

# **SOFTWARE AG UPGRADE ASSESSMENT AND READINESS GUIDE**

## CONTENTS

<b>Overview</b>	<b>3</b>
<b>Migration Components and Approach</b>	<b>3</b>
<b>Assessing Your Project</b>	<b>4</b>
<b>Safely Migrating Your Environments</b>	<b>7</b>
<b>Phases of Migration</b>	<b>8</b>
<b>Preparing for Migration</b>	<b>8</b>
<b>Testing</b>	<b>8</b>
<b>Work Breakdown</b>	<b>9</b>
<b>Product Support Lifecycle</b>	<b>10</b>

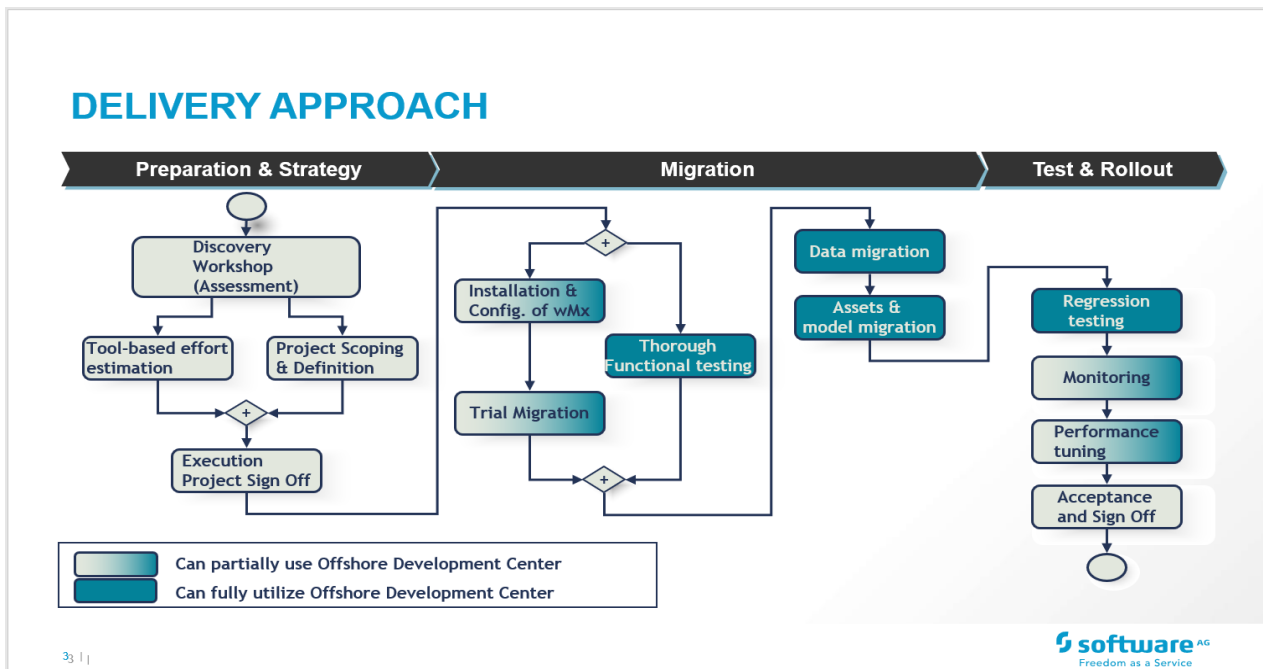
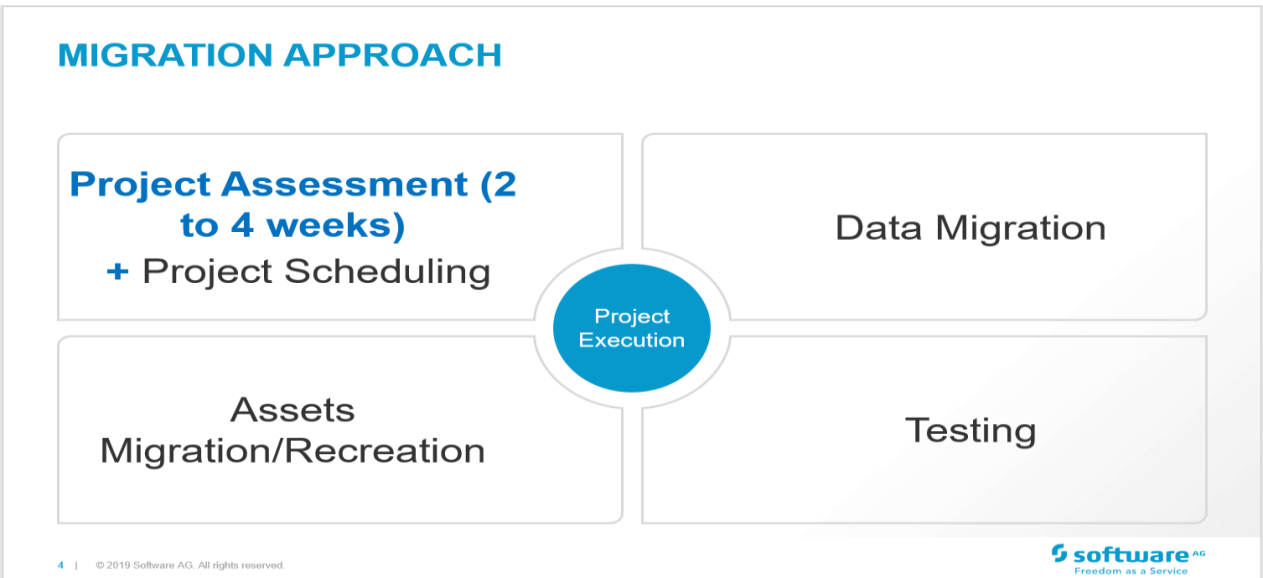
## Overview

A well-strategized migration approach can make upgrade an efficient journey. This paper can be used as the general foundation for a detailed project plan.

## Migration Components and Approach

Timelines will vary depending on the complexity and depth of the existing platform.

Sample Migration Plan	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	W18	W19	W20	W21	W22	W23
	Month 1			Month 2			Month 3			Month 4			Month 5			Month 6							
<b>Sample Upgrade Plan</b>																							
Migration Strategy and Planning	█	█	█																				
Installation and Configuration				█	█	█	█	█															
Migration						█	█	█	█														
Security Review and Configuration									█	█	█												
Unit Regression testing												█	█										
SIT promotion and support														█	█	█							
UAT Support																	█	█	█				
Go Live planning and training																							
Post Go live support																							



## Assessing Your Project

Assessment of middleware requires the following:

- Understanding services deployed and features built inside the middleware.
- External constructs of middleware, including the number of interfaces and end point applications.
- Internal constructs of middleware, including configurations, rules, transformation logic, flows, pattern usage, and custom code.
- Supporting information such as resource profiles in the current landscape, SLAs, quality management, governance, and non- functional requirements.
- Whether external clients connect to your applications. If so, typically host name and port numbers should remain the same, which requires planning.
- Understanding the complete landscape (number of environments, RDBMS, active developments, dependencies, and so on).

Assessment of the migration project itself requires the following:

- Decisions around project timelines such as when to migrate, migration timeframe, timelines for testing, and support from Software AG.
- Decisions around project cost and effort.
- Maximum permissible downtime for your product environment.
- Cutover period that is possible for your product environment on migration day.

Below is a typical upgrade assessment questionnaire used by Software AG.

Objectives of Migration
What are your reasons for upgrade (technical, functional, or strategic)?
What are your decision factors?
Technology-related (e.g., database migration required)
Landscape-related (e.g., production downtime)
Interface-related (many complex product interfaces)
Enterprise deployment norms (global architecture model for all PROD applications)
Are there any existing issues with current environment? If yes, provide details.
How many products are currently deployed on how many logical servers?
Do you prefer a big-bang or stage-wise migration approach?
Would this be an AS IS migration or are changes required to the current deployment architecture?
Is an OS change or upgrade required? If yes, please provide the target OS name and version.
Are you interested in consolidating or separating existing logical servers?
Are you interested in discussing a new deployment strategy for the new Software AG platform?

Would you like capacity recommendations to be built into your migration or are you confident that current systems will be able to support future growth?

Are there any new features that you would like to introduce on the new platform after migration?

Has your system been migrated in the past? If so, from what?

Is the system being accessed from multiple countries?

Do you use any custom adapters?

How many application and projects need to be migrated?

What is the primary OS (Windows, Unix) that is being used?

What is the primary database that is being used?

What are the current daily, monthly, and yearly volumes in PROD?

What is your current data growth per month?

What was your percentage transaction growth over the past 3 years?

### **Interfaces**

Do you interface to any external systems? If so, please list them.

Briefly describe the underlying technologies used in interfacing.

Are any external jars used? Will these also need to be migrated?

Have you developed any custom applications using DSP pages?

### **Versioning**

Is version management done for your project code? If yes, list the assets that are version controlled.

Which version management tool are you using?

Are project versions being prepared locally or using a centralized server?

### **Code Deployment**

Is there any procedure and tool used for code movement across environment? If yes, provide details.

Are any automated deployment scripts being used for code deployment?

Are any code releases or projects planned during the migration project or in process?

<b>Testing</b>
Are there any business-critical flows?
Are there any volume-critical flows?
Does Software AG need to perform functional testing of business flows or can that be handled by you?
Should Software AG perform load or workload testing for volume critical flows?
Which environment in migration will have all end system connectivity & interfaces?
On which environment can above testing be performed? This environment needs to be connected to end systems used in flows.
Do you have systems to test functional aspects after migration? Will you cover this testing yourself?
On what parameters would the acceptance plan of migration plan be based?

<b>Project Timelines</b>
When are you interested in starting your migration project?
What is your proposed time frame for migration?
Do you prefer a big bang migration or staggered approach?
Can existing services data remain on your current platform or do they also need to be migrated to the Software AG platform?
Indicate number of days that you need Software AG to support integration testing
Indicate number of days that you need Software AG to support UAT testing
Do you have any post Go-live support requirements? If yes, provide number of days.

<b>Migration Preparedness</b>
Has any migration assessment been done before?
How many environments can be made available to Software AG for the migration?
Would it be feasible to capture your services pipeline on PROD?
Are there any application databases separate from product databases?
Do you have any other on-going projects that will impact the migration?
Is active development being carried in any of those projects?

Does the Software AG platform need to be available on a 24x7 basis?
What is your maximum permissible downtime on PROD?
Is a code freeze possible during the period when migration is being performed?
What is the cutover period that is possible on PROD on migration day?

<b>Availability</b>
Have you configured high availability? (hardware, RDBMS, cluster)
What type of clustering solution do you use (e.g., Veritas, VMware, ESX)
Do you have a disaster recovery methodology?
Describe other plans or concerns regarding the failover implementation.

## Safely Migrating Your Environments

The most important question is the number of environments you have. It is quite common to have a development, test, and production environment. The first step could be to migrate your development environment, and to note the steps you are taking. As you go along, perform basic sanity checks and to make sure everything looks perfect.

When you think that you have documented that procedure, run through the procedure in your test environment as if you are doing the migration in your production environment. Perform detailed testing to make sure that everything works fine in your new environment. If you encounter issues, adjust your documented procedure if necessary. If you feel that the issue will have a big impact, you can choose to go back and rerun the tests in your test environment and then update the procedure and make sure that it will run smoothly when you apply it to your production environment.

Documenting the procedure will also help you determine how long will it take to perform the actual migration so you can plan for the downtime window. Most organizations have a maintenance window during which to migrate, optimally a weekend, but sometimes only overnight (6-8 hours). If you have less time than that, you will need more careful planning and preparation, particularly when you switch from one environment to another.

If you have a larger landscape, it might become impractical to migrate all servers within that landscape within a single maintenance window. In this case you can use phased migration. In phase migration, you try to migrate one set of servers that comprises one landscape, prepare the checklist of findings, and apply the same procedure to all other sets of servers one by one.

In summary:

- Create a checklist that contains the following
  - Assets in the old environment
  - Known issues in the old environment
  - Performance metrics in the old environment
- First migrate dev environment
  - Document the steps that you perform in form of a checklist
  - Perform basic testing

- Test documented migration procedures on Test environment
  - Perform detailed testing
  - Adjust the document if necessary
- Finally perform the migration procedure on the production environment

## Phases of Migration

- Pre-requisites
- Preparation for Migration
- Cut over plan preparation
- Development of target environment
- Migration of QA
  - Code Migration
  - Asset Migration
  - Data Migration
- Migration of Pre-Prod
- Code Merge
- System Integration Testing Support
- UAT Support
- Production Roll Out Planning and Go-Live
- Post Go Live Support

## Preparing for Migration

- Document the steps that you do in the form of a checklist.
- If working with short deadlines, consider de-prioritizing interfaces that are less critical or those that can be mitigated by manual processes.
- To begin with, we look for relatively simple yet meaningful interfaces that can help gain comfort with the new platform.
- Make sure all tasks, processes, schedulers, services are completed/suspended before upgrading/shutting down.
- Assets created using same framework/technology can be easily migrated/ deployed to the Software AG platform.
- Some assets may require recreation to align with the Software AG Platform.

## Testing

The most vital and biggest time-consuming task of the migration process is the testing.

- Make sure that the migrated environment is correctly configured and there are no configuration issues.
- Perform Unit testing.
- Perform Functional testing.
- Perform User Acceptance testing.

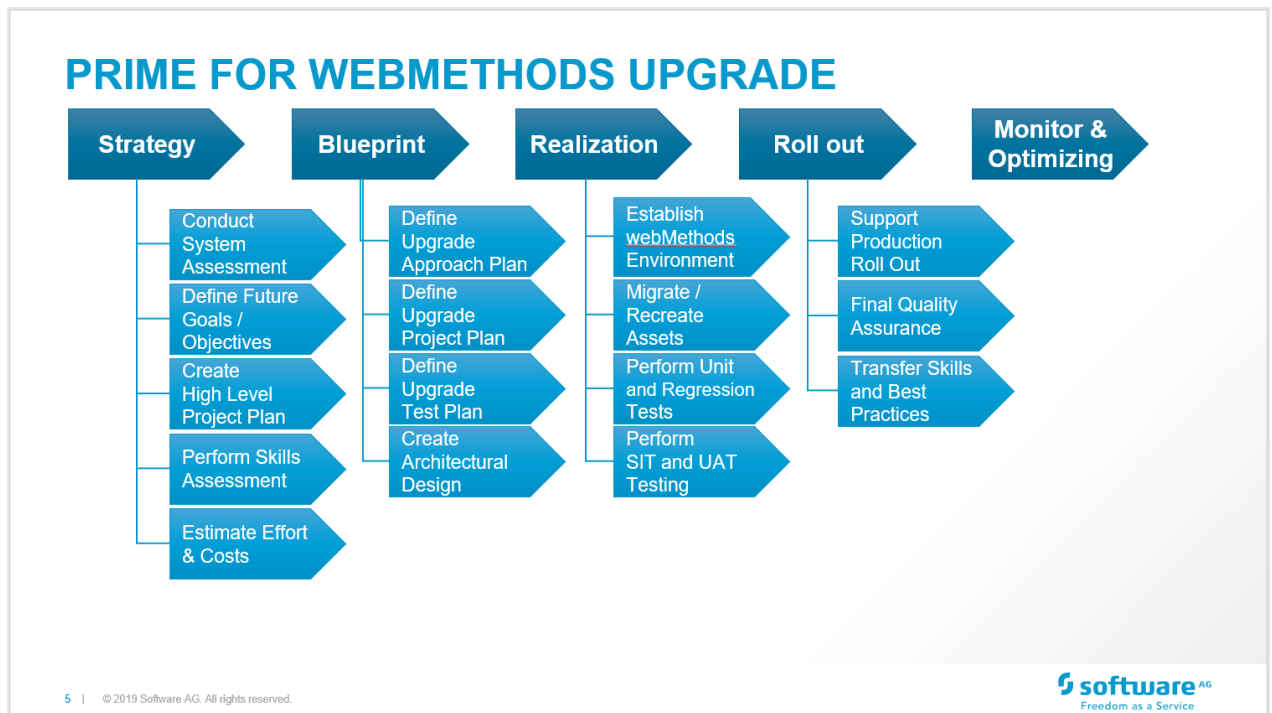


- Perform E2E testing and make sure there are no integration problems.
- Perform security and performance testing and make sure the performance test results match with older metrics or should be even better.

Listed below are some example of very basic testing tasks. Prepare a list of all the testing activities.

Task	Type
Stop any scheduled services.	Pre migration
Remove all user access. Lock the database schemas.	Pre migration
Start new servers.	Post Migration
Check and enable scheduled services.	Post Migration
Make sure all connections point to the right database.	Post Migration
Create all required directories on the file system.	Post Migration
Make sure all script directories have the necessary scripts and runtime privileges.	Post Migration
Check your list of assets.	Post Migration
Make sure all users are set up with correct privileges.	Post Migration

## Work Breakdown



## Product Support Lifecycle

