Freedom as a Service

SOFTWARE AG UPGRADE PREPARATION AND PLANNING GUIDE

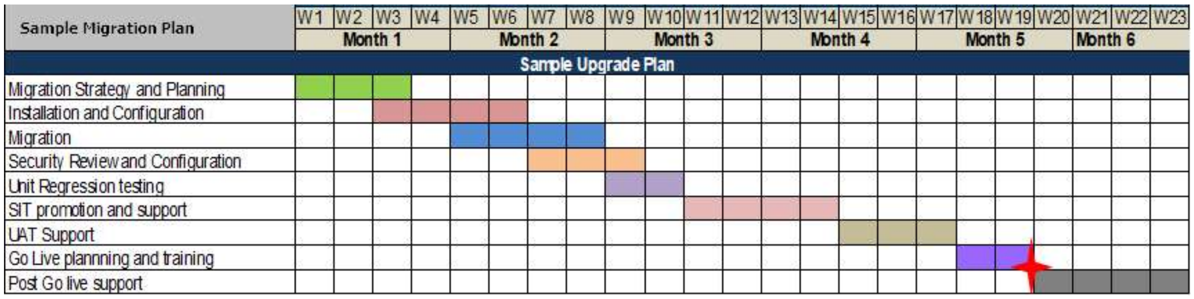
CONTENTS

Overview

A well-strategized migration approach can make upgrade an efficient journey. This guide can help you form the general foundation for a detailed project plan.

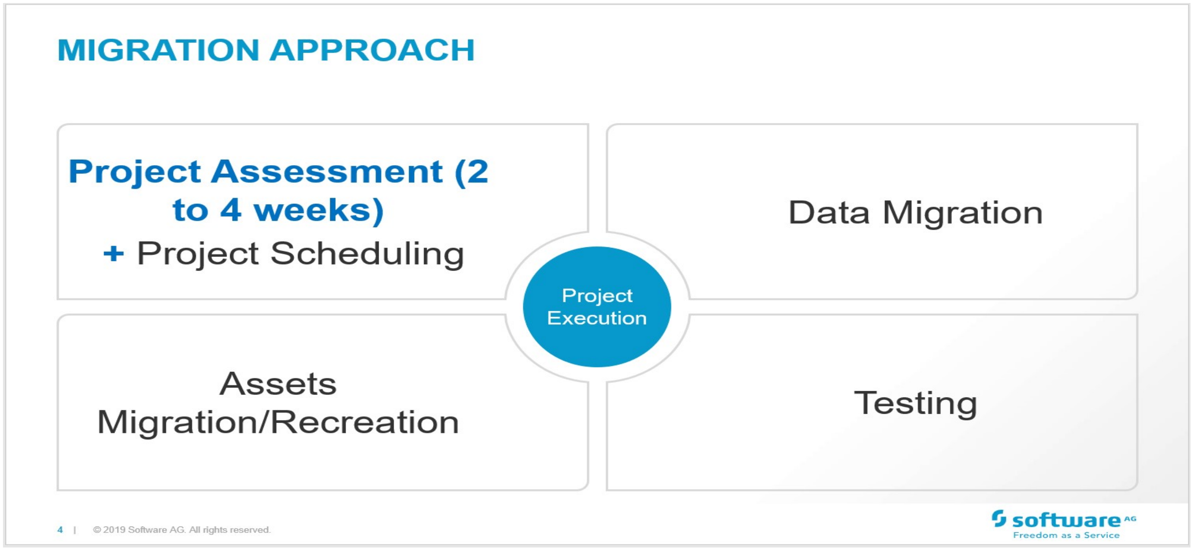
Migration Approach and Phases

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



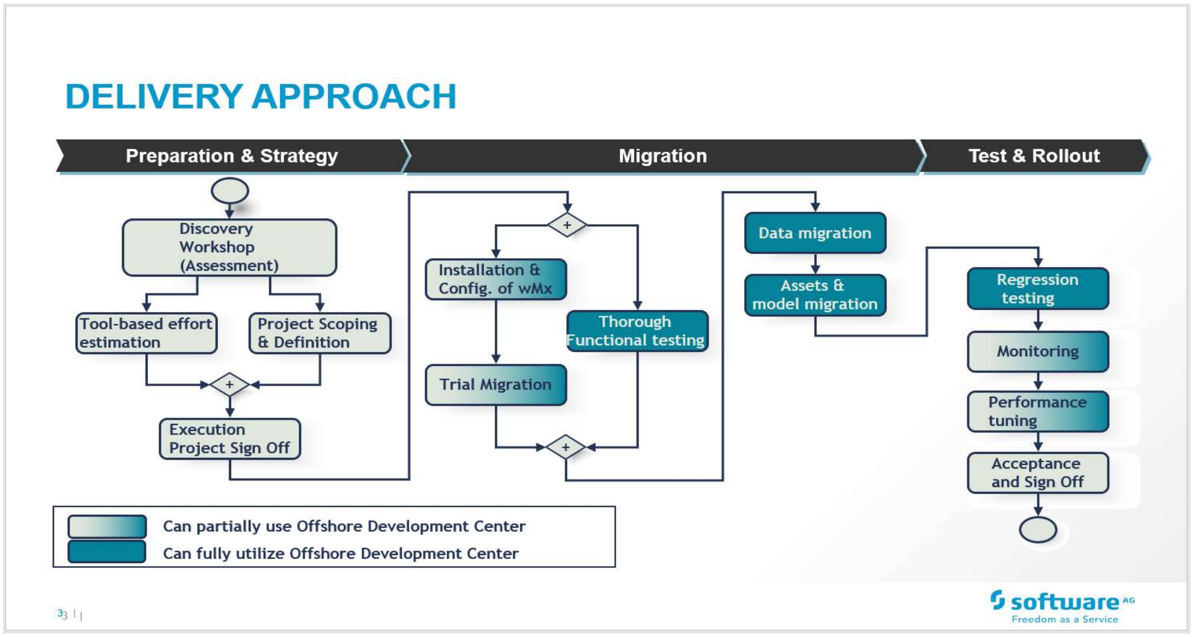
Sample Migration Plan	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	W18	W19	W20	W21	W22	W23
	Month 1				Month 2				Month 3				Month 4				Month 5				Month 6		
Sample Upgrade Plan																							
Migration Strategy and Planning	🟩	🟩	🟩																				
Installation and Configuration			🟥	🟥	🟥	🟥																	
Migration					🟦	🟦	🟦	🟦															
Security Review and Configuration							🟧	🟧	🟧														
Unit Regression testing									🟪	🟪													
SIT promotion and support											🟥	🟥	🟥	🟥									
UAT Support															🟫	🟫	🟫						
Go Live plannning and training																		🟪	🟪				
Post Go live support																				⬛	⬛	⬛	⬛

- 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



MIGRATION APPROACH

Project Assessment (2 to 4 weeks)
+ Project Scheduling
Data Migration
Project Execution
Assets Migration/Recreation
Testing
4 | © 2019 Software AG. All rights reserved.
software AG Freedom as a Service



DELIVERY APPROACH

Preparation & Strategy
Migration
Test & Rollout
Discovery Workshop (Assessment)
Tool-based effort estimation
Project Scoping & Definition
Execution Project Sign Off
Installation & Config. of wMx
Thorough Functional testing
Trial Migration
Data migration
Assets & model migration
Regression testing
Monitoring
Performance tuning
Acceptance and Sign Off
Can partially use Offshore Development Center
Can fully utilize Offshore Development Center
33

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).
- Assets created using same framework/technology can be easily migrated/ deployed to the Software AG platform, but some assets may require recreation to align with the Software AG Platform.

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?

Testing
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?

Project Timelines
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.

Migration Preparedness
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?

Availability
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.

Preparing for Migration

- Read Software AG-provided instructions around upgrade in their entirety. You can find the following sources of information in the *Complete Installation and Upgrade Information for Software AG Products* webhelp:
  - Read *Upgrading Software AG Products* in its entirety so you are familiar with all tasks you will need to perform.
  - Read the product release notes. The release notes provide information on new functionality. Read the information for every release for your old release + 1 through the new release.
  - Read the product readmes for your old release+1 through the new release, including the readme for the Software AG Infrastructure. For example, if your old release is 9.9, read the information releases 9.10 through 10.7. The product readmes contain this information:
    - Critical information and known and resolved issues for your products.
    - Changes to product behavior, services, parameters, properties, and APIs. Such changes can include additions, changes, deprecations, and removals. This information is especially important because you might need to modify product files or assets after migration to accommodate the changes.
  - Check *System Requirements for Software AG Products*. If the RDBMS version you are using is not supported by your new products, you will need to upgrade to a supported RDBMS version.
- If you are upgrading Integration Server, go to the Software AG Tech Community and download the Pre-Upgrade Analyzer at https://tech.forums.softwareag.com/t/integration-server-pre-upgrade-analyzer/238942. This tool analyzes your old Integration Server installation and generates reports that list the following:

    - Custom services using built-in services that were deprecated, removed, or changed in a later release. Full details about each deprecation, removal, or change is available in the Integration Server readme.
    - Custom packages using custom services or documents that do not exist.
    - Custom assets that are not used by any of your existing integrations.
    - Custom services using Integration Server debug services (pub.flow:debugLog, pub.flow:savePipelineToFile).
- As you progress through the upgrade, apply newly released fixes for your products. Read the fix readmes to become aware of the issues they resolve and any changes in behavior.
- Go to the Knowledge Center on the Software AG Empower Product Support Website and read tech articles (for example, KB 1788395). Use keywords such as upgrade to search for relevant tech articles, and search for your old release, your new release, and all releases in between.
- Set up operating guidelines that cover normal project management requirements across all affected groups in your organization.
- Define an approach to source control and set up release management tools and procedures that support the migration of data into target environments.
- Account for externals to your Software AG environment, such as clients that communicate with your Software AG products and that might need endpoint URL, host name, or IP address updates.
- As part of upgrading, take a baseline of the old environment and use that baseline to perform the upgrade procedure. Software AG recommends a code freeze while you are migrating the baseline to the new environment.
- Many upgrade issues are caused by forgotten environmental settings. Plan, document, and test these thoroughly.
- For complex projects that require extensive system testing, extended code freeze might not be possible. In this case, define a change management strategy to track code changes that occur in the old environment. After testing is complete on the baseline, this will enable you to merge the code changes into the baseline and re-test the new baseline.
- Look for relatively simple yet meaningful interfaces that can help gain comfort with the new platform. If working with short deadlines, consider de-prioritizing interfaces that are less critical or those that can be mitigated by manual processes.
- If you need help creating your planning approach, contact Software AG Professional Services. You can also go to the Software AG TECHcommunity and find tips and tricks, videos, blog posts, articles, and other information in the upgrade wiki (Wiki > webMethods > Upgrade).

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 in the form of a checklist. 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 lists assets, known issues, and performance metrics in the old environment
- Migrate dev environment, document the steps that you perform in form of a checklist, and perform basic testing
- Test documented migration procedures on test environment and adjust the checklist as necessary
- Finally perform the migration procedure on the production environment

Testing

The most vital and most time-consuming task of the migration process is the testing. Software AG strongly recommends that you upgrade in a controlled test environment and test that environment for proper operation before promoting to your production environment. Testing can include the tasks listed below.

- Provision new environments to support the upgrade. Environments in which you conduct testing should mirror the production environment.
- Obtain test and performance data from the source environment and establish baseline results.
- Many upgrade issues are caused by forgotten environmental settings. Plan, document, and test these thoroughly.
- Identify test tools for unit, functional, and non-functional regression testing, and for performance and security testing.
- Define acceptance and sign off criteria.
- Automate a core set of regression tests for the upgrade. Software AG recommends focusing your tests most heavily on product areas that have undergone the most change. The product readmes and release notes provide all product change information.
- If upgrading to new machines, make sure connections and firewalls are open to back-end systems before starting any system testing.
- Use an issue management system to track defects and issues.
- If code freeze of the old environment during test is not possible, merge code changes from the old environment into the baseline and re-test the new baseline.
- Make sure the migrated environment is correctly configured and there are no configuration issues.
- Perform unit testing.
- Perform functional testing.
- Perform user acceptance testing.

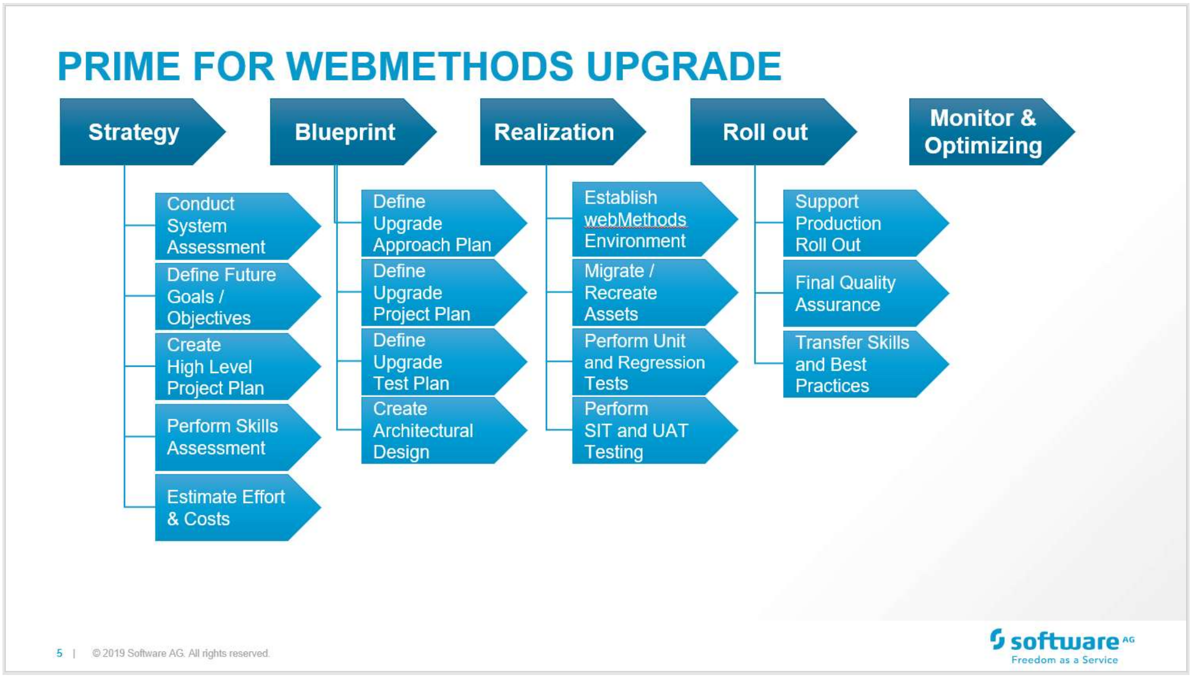
- Perform E2E testing and make sure there are no integration problems.
- Perform security and performance testing and make sure the performance test results match or exceed older metrics.

If you need help creating your testing approach, see the *Upgrade Testing Guidelines* in *Installation and Upgrade Information for Software AG Products* or contact Software AG Professional Services.

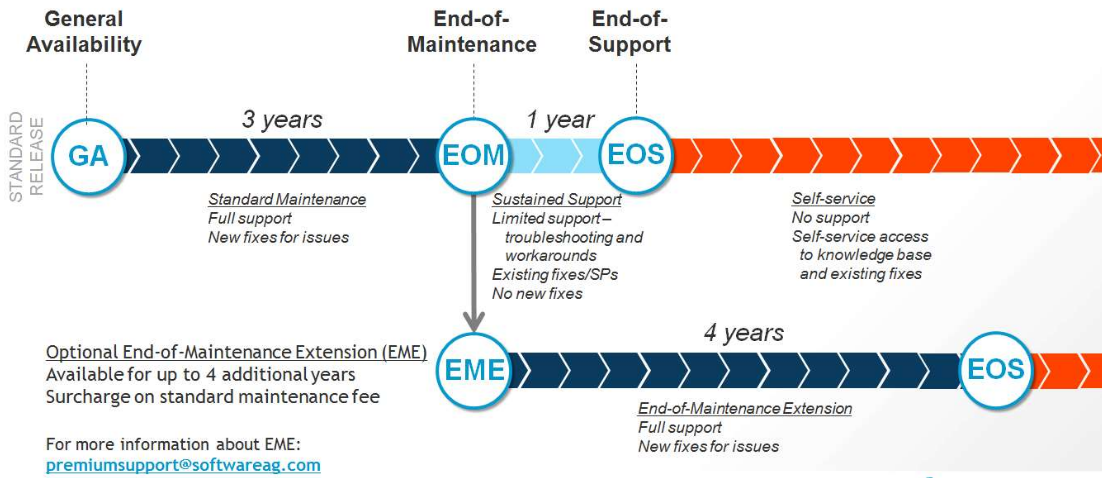
Listed below are some example of very basic testing tasks. Prepare a list of all the testing activities.

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


PRIME FOR WEBMETHODS UPGRADE
Strategy
Conduct System Assessment
Define Future Goals / Objectives
Create High Level Project Plan
Perform Skills Assessment
Estimate Effort & Costs
Blueprint
Define Upgrade Approach Plan
Define Upgrade Project Plan
Define Upgrade Test Plan
Create Architectural Design
Realization
Establish webMethods Environment
Migrate / Recreate Assets
Perform Unit and Regression Tests
Perform SIT and UAT Testing
Roll out
Support Production Roll Out
Final Quality Assurance
Transfer Skills and Best Practices
Monitor & Optimizing
5 | © 2019 Software AG. All rights reserved.
software AG Freedom as a Service

Product Support Lifecycle


General Availability
End-of-Maintenance
End-of-Support
STANDARD RELEASE
GA
3 years
EOM
1 year
EOS
Standard Maintenance
Full support
New fixes for issues
Sustained Support
Limited support – troubleshooting and workarounds
Existing fixes/SPs
No new fixes
Self-service
No support
Self-service access to knowledge base and existing fixes
Optional End-of-Maintenance Extension (EME)
Available for up to 4 additional years
Surcharge on standard maintenance fee
For more information about EME:
premiumsupport@softwareag.com
EME
4 years
EOS
End-of-Maintenance Extension
Full support
New fixes for issues