Replacing an EAI Platform with SAP PI: Report from a Real Project

Applies to:

SAP NetWeaver PI. For more information, visit the Service Bus-based Integration homepage.

Summary

This article is about a real EAI project in 2007/2008. The project order was to replace an existing EAI platform by SAP PI 7.0. The article reflects some observations as well as some continuative thoughts and conclusions.

Author: Thorsten Wilde

Company: SAP Deutschland AG & Co. KG

Created on: 16 December 2008

Author Bio

Thorsten Wilde studied computer sciences and started as a "classical" software developer working for different companies. In 1997 he had his first contact with ABAP and 2001 he joined SAP as an integration consultant. Thorsten Wilde has been working in numerous projects covering both pure integration topics and integration related ABAP development. Since 2004 he has been working with SAP PI.

Table of Contents

Motivation

This article is about a real EAI project in 2006/2007. The project order was to replace an existing EAI platform by SAP PI 7.0. The article reflects some observations as well as some continuative thoughts and conclusions.

About 400 interfaces had to be migrated. Furthermore there were dozens of interfaces required by ongoing integration projects which had to be implemented on SAP PI a priori. All in all approximately 600 interfaces had to be implemented. The majority (about 75%) was either SAP to non-SAP or between non-SAP systems. The most used technical PI adapters were RFC, file/ftp and JDBC. To minimize the risk of resource bottlenecks the customer decided to contract two implementation partners. Both partners employed onsite consultants as well as offshore teams in India. One of the partners was SAP Consulting. Their offshore team was staffed with SAP Mapping Factory consultants. Mapping Factory is an approach to minimize implementation costs particularly for EAI replacement projects or even PI Upgrade projects. For detailed information please refer to SAP's Mapping Factory homepage: http://www.sap.com/mappingfactory. The project lasted about 18 months.

Why EAI Replacement?

There are many reasons to replace an existing EAI platform. Some of them are

- **Lower Costs** – Saving license costs and cost of operation
- **EAI Consolidation** – Having one central EAI platform instead of multiple single or even local EAI platforms
- **EAI Standardization** – All interfaces are adhering to the same design rules and naming conventions
- **Re-Design** – Replacement of technical outdated integration approaches
- **Documentation Improvement** – Consistent documentation or availability documentation at all
- **Monitoring Improvement** – Older EAI platforms might lack features such as altering, message persistence, etc.

The main cause for the replacement was to reduce license and maintenance costs but the other benefits did influence the decision as well – although their prioritization might have been slightly different.

Basic Considerations

Solution Strategy

There are several possible solution strategies. In the described project it was decided to go for the 1:1 transfer of the existing interfaces. The intention was to keep the involvement of the business system owners as low as possible. Their involvement was supposed to be limited to the customer acceptance test shortly before go live. Another advantage of this approach was that the interfaces could be moved step by step from the old integration platform to SAP PI with limited planning overhead. Amongst others, this was important to avoid excessive alignment effort between the two implementation partners working at the same time without having the risk of negative interferences. Finally the 1:1 solution strategy allowed in case of any problems to go back to the old integration platform. This decreased the project risk significantly. A disadvantage of the 1:1 approach is that technical outdated solutions are kept alive. Furthermore the chances to benefit from synergies created by streamlined integration processes are passed up.

Another possible solution strategy is a total re-design of the current integration. It opens up a wide range of optimizations but usually it will go far beyond a simple platform replacement. In addition to simplifying single interfaces and moving forward to state-of-the-art technology this could be the first step towards SOA. Since re-designing is rather a new implementation than a replacement project it brings more risk. This can be managed by increasing the test effort but especially in this case it was undesirable to involve business too much. Moreover increasing the test effort implies increasing costs.

The third option is to combine a 1:1 strategy with a partial re-design. This means to find a compromise between 1:1 implementation and a total re-design. For example, with PI some monitoring and/or persistency steps can be removed from integration scenarios because they are already provided by SAP PI standard functionality. This doesn't mean any change from end-to-end perspective but will streamline the respective integration. Another opportunity could be to review original SAP interfaces (e.g. move business partner replication from IDocs to ABAP proxies). But in some cases it makes sense to consider even fundamental changes, e.g. moving from a flat file interface to a web service or an ABAP proxy.

Implementation Approach

Implementation approach is the way how the replacement project will be set up. The conventional way is to have a project team which gathers the requirements and afterwards implements, tests and deploys the solution.

For the referred project both implementation teams chose an offshore approach. Basically that means that a relative small onsite team gathered the requirements and created the technical specifications. These documents were handed over to the respective offshore development teams in India. Thus the customer benefited from the lower daily rates for the offshore team and from the cost savings due to synergy effects: an offshore developer who already implemented some interfaces for the one object, e.g. customer master data, will need significantly less time for implementing another interface for the same object.

At first glance the implementation approach seems to be closely coupled with the solution strategy: partial or full re-design to the project implementation approach and 1:1 implementation to the offshore approach. But the experience from the referred project shows that even the combination of 1:1 implementation and partial re-design is applicable to the offshore approach.

Go Live Strategy

In the referred project an iterative go live strategy was decided, i.e. single interfaces or groups of interfaces were set live after a successful user acceptance test. For smaller projects a "big bang" approach (i.e. simultaneous go live of all interfaces) is possible as well but this is a big challenge for the monitoring and support teams. One of the biggest advantages of the iterative strategy is that workload for the monitoring and support team and thus the risk is much more manageable. In the worst case the faulty interface has to be switched back to the old integration platform.

Another advantage was that the time between development/test and go live was pretty short. In most cases the original developers were still available for bug fixing and this decreased the time for error correction. If there are 6 months or even more between development/test and go live there is a risk that the original developers are not available anymore.

Preparation and Planning

The most important document for preparation, planning and tracking was the interface inventory which mentions all interfaces on a sender/receiver/data object level together with some grouping information. In retrospect, unique interface numbering instead of using the ambiguous data object names would have been very helpful.

Even if not used directly for daily operations the interface inventory was the basis for

- Definition and assignment of work packages
- Coordinating change management organization
- Status reporting
- Deployment scheduling
- The folder sub-tree structure for document management

For replacement projects internal marketing is a key factor for success since the commitment and support of the business system owners are indispensable. Unfortunately a middleware replacement brings some additional effort and risks to the business system owners and therefore it is a good idea to convince them by explaining the benefits of replacing the current middleware. Moreover they should be involved as soon as possible. In the referred project the original plan was to involve them only for the customer acceptance test. But experience showed that they are needed much earlier for test planning and even for having proper test data for unit and integration test.

As a consequence of the iterative go live strategy the project came to a permanent go live phase. Due to limited resources for testing, deployment, monitoring and support it became more and more crucial to have a careful deployment planning to ensure that only a manageable number of interfaces were replaced per day. For further replacement projects this potential bottleneck will have to be considered much more.

Another crucial preparation activity was to gather the available documentation. In this case it was an enhanced functional specification and the mapping description for each interface that were provided by the customer. "Enhanced" means that the functional specifications provided also the required technical information. Since there was no (or no sufficient) documentation available the customer established a special team to fill this gap. By means of these documents the technical specifications were created. For to have standardized deliverable documents, technical specification and all the other documents had to be created mandatory based on templates. It is crucial to have them aligned prior to the development process.

Implementation

To ensure consistent development – particular since there were two implementation partners each with multiple teams and moreover some of them working offshore – following documents were mandatory before implementation could start:

- Design Guideline: It defines how to approach certain requirements, e.g. DB access via JDBC, ftp handling, ccBPM, etc.
- Software Architecture Concept: It defines how to maintain the software structure in SLD
- Naming Conventions: How to name all the development objects in SLD, Integration Repository and Integration Directory
- Cooperation Guidelines: How to manage parallel working with the same objects, e.g. receiver determinations
- Development Process Description: All single steps from specification to deployment in production
- Test Strategy: Which tests have to be conducted and who is responsible

Furthermore some fundamental developments for common use such as UKMS access were provided at the beginning of the project. As an improvement for further PI projects UDF governance will be established.

Due to the late involvement of the business system owners it became difficult in some cases to get test data even for unit testing. To assure compliance to the given standards each interface was reviewed by a special team. Experience showed that even for the technical specification and the implementation the appliance to the given standards should be checked.

As already experienced in earlier projects, in this project too, it was very important to have a strong SAP basis support when starting the development. Particularly when an offshore team is involved idle times potentially can become very expensive for the customer: Conventional project teams often can balance system outages by temporary switching to another task. This is no option for offshore team members as they are usually focused on development. Moreover they are fully assigned to the particular project and thus idle times will be charged to the customer.

The way how changes are managed has an impact on the development. The worst case is that there is no change management established at all. The best case is that there is a lean change process established which allows managing changes adequately but doesn't hamper deployment of new developments. Between these options there is a wide range how to manage changes. In the referred project there was a change process established but it was not really suitable for PI changes and brought a lot of additional administration effort to the project teams. The conclusion for further projects is either to streamline the change process according to the PI requirements or to have a project team member who is responsible for change management. Ideally this resource should be provided by the customer.

Testing

Similar to normal IT projects there were three test types:

- Unit Test: To be conducted by the developer to prove that the respective interface is working properly
- Integration Test: Verification that the newly developed interfaces are behaving as the same interface on the old EAI platform. This includes picking up/receiving the data, processing them and delivering them to the receiver
- Customer Acceptance Test: Finally the owners of the affected business systems have to approve that the respective process is working identically with the old and the new EAI platform

Due to the late involvement of business system owners in some cases it was hard to get appropriate test data. In most other cases team members were able to create test data by triggering messages themselves in the source system. Unfortunately there was no certainty about the appropriateness of the test data.

For integration test it was decided to trigger the same data twice from the sending system: First via the old EAI platform and then via SAP PI. So the main challenge was to extract the data from the target system for comparison. Both results were expected to be identical. Otherwise some investigation would have to be done to find the reason. Even here in some cases the acquisition/provision of test data was not satisfying.

For Customer Acceptance Test there were appropriate test data available since they were driven by the business system users.

The conclusion from the tests is that it is indispensable to involve business system owners as soon as possible in the replacement project to gather appropriate test data. Otherwise there is a risk that major issues will come up only during customer acceptance test. Supposedly this consumes the same time from business system owners then providing test data in an early stage – or even more.

Go Live and Handover to Support

Before go live was approved following mandatory documents had to be provided:

- Interface Documentation
- Go Live/Cut Over Task List
- Adapter Configuration Sheet

Interface documentation should be more or less self-explaining. Nevertheless it was accompanied with an overview presentation to introduce the people in operations in an easy way. The go live task list was a schedule which described every single step required to deploy a certain interface (interface group) in production environment starting with the change management approval and ending with activating the affected communication channels. Furthermore there was a description to go back to the old EAI platform in case of major issues. After the first go lives transport of receiver determinations became a pain point. In many cases several teams developed interfaces with the same source system and same interface but different target systems. In PI there is only one receiver determination which routes the message to the target systems. Due to the different time schedules for the different development teams sometimes some older receiver determination versions did overwrite newer ones in one of the four system layers (development, test, acceptance and production) and corrupted already live interfaces by overwriting the receiver determination. Finally it was decided to exclude receiver determinations from transports and to maintain them manually in a test, acceptance and production environment. The description how to maintain receiver determination was an important topic of the go live task list.

The third mandatory go live related document mentioned the connectivity parameters and credentials for connecting source and target system to PI.

In most cases each cut over was following the same principles:

- Deploy all transports to productive PI system and verify consistency
- Connect SAP PI to the target system
- Interrupt the connection between target system and old EAI platform
- Wait until the backlog in the old EAI platform has been processed

- Interrupt the connection between old EAI platform and target system
- Connect SAP PI to the target system
- Check whether some messages stuck in PI due to not established connection to target system and restart them
- Monitor the new interfaces

For a certain sustain phase after go live the development team was still in charge to ensure smooth operation of the interfaces. Then the handover to the support service provider was initiated by providing the support team with the documentation. A few days later a handover workshop was conducted where first the documentation was presented to the support team. Afterwards there was a Q&A session. After this workshop and – if needed – after doing some documentation enhancements the support team took over support for the interfaces with some more days shadow support from the development team.

Resume

After more than 18 months project duration all planned interfaces were deployed. The Mapping Factory involvement proved to be very successful. Currently some minor development for some newly identified interfaces is ongoing. The main objective – reduction of license and maintenance fees – has been fully achieved.

Copyright