NGE Solutions

Building the Next Generation Enterprises

PISA

(Planning, Integration, Security and Administration)

An Intelligent Decision Support Environment for IT Managers and Planners

CASE STUDY: SOA Planning Through PISA-AIM

January 22, 2008

NGE Solutions, Inc. (www.ngesolutions.com)

Table of Contents

IMPORTANT NOTE TO THE READERS

This case study is extracted from another document "SOA Planning Through SOA". If you have already read that document, you wont find anything new here.

1. Overview of Case Study

To illustrate the main issues addressed by AIM, let us consider the following case study about a retail store (Xshop). To improves sales, the company needs a very flexible online purchasing (OP) application that is based on SOA. The company needs help in addressing the following issues: what other applications interface with OP, how will they be impacted if OP is transitioned to SOA, what happens if OP is outsourced and hosted elsewhere, how will OP be accessed from a wide range of user devices, what type of integration technologies will be most suitable, and what will be the cost of transitioning OP to SOA?. Additional issues include: are there commercial-off-the-shelf products that can be used for OP, what type of middleware technologies are needed to support different architectures, which ESB (enterprise service bus) platform should be used, what are the performance and security tradeoffs when different components of this application participate in B2B trade, and what type of cost/benefit analysis need to be considered while evaluating these alternatives. These are non-trivial questions that require a great deal of time and effort to answer in a purely manual approach. In the following sections, we will illustrate how AIM can possibly help.

2. AIM Methodology

Development of an integration plan is a complicated task with many challenges. Instead of a 'big bang' approach where all enterprise systems are converted to SOA in an afternoon, AIM supports a gradual approach where the enterprise achieves an integrated architecture one business (application) area at a time. The AIM methodology, discussed later, guides the user through the iterative process of choosing a business problem and then developing and evaluating integrated architectures for the chosen problem.

Integration projects can be large scale enterprise-wide undertakings that may involve numerous applications. The methodology displayed in Figure 1 allows the users to break large scale integration projects into smaller pieces that can be understood, integrated and then composed into enterprise wide solutions.

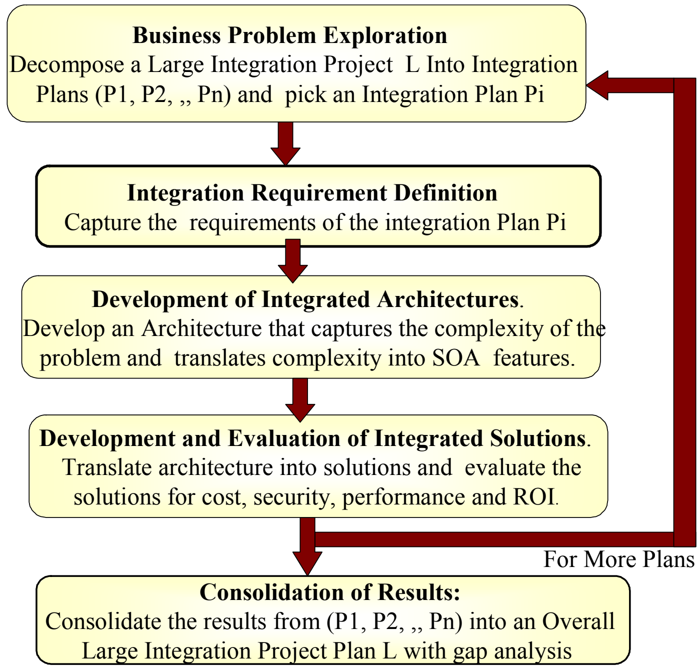


Figure 1: AIM Methodology

The main steps of this methodology are:

- **Business Problem Exploration: D**efine a Large Integration Project L that covers the overall SOA project and decompose it into Integration Plans (P1, P2,,, Pn), i.e., L= (P1, P2,,, Pn). For a small project, L = (P1). The integration plan may be defined at a Business Process (lower granularity) or Business Function (large granularity) level. For small projects involving a few applications, it is a good idea to stay at low granularity. For enterprise-wide integration projects, large granularities are better. The output is an integration project that identifies critical applications and a decomposition of the plan. **Business Problem Explorer (BPE)** supports this stage by helping the user to select and define an integration project Pi in terms of participating applications. For large scale enterprise integration projects, the user goes through this process iteratively.

- **Integration Requirement Definition.** Use the PISA Application Repository (AR) to define the integration requirements of the selected 'Target Applications'. At the core of each integration plan is a critical (target) application that is important to the business. This approach is based on the well known Critical Success Factors (CSF) methodology (Rockart, J.F. "Chief executives define their own data needs", Harvard Business Review", Vol. 57, pp. 81-93, 1979). The CSF methodology concentrates on a core set of critical issues and addresses them instead of analyzing every thing in detail. By using CSF, a user can concentrate on the apps that are critical to the success of the business and understand how their transition to SOA will impact the surrounding applications. The PISA Application Repository shows interactions between various business processes (BPs), business functions (BFs) and applications (automated BPs and BFs) and thus helps in determining critical applications. The interacting applications for each integration plan define an "Application Group" (a group of applications that will participate in an integration project). For example, if an order processing (OP) application is to be integrated in an integration plan, then the application group consists of OP plus interacting applications such as inventory, payment and shipping. **Intelligent Requirements Generator (IRG)** supports this stage by helping the user to quickly generate requirements documents that capture the essence of the integration problem for the selected business area.

- **Development of Integrated Architectures**. For each chosen integration plan, develop an integrated architecture based on SOA principles. The main objective is to capture the complexity of the problem and translate the complexity into SOA features. The process used in this stage is: a) develop a logical architecture to capture the basic complexity, b) select integration strategies (e.g., migration versus integration-in-place) for specialized considerations, and c) construct a physical service oriented architecture (SOA) based on SOA patterns that captures the key features. The output is a detailed SOA architecture that highlights the key features needed. **Integrated Architecture Advisor (IAA)** helps the user through various steps of this stage.

- **Development and Evaluation of Integrated Solutions.** The objective of this stage is to translate architecture into solutions and evaluate the solutions based on metrics (e.g., costs, security, performance and return on investment -- ROI). The process used in this step is: a) translate the selected architecture *A* into plausible solutions (*S1, S2,,Sn*) by using different product mappings in terms of COTS (commercial off-the-shelf product), and b) evaluate the solutions (*S1, S2,,Sn*) in terms of metrics such as cost, security and performance. Cost estimates due to the chosen architecture are based on the complexity of the ESB selected, the type and number of adapters needed, platforms to be bought/used, commercial-off-the-shelf (COTS) packages to be used, etc. Security implications are based on security patterns chosen in the architecture and performance implications are based on the configurations and allocations. **Intelligent Solution Advisor (ISA)** supports this stage by guiding the user through the process of cost, performance and security estimates and producing ROI (return on investment) analysis of the integration project.
- **Consolidation of Results**. After evaluating the solution by ISA, the user can go back and re-evaluate the same problem for different architectural configurations or pick another integration plan by going back to the BPE. After reiterating through the individual integration plans (P1, P2,,, Pn) of the large integration project L, now consolidate the results into an overall project document (Grand Consolidated Report). The objective of this stage is to re-iterate, consolidate the results from different projects and do gap analysis. The consolidation effort may be zero for small single application projects but may be considerable for enterprise-wide application integration projects. The process used in this stage consists of several steps: a) for large scale projects, re-iterate to pick another critical application and go through previous stages, b) consolidate results at the end of iterations to develop a business case, including total ROI and c) develop gap analysis by determining an FMO (Future Method of Operation), a PMO (Present method of Operation) and developing a transition plan for going from PMO to FMO.

3. Stage 1: Business Problem Exploration - Understanding the Problem

This stage is supported through the Business Problem Explorer (BPE) that allows the users to browse through the AIM knowledgebase to select applications that will participate in an integration project. For example, the user selects OP by using the PISA knowledgebase. The knowledgebase consists of 3 parts: pattern[1] repository, object models, and COTS database. The Pattern Repository (PR) plays a central role in AIM because we heavily use patterns to quickly develop solutions. In particular, i*ndustry patterns (IPs)* are the main starting point for this stage. Figure 2 shows example of an industry pattern (IP) that captures a high level view of a retail company, similar to XShop, in terms of enterprise functional areas (e.g., sales, corporate management, back-office operations, supply chain management), the major business processes in each functional area (e.g., purchasing and payment within sales) and the key interactions between these processes. We have created IPs for 12 industry segments that include manufacturing, healthcare, telecom, and others. These patterns are stored in the Patterns Repository, part of the planning knowledgebase, as XML documents so that they can be analyzed and modified based on a simple interview.

The user starts by invoking the BPE to choose an industry segment and thus fetching appropriate IP for that industry. The user reviews the IP, modifies it if needed, and selects the critical applications that drive the SOA projects. For example, Figure 3 shows the result of choosing order

[1] A pattern, simply stated, is a sketch that can be refined and specialized for different situations.

processing (OP) as a critical application. This screenshot of BPE shows the external interfaces of order processing application such as selling chain management, purchasing, customer payment, These are the applications that will be affected if OP was transitioned to SOA and thus help in understanding the complexity and the impact of transition OP to SOA.

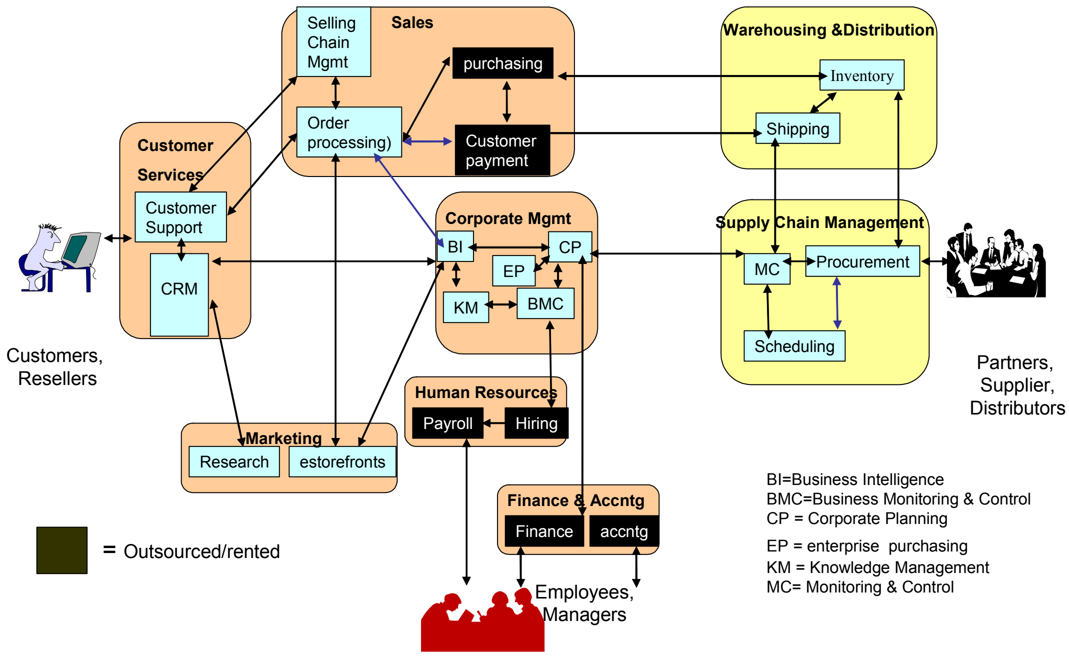


Figure 2: Industry Pattern (IP) for a Retail Company

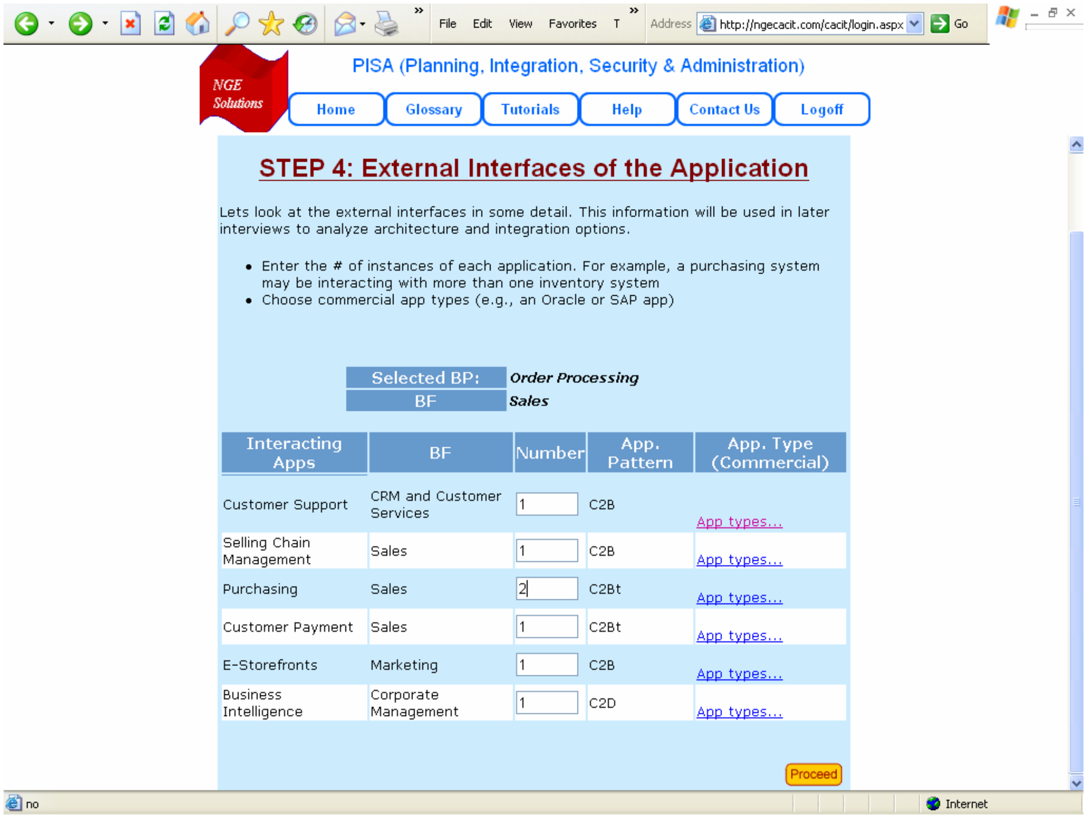


Figure 3: Selection of Order Processing as a Target (Critical) Application

4. Stage 2: Integration Requirements Generation – Documenting the Problem

Development of an integration plan is a complicated task with many challenges. Instead of a 'big bang' approach where all enterprise systems are converted to SOA in an afternoon, a gradual approach is better where the enterprise achieves an integrated architecture one application area at a time. Before proceeding with technical decisions, it is important to develop an understanding of the problem, establish a business case by identifying the applications that will actually participate in an SOA project and capture the key integration requirements.

Development of integration requirements is an important but extremely time consuming process. An Integration Requirement Generator (IRG) helps the user to quickly develop a requirements document. The heart of IRG is an interview that starts with the information already captured by the BPE in the previous stage. Additional information is gathered through the interview that considers factors such as user access, back-end apps, B2B apps, transaction value, transaction volume, number of partners, mobility, personalization, etc. Figure 4 shows a partial snapshot of the interview. The outputs of this interview are used to populate the requirements document. In short, to develop a requirements document for integration of online purchasing application, the user basically fills out an interview form shown in Figure 4. As a result of this interview, IRG selects appropriate integration patterns from the Pattern Repository and customizes them based on the results of the interview.

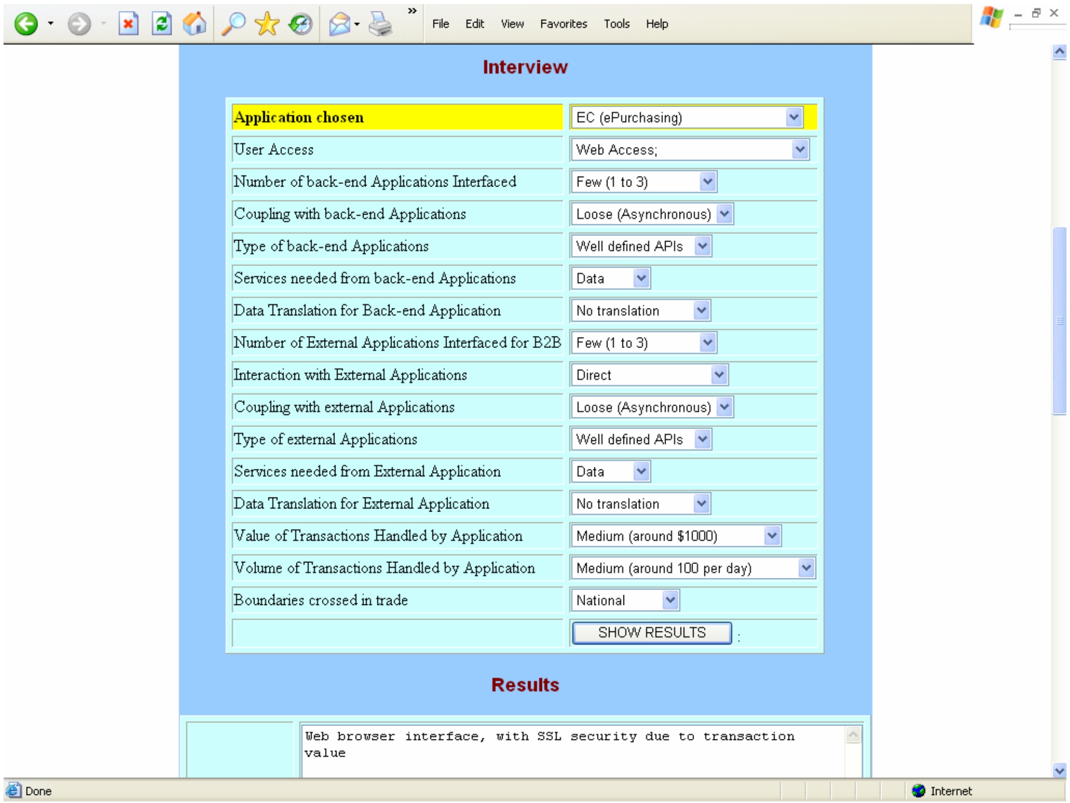


Figure 4: Sample Interview

5. Stage 3: Integrated Architecture (SOA) Stage – Capturing the Complexity

This stage translates the requirements model created in the previous stage into a component based Service Oriented Architecture (SOA). The output of this key stage is a detailed architecture document that captures the complexity of the problem and translates it into SOA features by using the following steps (see Figure 5):

- **Development of a Logical Architecture based on SOA (shown in** Figure 5**a):** Assumes that an application consists of N large grained components, each providing a set of business services. The components are arranged in several tiers: front-end integration, business logic, etc. This logical architecture can be used to determine the different types of adapters needed for different tiers.
- **Selection of Integration strategies (shown in** Figure 5**b):** The user chooses an integration strategy such as integration in place (i.e., integrate existing systems without changing any), data warehouses (develop a common database to be shared by multiple applications), migration (gradual or sudden replacement of existing apps) or composite (all of the above). This helps in selection of the SOA patterns for different integration strategies.
- **Construction of a Physical Service Oriented Architecture (shown in** Figure 5c**)**. In this step, the logical architecture is translated into a physical architecture by using SOA patterns. . The appropriate SOA ESB (Enterprise Service Bus) configuration plus the infrastructure components (adapters, registry, hubs, zones, etc) are chosen to support the different integration strategies shown in Figure 5a.

The Integrated Architecture Advisors (IAA) supports this stage by invoking three different interviews support the aforementioned three steps. These interviews gradually capture the complexity of the integration problem. Figure 6 displays a sample interview that shows the type of information (e.g., type of platform, types of services needed, and the type of data translation) needed for each application that interacts with order processing (e.g., customer support, selling chain management, and purchasing). This interview identifies the types of integration adapters that will be needed to integrate order processing with its interacting applications. As a result of the interviews in this stage, a detailed architecture document is generated that contains the adapter information, the ESB features needed, and overall SOA-based architecture.

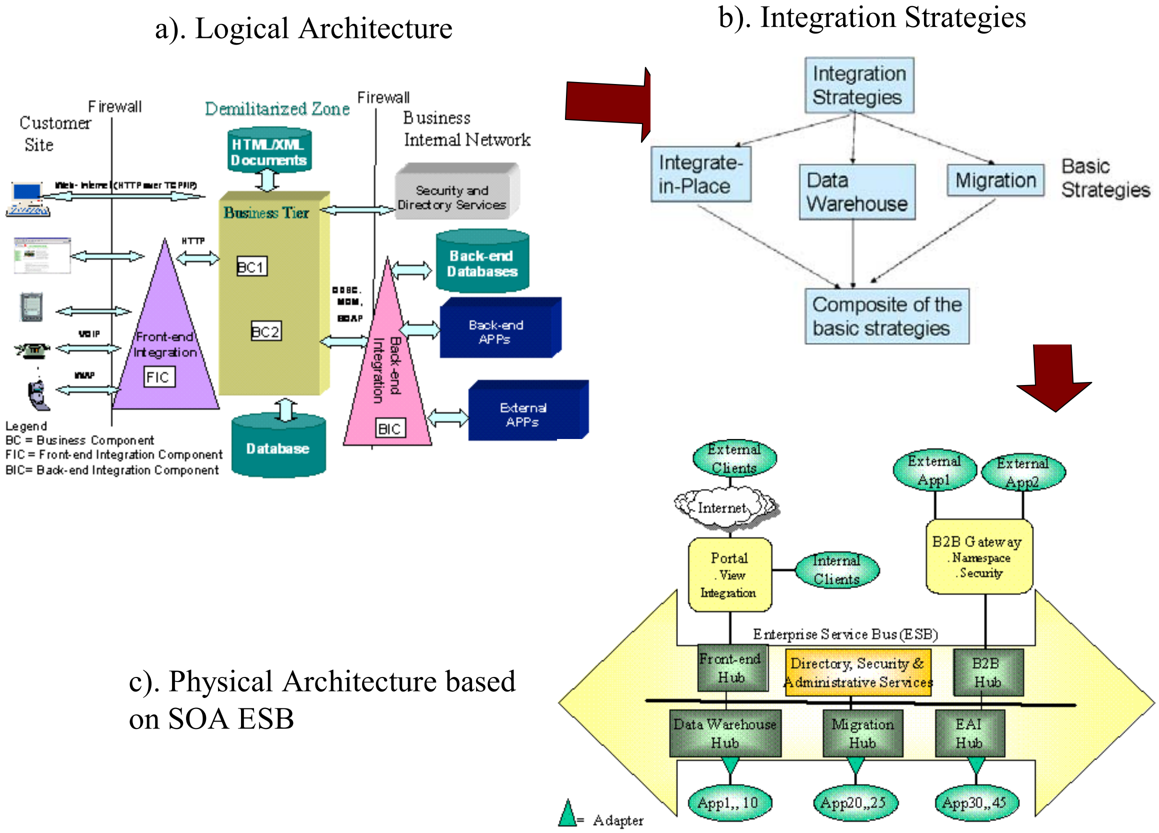


Figure 5: Gradual Development of an Architecture

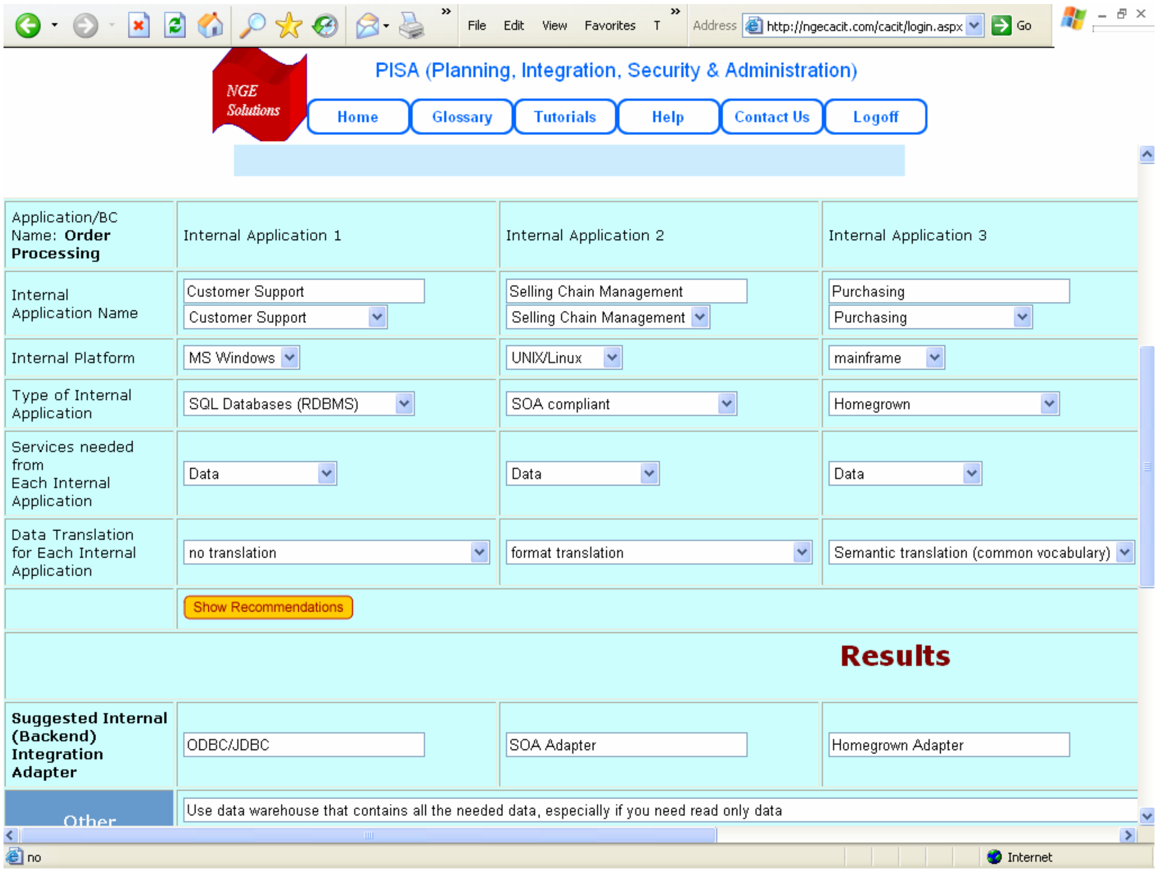


Figure 6: Interview to Capture the Complexity of the Problem

6. Stage 4: Solution Evaluation – Cost, Security and Performance Analysis

This is the most important stage from a management point of view because it involves estimation of costs, performance and security issues for each architecture solution. Specifically, this stage goes into further details by translating the SOA architecture *A,* produced previously, into plausible integrated solutions (*S1, S2,,Sn*) with appropriate commercial-off-the-shelf (COTS) packages and cost/ performance/security evaluations. The main activity of this stage is to evaluate the solutions (*S1, S2,,Sn*) in terms of the following:

a) **Cost estimates** due to the chosen architecture. The cost estimates are based on the complexity of the ESB chosen, the type and number of adapters needed, platforms to be bought/used, commercial-off-the-shelf (COTS) packages to be used, etc. Cost estimates include:

- Platform costs that show the ESBs, front-end portals, B2B gateways, adapters, and other platform component costs
- Development costs that show the development costs (e.g., developing an adapter) and installation/maintenance costs
- Miscellaneous costs that include training costs and the costs of rework due to errors

b) **Security implications** based on SOA and other technologies. The security issues due of SOA are investigated by using attack trees and security patterns. In particular, the following security issues are noted:

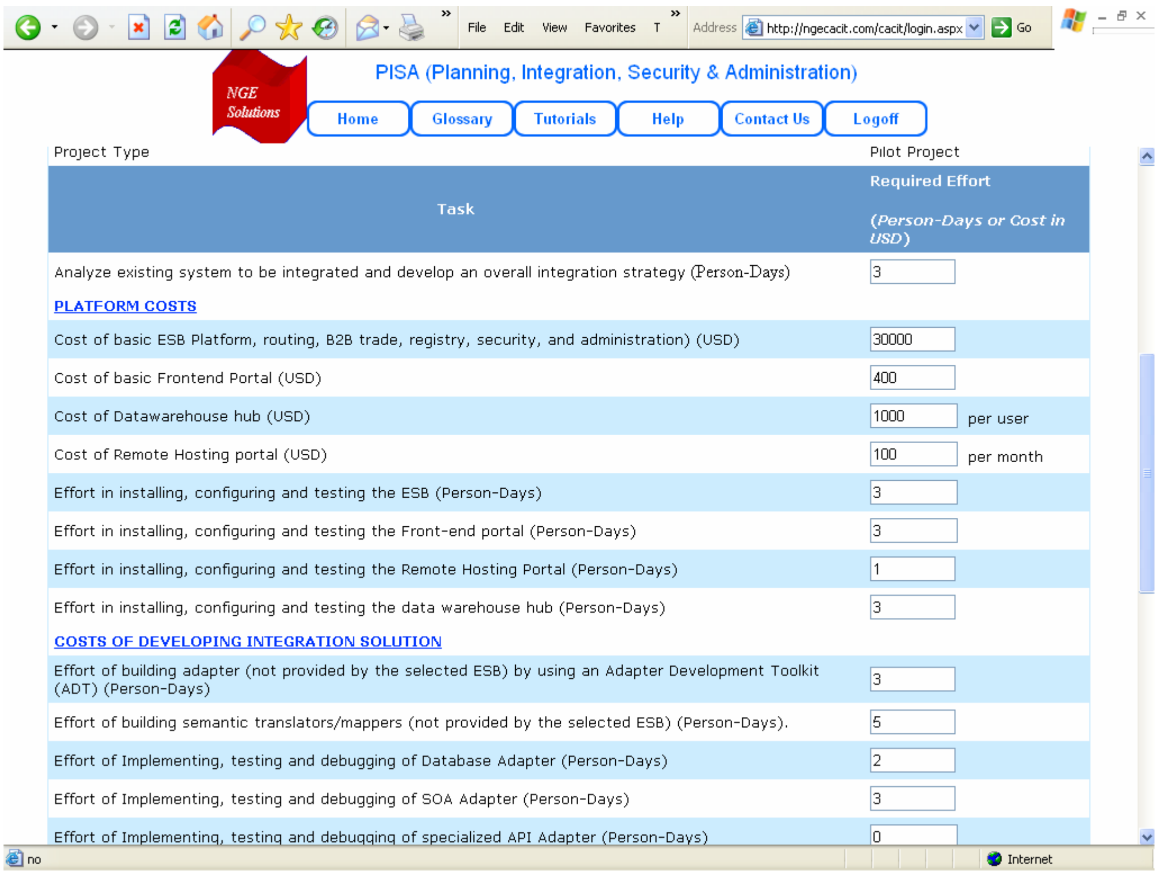
- Security of the ESB facilities (e.g., protecting the ESB directory)
- Security of the service providers and service consumers that use the ESB

c) **Performance implications** based on the configurations and allocations. An analytical queuing model is used to estimate performance bottlenecks. The main focus is to determine how many servers will be needed to support the ESB.

These steps produce a table (Table 1) showing the evaluations for different solutions for the order processing application. This stage produces several details reports. Figure 7 shows partial view of a sample cost estimation report produced by the Integrated Solution Advisor (ISA) that supports this stage. The sample report shows the platform as well as development costs.

Table 1: Example of Solution Evaluations for a Small Company

Choices	**Estimated Costs ($)**	**Performance**	**Security Issues**	**Comments**
Integrate	$120K (it is relatively cheaper to install an ESB and adapters)	2 seconds. (adapters introduce delays)	ESB & adapters may be targets for attacks & need to be secured	May need to migrate in future
Migrate and replace with an ERP	$500K million (it is expensive to completely replace a system with an ERP system)	1 second (no adapters are needed, hopefully, for an integrated ERP system)	Security can be designed for the new system from scratch	Migrations are typically expensive and require staff training
Data Warehouse	$200K (it is expensive to convert data and construct a data warehouse)	0.7 seconds (data level access is usually faster due to no overhead)	ETL needs to be protected, data level access needs protection	Data warehouses create duplicate data that needs to be synchronized



Project Type	Pilot Project	
Task	Required Effort (Person-Days or Cost in USD)	
Analyze existing system to be integrated and develop an overall integration strategy (Person-Days)	3	
PLATFORM COSTS		
Cost of basic ESB Platform, routing, B2B trade, registry, security, and administration) (USD)	30000	
Cost of basic Frontend Portal (USD)	400	
Cost of Datawarehouse hub (USD)	1000	per user
Cost of Remote Hosting portal (USD)	100	per month
Effort in installing, configuring and testing the ESB (Person-Days)	3	
Effort in installing, configuring and testing the Front-end portal (Person-Days)	3	
Effort in installing, configuring and testing the Remote Hosting Portal (Person-Days)	1	
Effort in installing, configuring and testing the data warehouse hub (Person-Days)	3	
COSTS OF DEVELOPING INTEGRATION SOLUTION		
Effort of building adapter (not provided by the selected ESB) by using an Adapter Development Toolkit (ADT) (Person-Days)	3	
Effort of building semantic translators/mappers (not provided by the selected ESB) (Person-Days).	5	
Effort of Implementing, testing and debugging of Database Adapter (Person-Days)	2	
Effort of Implementing, testing and debugging of SOA Adapter (Person-Days)	3	
Effort of Implementing, testing and debugging of specialized API Adapter (Person-Days)	0	

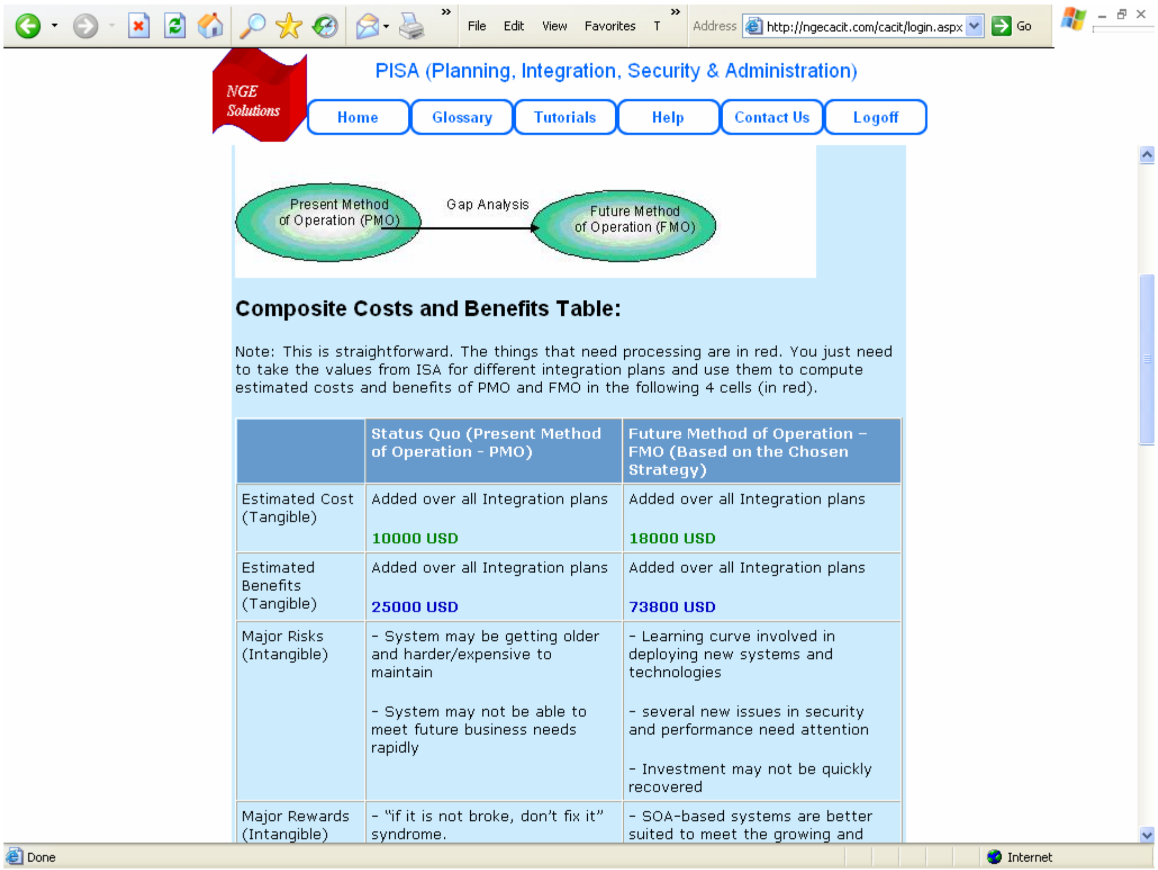
Figure 7: Sample Cost Estimates

7. STAGE 5: REITERATE AND CONSOLIDATE RESULTS AT CONCLUSION

The objective of this stage is to re-iterate, consolidate the results from different projects and do gap analysis. The consolidation effort may be zero for small single application projects but may be considerable for enterprise-wide application integration projects that require many applications to be integrated. For large scale projects, each iteration handles only a few applications so several iterations are needed. There is a need to consolidate results at the end of iterations and to develop an overall business case, including ROI (return on investment) and gap analysis. *G*ap analysis can be conducted by using the following approach:

- From the solutions (*S1, S2,,Sn*) produced in iterations 1 through n, respectively, choose the best solution S* based on evaluation
- Use S* as the FMO (future Method of operation)
- Use S0, the current system, as the PMO (Present method of Operation)
- Do a cost-benefit and ROI analysis of transitioning from PMO to FMO This includes tangible as well as intangible costs as well as benefits of the PMO, the PMO and the transitions.

Figure 8 shows a sample of gap analysis. Although some of these gap analysis and ROI calculations can be done manually, it is virtually impossible to do a good gap analysis without having a clear picture of the FMO based on a systematic analysis and an automated tool as suggested by the various stages of this model.



Composite Costs and Benefits Table:

Note: This is straightforward. The things that need processing are in red. You just need to take the values from ISA for different integration plans and use them to compute estimated costs and benefits of PMO and FMO in the following 4 cells (in red).

	Status Quo (Present Method of Operation - PMO)	Future Method of Operation – FMO (Based on the Chosen Strategy)
Estimated Cost (Tangible)	Added over all Integration plans 10000 USD	Added over all Integration plans 18000 USD
Estimated Benefits (Tangible)	Added over all Integration plans 25000 USD	Added over all Integration plans 73800 USD
Major Risks (Intangible)	- System may be getting older and harder/expensive to maintain - System may not be able to meet future business needs rapidly	- Learning curve involved in deploying new systems and technologies - several new issues in security and performance need attention - Investment may not be quickly recovered
Major Rewards (Intangible)	- "if it is not broke, don't fix it" syndrome.	- SOA-based systems are better suited to meet the growing and

Figure 8: Sample Gap Analysis

