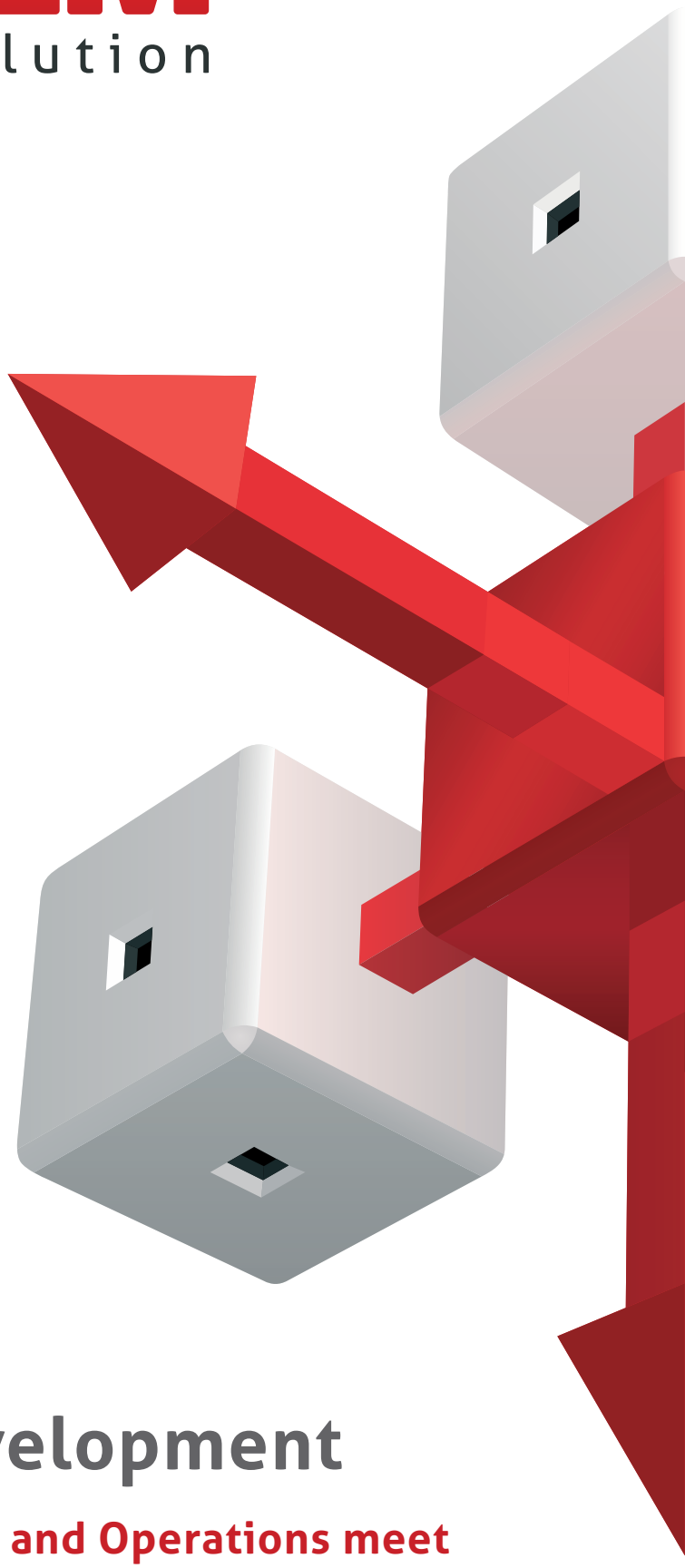
IKANALM

ALM through evolution



SAP and IKAN Development

Where Development, Testing and Operations meet

Table of contents

Executive summary

Application Lifecycle Management (ALM) is a key discipline aiming at managing the whole software development process from requirements through deployment, involving Business Analysis, Project Management, Development, Quality Assurance (QA) and Testing and, finally, the delivery of the results to Operations.

The trend to globalization (distributed teams), the evolving methodologies (from Waterfall to Agile), the omnipresent budget restrictions and the necessity to deliver more and faster, increase the need to automate the complete ALM process in order to:

- make it enforceable,
- improve communication between all stakeholders,
- enhance close collaboration and teamwork,
- more efficiently allocate the appropriate resources,
- deliver faster and with higher quality
- and, as a result of the previous, to reduce the overall cost and increase the efficiency.

SAP provides a set of standards for ALM. Its ALM portfolio consists of processes, tools, services, and best practices, to manage SAP and non-SAP solutions, throughout the entire application lifecycle. SAP follows the IT infrastructure library (ITIL) guidelines, which define six phases for the Application Lifecycle.

According to SAP, two things are key to have a good working ALM solution: the right infrastructure, and a clear definition of the processes, including all necessary activities, responsibilities, and service levels. The infrastructure is provided by **SAP Solution Manager** as a collaboration platform. It provides all functions required (provided either by SAP Solution Manager itself or by integrated tools) via work centers. SAP has also leveraged the experiences of their customers, together with these of their own application lifecycle management experts, to create best-practices.

IKAN ALM – as an integrated tool on the SAP ABAP and Java Application Servers – concentrates on what happens after the actual software development stage: the Build process, Testing and the Deployment into Production. More specifically, IKAN ALM offers the following functions: commit to versioning (if not standard available), build process, creation and management of the lifecycle (Development, Test & Acceptance and Production) and an approval process.

IKAN ALM respects the ALM process as designed by SAP whilst adding value to the Build, Test and Deploy steps by adding specific features and functions. IKAN ALM fully uses the SAP Solution Manager and the SAP Transport Manager and CTS+.

This White Paper targets all parties interested in ALM Build and Deploy solutions for SAP products, be it executives, technical managers, software architects, operations people or developers.

Problem statement

- SAP lacks a flexible versioning system for ABAP. Change requests are not linked with Versioning, Build or Deploy.
- Dependency management in SAP is a complicated, non-automated process. Between ABAP and Java components, there is no dependency management: SAP does not guarantee a connection between versions of ABAP and Java added to the same transport request.
- There is no integrated Build Process for managing both ABAP and Java components. A lot of manual intervention is needed, for example if you want to move Java components with ABAP, the Java components need to be added manually.
- SAP has an automated Deploy mechanism (transfer from one system to another), but it is a complex task to configure it and, as mentioned in the previous point, before transporting to another system a lot of manual intervention is needed (The Java Developer must build the package and either he or the Release Manager must add it to the transport request, after which it can be transported to the next system).

Solution Description

The integration of IKAN ALM with SAP ABAP and Java application servers:

- provides a solution for the integration of source control for ABAP with Subversion,
- allows to define dependencies between ABAP and Java code,
- ensures a fully integrated build and deploy process,
- has the possibility to establish a link with any change request system using the SAP Change and Transport System (CTS).

Versioning

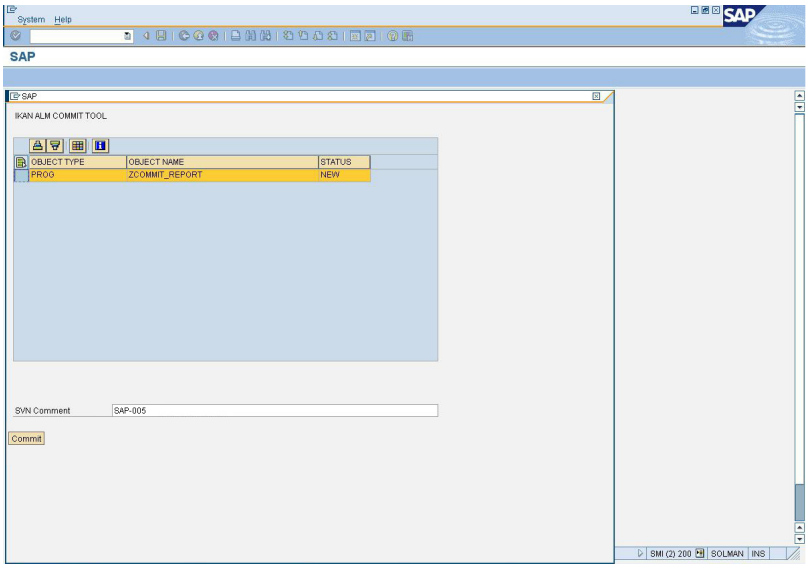
IKAN ALM offers source control integration for ABAP with Subversion: **the IKAN ALM ABAP Versioning Program**. As a consequence, ABAP components can as easily be versioned as NetWeaver Java components. In addition, IKAN ALM allows you to integrate any other language, such as C or COBOL, with ABAP or JAVA.

This facilitates the ALM process and adds value to SAP Solution Manager which itself does not dispose of elaborated source control facilities.

Example:

An ABAP Developer will use the standard ABAP Editor, and, through the IKAN ALM ABAP versioning program, the ABAP source code will be committed to Subversion. The NetWeaver Development Studio for JAVA has a standard, integrated Subversion client: all JAVA source code will be versioned in Subversion.

Thus, for both ABAP and JAVA, source code can be committed automatically from the respective IDEs into Subversion. All code, JAVA and ABAP, will reside in one place and you will enjoy all benefits of the Subversion versioning system.



Commit ABAP code to Subversion.

When committing, Developers can also add comments as to explain which Request, Issue or Defect has been solved by his code. IKAN ALM will parse those comment statements to update the Issue or Defect Tracking System.

Non-SAP objects can also be versioned by using the standard Subversion functionality.

Dependencies

Moreover, IKAN ALM allows defining Dependencies between the ABAP and JAVA code, which assures that both are always kept synchronized.

Suppose you have an ABAP and JAVA development. With IKAN ALM you can define a project A for ABAP and a project J for JAVA, where you indicate that the JAVA project depends on the ABAP project. By defining such a dependency, you are sure that, when creating a Transport Request, both projects are transported together and this to all systems in the System Landscape.

Once a dependency has been defined, it will always be available.

Integrated Build and Deploy Process

IKAN ALM offers a centralized and therefore easy-to-manage Build/Deploy Process.

Today in SAP:

- For ABAP, there is no available Build/Deploy Process. Instead a Transport Request is generated. The disadvantage is that you cannot see or track the changes in the Transport Request.

- For Java, there are two different ways to build. The first way is to do the Build locally, on the developer's machine and then do an import in the target server. An alternative is to use the NetWeaver development infrastructure: Java developers add their code and the Release Manager will initiate the Build process.

IKAN ALM adds value to the SAP environment by tracking both the ABAP and JAVA changes and by synchronizing both ABAP and JAVA changes in one Transport Request.

IKAN ALM offers automated and centralized Release Management for a whole project.

Ticketing

IKAN ALM has an Issue Tracking System Plugin. Through that Plugin, any Change Request system can be linked to IKAN ALM. This allows you to check which change solves which Change Request.

Short description of IKAN ALM

Lifecycle

IKAN ALM offers the possibility to implement a Lifecycle. A Life-Cycle defines the logical steps of the ALM processes. Such a logical step is called a Level in IKAN ALM. A Level consists of one (or more) Build and/or Deploy Environment(s) which are physical environments.

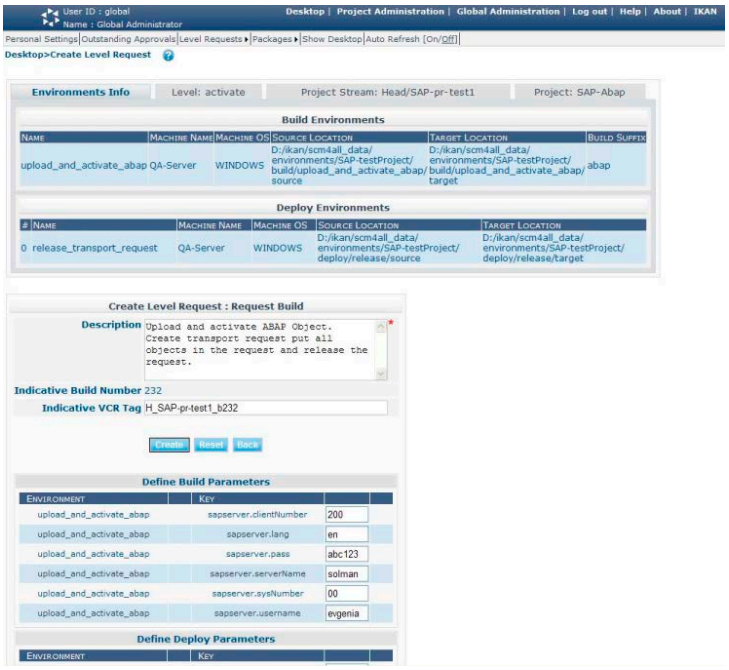
Example of an IKAN ALM lifecycle

IKAN ALM has three Level Types: Build, Test and Production.

Build Process

Once the code has been committed to or referenced in the versioning system, IKAN ALM can DEPLOY and ACTIVATE the respective ABAP objects and JAVA code or NetWeaver projects.

The IKAN ALM Build process supports both traditional and Waterfall development methods. Builds can be launched automatically at predefined moments (daily, weekly, ...) or will be launched automatically every time a COMMIT occurs in Subversion. A Build can also be launched manually by an authorized user.



Example of an IKAN ALM Build to activate ABAP code

IKAN ALM offers a centralized and fully automated activation and build process for both ABAP and JAVA components, and also takes care of the synchronization of the ABAP and JAVA components.

The IKAN ALM Build process will also create and release SAP Transport requests and IKAN ALM is fully integrated with SAP TMS and CTS+.

Non-SAP objects will also be integrated in the generated Transport Requests.

Test Process

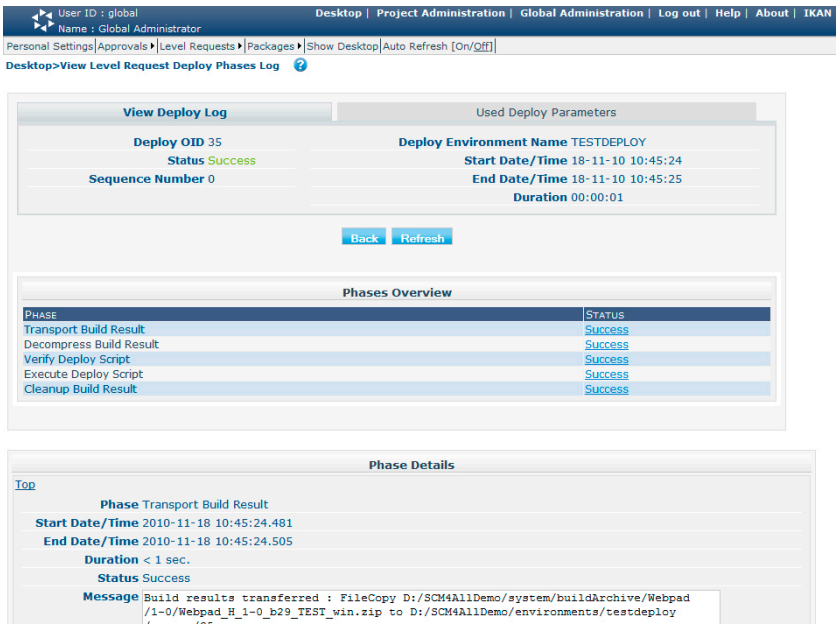
With IKAN ALM you can define as many Test Levels and Environments as you want. Deploys from one Level to another can be approval-based. This to ensure that there is always a trace of who, when and why a Test Level has been approved and deployed to a next Level.

Note: IKAN ALM is also fully integrated with HP ALM 11.0 (formerly called HP Quality Center). For more information: http://www.ikanalm.com/whitepapers/HP_IKAN_integration.pdf

Deploy Process

The Deploy step moves a project to the next Level. The Deploy gives you a centrally controlled, automated Release Management for a complete project. The delivery to all systems in a project landscape is fully automated and all relations between the different projects and sources (particularly between JAVA and ABAP) are guaranteed.

IKAN ALM not only has the ability to deploy new releases, it can also revert to previous versions (rollback).



Example of an IKAN ALM Deploy

Benefits

The integration of SAP with IKAN ALM provides the following benefits.

Benefits for SAP customers:

- better control over ABAP and JAVA source code and the source code dependencies
- versioning of both ABAP and JAVA code
- automated creation and execution of SAP Transports
- Integration with HPALM 11.0 for testing : functional test results will be started after each build and automatically reported to the HP ALM testing module

Benefits for IKAN ALM customers:

- fully automated SAP solution
- test scheduling and execution for SAP

Summary/Conclusion

Combining SAP and IKAN ALM brings together the best of both worlds.

IKAN Development provides an integrated web-based Application Lifecycle Management (IKAN ALM) platform for both Agile and traditional software development teams.

It combines Continuous Integration and Lifecycle Management, offering a single point of control and delivering support for build and deploy processes (manually generated or automated), approval processes, release management and software lifecycles.

IKAN ALM tightly integrates with leading existing third-party versioning solutions (e.g. IBM® Rational® ClearCase®, Microsoft® Visual SourceSafe®, Serena® PVCS Version Manager, CVS, Subversion) and build and deploy tools (Make, Ant, NAnt, Maven 2), and also provides a defect tracking software plugin (Atlassian® JIRA®, ...). This results in being a unique cross-platform ALM solution.

SAP is a market-leader in enterprise application software. SAP helps companies of all sizes and industries run better. Founded in 1972, SAP (which stands for "Systems, Applications, and Products in Data Processing") has a rich history of innovation and growth as a true industry leader.

Today, SAP has sales and development locations in more than 50 countries worldwide. SAP applications and services enable more than 109,000 customers worldwide to operate profitably, adapt continuously, and grow sustainably.

IKAN Development together with SAP creates an integrated ALM solution which establishes an environment where developers, testers and operations resources seamlessly work together, each of them doing what he is best at without having to waste time in figuring out what he needs from another stakeholder or what he needs to deliver.

Our solution fully synchronizes all Development, Test, Build and Deploy activities, and will obviously lead to enhanced collaboration, higher quality, faster delivery times and reduced costs.

For more information

To know more, visit http://www.ikanalm.com

Contact IKAN Development at info@ikanalm.com

IKAN Development N.V.
Kardinaal Mercierplein 2
2800 Mechelen
Tel +32 15 797306
www.ikanalm.com
info@ikanalm.com