	+			
	Application with unpredictable or largely varying usage and resource consumption (scalability and elasticity)	+			

[a] UI adaptation for SAP GUI for HTML (Web GUI) and Web Dynpro applications require modifications and classic enhancement techniques.
[b] Data entry automation, for instance, might require application logic to be implemented on SAP Cloud Platform.
[c] Simple use cases can be easily implemented with the help of in-app extensibility.
[d] Use the SAP Analytics Cloud solution for data analysis while building a custom API for data provisioning.

Follow all of SAP

www.sap.com/contactsap

or S4MOVE@sap.com

THE BEST RUN SAP®