Dependable System Architecture for Businesses: Analysis of an Enterprise Resource Planning System

By **Kannakumar Kittusamy**

Submitted to the System Design and Management Program and the Department of Civil and Environmental Engineering in Partial Fulfillment of the Requirements for the Degrees of

Master of Science in Engineering and Management

and

Master of Science in Civil and Environmental Engineering

at the
Massachusetts Institute of Technology
June 2006

Signature of Author______________________________
Kannakumar Kittusamy
System Design and Management Program
Department of Civil and Environmental Engineering
February 09, 2006

Certified by______________________________
Michael Cusumano
Thesis Supervisor
Sloan Management Review Distinguished Professor, MIT Sloan School of Management

Accepted by______________________________
David Simchi-Levi
Thesis Reader
Professor of Civil and Environmental Engineering and Engineering Systems

Accepted by______________________________
Patrick Hale
Director, System Design and Management Program

Accepted by______________________________
Andrew J. Whittle
Chairman, Departmental Committee for Graduate Students
Department of Civil and Environmental Engineering

Dependable System Architecture for Businesses:

Analysis of an Enterprise Resource Planning System

By

Kannakumar Kittusamy

Submitted to the System Design and Management Program and the Department of
Civil and Environmental Engineering in Partial Fulfillment of the
Requirements for the Degrees of
Master of Science in Engineering and Management
and
Master of Science in Civil and Environmental Engineering

Abstract

This thesis defines the concept of Dependable System Architecture (DSA), evaluation criteria for DSAs and applies those criteria to evaluate SAP, a leading ERP system. The evaluation criteria employ system frameworks, concepts, tools, components and methodologies. During the process of evaluation, this thesis touches upon many different architectural evaluations such as technical, operational, implementation and service and support. Finally, this thesis concludes whether SAP provides Dependable System Architecture for businesses.

Thesis Supervisor: Michael Cusumano
Title: Professor, MIT Sloan School of Management

Thesis Reader: David Simchi-Levi
Title: Professor of Civil and Environmental Engineering and Engineering Systems

Acknowledgments

I would like to thank the following people. Professor Michael Cusumano for supervising my thesis, lecturing the Software Business course, inspiring to research on ERP systems and providing me guidance and direction. Professor David Simchi-Levi for being my advisor and helping me to learn variety of topics throughout the year. Professor Edward Crawley for lecturing the System Architecture course. In this thesis, I have used his pictures and concepts. Mr. Matti Kinnunen for studying with me throughout the year, inspiring me to appreciate 'system architecture' and most of all for his great friendship. Mr. Subu Govindaswamy, my cousin, for his caring support during this and many previous years.

Contents

List of Figures

List of Tables

1. Introduction

Survival of an existing or a new business depends upon how well it executes its business strategy. This means converting the business strategy into day-to-day operational strategies. A variety of reasons create gaps in this conversion process. To bridge these gaps, a business can use Enterprise Resource Planning (ERP) system with dependable system architecture. This thesis defines the concept of Dependable System Architecture (DSA), evaluation criteria for DSAs and applies those criteria to evaluate SAP, a leading ERP system.

In chapter 2, the definition for Dependable System Architecture is established; one of the important components is discussed; some of the important characteristics are introduced; and criteria to evaluate a system are proposed.

SAP, a leading ERP system, has a complex system architecture, which has many characteristics of a DSA. Therefore, criteria defined in the chapter 2 are used to evaluate SAP system. Chapter 3 deals with one of the criteria 'critical analysis of SAP's architectural framework'. This is a combination of architectural analysis at the SAP system level, capability analysis of SAP at company level and competitive analysis at the market level. Also, an analysis for platform leadership is accomplished at the end.

Chapter 4 critically analyzes and evaluates the technical system architecture of the SAP system. This includes system layers, solution independence nature, installation architectures and dependability. Also, the traditional and new SAP system architectures are compared with each other and architecture of SAP NetWeaver is described. Strengths and weakness of SAP architecture is discussed as well.

In a company, SAP typically thrives and creates an eco-system. The eco-system includes many solutions. Most of the possible solutions are discussed in chapter 5 with emphasis on SAP NetWeaver. The chapter explains how the use of a standard software platform in SAP eco-system makes the eco-system dependable to businesses. Strengths and weaknesses of SAP eco-system and mitigation strategy are also discussed from the perspective of businesses.

Stakeholder analysis helps to improve the implementation and operation architectures of the SAP system. Chapter 6 identifies SAP system stakeholders in a typical company and relationships among them. Moreover, the chapter maps the value flow among the stakeholders and looks at the operational architecture of SAP from an enterprise and societal context. In the end, SAP's business value proposition is covered in brief.

In chapter 7, the reasons for SAP service and support processes being critical to the survivability of the SAP eco-system are discussed.

Analyzing the future trends in SAP technology is critical for the complete evaluation of DSA. Therefore, chapter 8 explains SAP's Enterprise Services Architecture (ESA) framework and its relation to dependability. Also, ESA is compared against two competing Service Oriented Architecture (SOA) frameworks from SAP's competitors.

Chapter 9 is a case study that discusses SAP's Auto ID Infrastructure, RFID solution, in detail. Chapter 10 is another case study that discusses the emergence of on-demand Customer Relationship Management (CRM) solution for businesses.

After multiple analysis and evaluations, this thesis concludes that the SAP eco-system does have Dependable System Architecture for businesses. Again, based on the very definition, Dependable System Architecture emerges in order to meet the goals and objectives of the stakeholders within SAP's surrounding business context.

2. Defining Dependable System Architecture

2.1 Definition

Software architecture, according to Shaw and Garlan [1], is about "... the organization of a system as a composition of components; global control structures; the protocols for communication, synchronization and data access; the assignment of functionality to design elements; the composition of design elements; physical distribution; scaling and performance; dimensions of evolution; and selection among design alternatives." This definition clearly demonstrates the vital essentials of software system architecture. Defining whether a software system architecture is 'dependable system architecture' for a business is much harder than this definition. Apart from the system details, the definition of dependability should also include the function by the 'manifestation' of the software code in its surrounding context.

Therefore, 'dependable system architecture for a business' is defined as "software architecture that satisfies the underlying requirements behind its existence and manifests itself to meet the goals and objectives of the stakeholders within its surrounding business context". The complete creation, organization, management and evolution such architecture will decide whether it is a 'strategic fit' to a business. Measuring the level of dependability is a tedious process; however, it is possible to define generic criteria that will assess the level of dependability of software architecture within a specific business context. Software is extremely flexible in terms of its configuration and implementation and that will also play a major role in the level of dependability for a business.

While addressing about dependability, it is also important to discuss about failure. Many factors could lead to failures. The usual suspects are the structural defect or design defect of the software. However, many times the software architecture failures are caused by the way it is implemented. For example, factors such as configuration errors, inappropriate system landscape, poorly designed interfaces and also application integration errors contribute to major architecture failures. Also, it is important to note that the application integration errors are observable in the behavior of the software architecture. Application integration errors usually increase the Total Cost of Ownership (TCO) for businesses. Integration points increase the complexity of a system and managing the complexity causes the TCO to increase.

Also, it is imperative to note that the business environment could some times become the source of software architecture failure. According to Jackson [2], any software-intensive system that interacts with the natural world is potentially vulnerable to that world's unbounded capacity for varied and novel behavior. Here the 'natural world' is nothing but the business environment where the software architecture is being used. Jackson [2] further adds, developing a successful system depends on identifying and selecting those behaviors that are likely to prove significant, and making soundly judged decisions about the system properties needed to deal with them effectively. Some of these statements reiterate the importance of the software configuration and implementation. System project/program management processes and tools can help to deal with the implementation and make sound system management decisions. Therefore, it is evident that deriving necessary system behaviors from the customer requirements is critical to the architectural dependability.

2.2 Dependable software platform

To develop dependable system architecture, the first and foremost step is to define a dependable software platform. According to Fricke [5], platform is a defined set of common elements and its interfaces. Elements could be architectural essentials such as system, subsystem, process, organization and component. Once the platform is defined, then its variants and derivatives create the product family or members of the software eco-system. In creating the variants, it is vital to consider the flexibility of the system architecture. As per Fricke [5], flexible system architecture could be shared by family of system; end up creating an eco-system or system-of-systems

According to Olivier de Weck [4], creation of an eco-system is constrained by something called 'platform bandwidth'. System variants that are derived from a platform determine the upper and lower bounds of a platform and this extent are known as 'platform bandwidth'. If development of a system falls out of this extent, the same platform cannot be used. Therefore, dependable system architecture always ensures its systems or the members of its eco-system fall within the extent of the 'platform bandwidth'. Building system variants from the same platform reduces the design and development time, promotes reusability, decreases the development and implementation costs and reduces the complexity among the members of the eco-system. Hence, a clear platform strategy is required along with the system architecture to create an eco-system or system-of-systems. Use of dependable system architecture would enable a business to respond to changing market needs faster. So, this discussion illustrates how important dependable software platform is for dependable system architecture.

2.3 Improving dependability through flexible adaptation

In order to improve dependability, flexible adaptation to change is crucial. To manage overall change to the business rules, processes and policies, dependable system architecture would allow two kinds of change [7]. First, the business domain related change. The business domain rules are handled by the business system users, who have no technical knowledge, but by implementing the rules they can adapt to the system in order to cope with changing business requirements. Second, the system domain related change. The system domain rules are intended for system developers to add new behavior without changing the original service implementation.

To allow the above mentioned changes, dependable system architectures should use 'configuration layers'. The act of configuration will ensure that the system will evolve according to given constraints such as business and system policies. At a very technical level, according to [7], the flexibility of adaptation to change is achieved by two means. The first is the definition of parameterized business rule types. The conditional parameters can be combined in arbitrary Boolean expressions to provide expressivity, and priorities among rules of the same type allow distinguishing between general vs. exceptional cases. The second is at run-time, from the actual data passed to the invoked service, a configurator component retrieves the applicable rules, parameterizes the service according to the rules, and creates the necessary binding instances. The bind will intercept some of the service's functionalities and replace it by the new behavior associated to the corresponding business rule. So, flexible adaptation to change is one of the required properties of dependable system architecture.

2.4 Survivability architecture

Enterprise software architectures are often massive in dimension and highly complex and they exceed the scope of comprehension of system designers and analysts. With this unimaginable complexity comes the potential for undetected errors in the system. While software often causes this problem, its form can be exploited to ameliorate the difficulty in what is referred to as survivability architecture [8].

Let's review one of the famous definitions. Survivability [9]: "A property of a system, subsystem, equipment, process, or procedure that provides a defined degree of assurance that the named entity will continue to function during and after a natural or man-made disturbance". Therefore, for a given system architecture, survivability must be qualified by specifying the range of conditions over which the entity will survive, the minimum acceptable level or post-disturbance functionality, and the maximum acceptable outage duration. Therefore, dependable system architecture is survivable if it complies with its survivability specifications [10]. Some may even argue that survivability and dependability are equivalent. However, this research considers survivability as one of the critical elements of dependable system architecture.

Survivability view is analogous to the informal view of reliability. It states that the system rarely or never fails. However the formal definition of reliability is the following. As per [9]: "The ability of an item to perform a required function under stated conditions for a specified period of time and the probability that a functional unit will perform its required function for a specified interval under stated conditions".

As per [10], here are the characteristics of systems that affect its survivability.

- System Size
    - Systems are very large
    - They are geographically diverse and topologically complex
    - Large numbers of heterogeneous computing, storage and network elements
- Externally Observable Damage

If the damage is extensive, it will be visible to the users of the system in the form of deteriorated quality of service.

- Damage and Repair Sequences.
    - Sequence of events that might occur over time in which each event causes more damage
    - Users of the system might experience progressively less service
- Time-Dependent Damage Effects.

The impact or loss associated with damage tends to increase with time.

- Heterogeneous Criticality

The requirements for dependability of systems are considerable but the requirements vary with time.

- Complex Operational Environments
    - Operating environments of the systems vary significantly across the eco-system
    - Operating environments include unprecedented complexity
    - Various sources could contribute to accidents or failures

2.5 Survivability architecture and fault tolerance

One of the ways to acquire survivability architecture is to have system architecture with fault-tolerant design. However, the specifications of survivability don't have to correspond exactly to the specifications of fault tolerant design. The role [8] of survivability architecture is to provide the necessary system-level framework to implement the fault tolerance necessary to meet the system's survivability goal.

Dependable system architectures should have in-built fault tolerance capability. Even when some of its subsystems fail, the overall system continues to provide remaining services through its other operating subsystems. The total failure of the system is an unexpected phenomenon for the fault tolerant capable system that is well aligned with the system's survivability goal. In order to accomplish fault tolerance, this research recommends Reconfiguration-Based Fault-Tolerance approach.

Reconfiguration could be defined as architecture initiated change to the configuration of software application during its runtime. According to [11], based on where the change occurs, two kinds of reconfiguration is possible.

1. Component level reconfiguration: Any change to the configuration parameters of single components.
2. Architectural level reconfiguration: Any change in the application topology in terms of number and locations of software components.

Type 1 reconfiguration is recommended when the software modules or subsystems are loosely coupled and there are no dependencies among them. On the other hand, type 2 reconfiguration is recommended when there are critical dependencies between the software modules or subsystems. If the architecture follows systems-of-system model, then it is very possible to apply component level reconfiguration.

Let's discuss about the elements that should be built-in to the reconfiguration functionality. First of all, the basic premise of this approach requires monitoring of the managed software application to detect failures at component, subsystem or system level. Second, making use of reconfiguration for error recovery and bringing the system state to the desired level or fault free operational level. Reconfiguration at run time requires pre-defined system parameters, rules and system policies. While building the policies, it is important to consider all possible cause of faults or failures. Also, there should be a set of policies for the unexpected faults or failures. Many times, unexpected faults or failures will end up requiring architectural level reconfiguration.

Another vital consideration to reconfiguration is the 'constantly evolving' nature of the system and its architecture. This often requires innovative hardware or software or network based fault-tolerant mechanism. In sum, dependable system architectures have built-in fault tolerant mechanisms that are well aligned with its system survivability goals. This process ultimately strengthens the survivability architecture of a complex system or system-of-systems. So, system managers should always recognizes the subtle difference between survivability and fault-tolerance. This is also conspicuous from the above discussion.

2.6 Self-Healing architectural model

Dependable system architectures should have some level of self-healing ability. The basic premise behind self-healing architecture is that the causes of the errors or faults or failures are known to or expected by the system architects. The problem could be software or hardware or network related. As soon as the problem is detected by the system, the architecture performs a 'diagnosis'. If the diagnosis indicates that the problem is a known one, the architecture informs the problem to a system management module called 'autonomous problem manager' (APM). APM engine has a database where all the known problems and their solutions are maintained. The solutions are typically in the form of scripts. When the appropriate set of scripts is executed with appropriate prioritization by the APM engine, the script performs series of actions that fixes the known problem and brings the system component to availability.

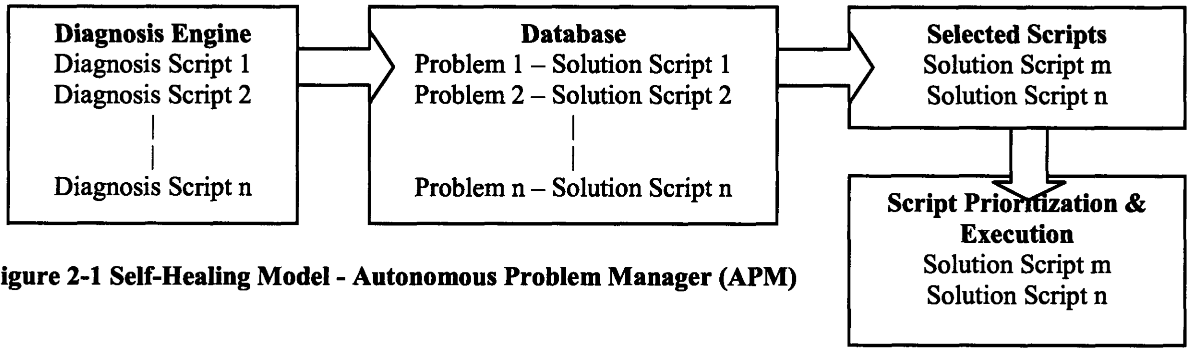


Figure 2-1 Self-Healing Model - Autonomous Problem Manager (APM)

The system or subsystem with the problem could be on the fault-tolerant mode (some of the system services are still available) or on the complete failure mode. The self-healing architecture with the help of the APM engine brings the system or subsystem back to the complete problem-free availability state. This system management function and APM could be services running on a stand-alone or shared hardware.

2.7 Evaluation criteria

Proving that a system has dependable architecture is extremely complex. However, it is possible to assess how well a software system or family of software systems meets some of the criteria for dependable system architecture. The following criteria are based on a pragmatic approach to architectural evaluation. Many of the important system properties that we discussed earlier in this chapter could be identified and validated using these criteria.

- Critical analysis of system architecture proves that the system is dependable enough for a business at present and in future.
- Technical system architecture exhibits the characteristics of standardized stable technology, easily separable layers or tiers, componentized logical software units and system of systems arrangement.
- The software eco-system has a clear boundary. The components of the eco-system use common components (e.g. reusable software platform).
- Stakeholder analysis is clearly defined; helps facilitate the implementation architecture through the program/project management organizations.
- System development models are comparable with industry-wide models. Also, best business practices are clearly spelled out and optimized for a particular industry use or a conglomerate company use.
- Future trends in the evolution of system architecture show a sense of industry technology leadership. Also, the upcoming products exhibit many of the characteristics of dependable system architecture.

3. Critical Analysis of SAP System Architecture

A framework for critical analysis of system architecture was developed by the author as part of the System Architecture (ESD.34) course at MIT. This developed framework is used to critically analyze the SAP system.

3.1. Analysis of the SAP system boundaries and functions

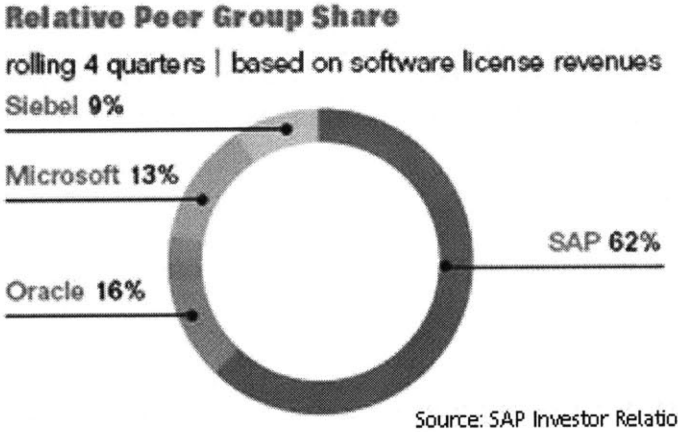
3.1.1. Corporate and functional strategies

- Be a recognized leader in providing collaborative business solutions (e.g. Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), Supply Chain Management (SCM) etc.)
- Serve all types of industries (e.g. Aerospace, Automotive, Pharmaceutical, High-tech, Non-profit)
- Provide solutions for every major market segments(Small, Mid and Large enterprises)
- Leverage experience and knowledge to deliver a comprehensive range of solutions to empower every aspect of business operations.
- The market scope of SAP software is to provide functionalities to Finance, Human Resources, Manufacturing and Customer Service organizations so that these organizations and industries can execute their business strategies effectively.

3.1.2. Industry competitive analysis

Major competitors for SAP in the ERP software space are ORACLE (+PeopleSoft, JD Edwards), Microsoft and Computer Associates. Competition from these matured companies exist, however, currently enterprise software industry is going through major consolidation. SAP possibly faces threat of new entrants. However, high capital requirements and the SAP's brand name pose a high entry barrier to other entrants. SAP is the leader in many of the ERP related market.

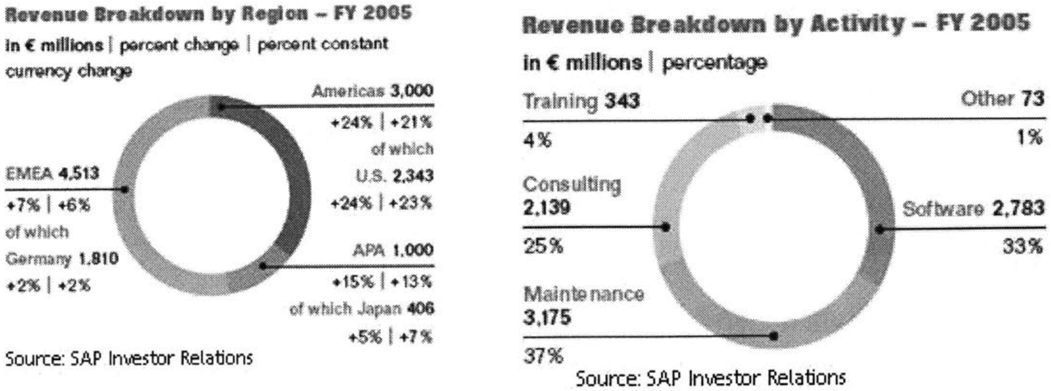
As per the 'January 2006 SAP FACT SHEET' released by the company's investor relations department, based on software revenue, SAP is the #1 business software supplier in every industry and solution segment. The company's share against its next three largest competitors is 62% globally and 47% in the United States.



Source: SAP Investor Relations

Figure 3-1: SAP- Relative peer group share

In addition, across the world, SAP also provides business support, consulting and training services, giving the company a well-diversified global revenue stream.

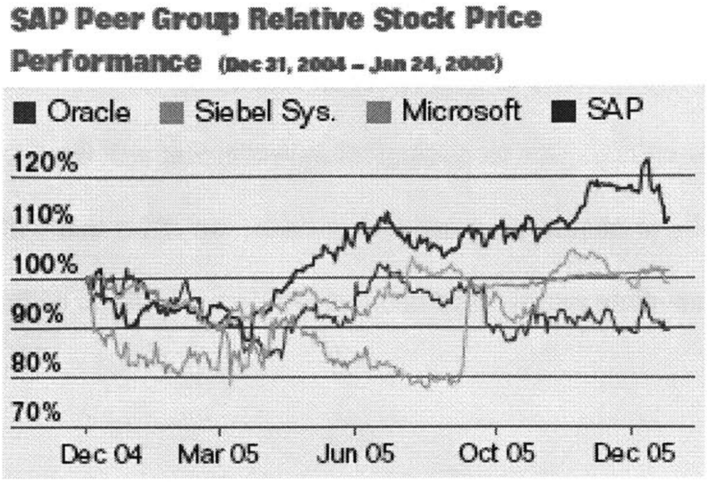


Source: SAP Investor Relations

Source: SAP Investor Relations

Figure 3-2: Revenue breakdowns by region and activity

SAP's market position and performance are sharply reflected in its stock price.

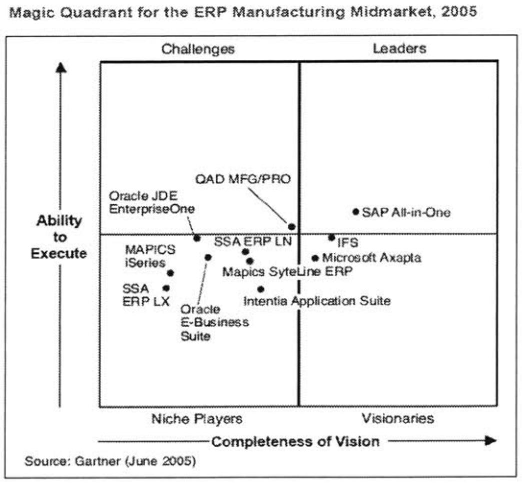


Source: SAP Investor Relations

Figure 3-3: Stock price performance (SAP peer group)

The Leader:

Also, to exemplify SAP's market leadership, here is a magic quadrant from Gartner, Inc. Gartner places SAP in the 'Leaders Quadrant' for the ERP Manufacturing Mid-market. The Leaders Quadrant (as per Gartner): Leaders deliver with strength in today's market and also have a well-developed vision for supporting and influencing the market of the future.



Source: Gartner (June 2005)

Figure 3-4: Gartner's magic quadrant for the ERP manufacturing mid-market, 2005

The challenge:

According to Gartner, the technology transformation to a service-oriented architecture (SOA) will have the largest effect on redefining the ERP mid-market since the term “ERP” was coined. This technological improvement will require large R&D investments to totally re-engineer a great deal of code – an effort that will not provide vendors with an immediate return on investment (ROI) and will put additional pressure on their survival.

3.1.3. Customers & Needs

Customers:

High-level: Business Enterprises across the Globe.

Needs:

- Day to day business operations of companies require sophisticated and robust enterprise software systems
- Different business units of an organization have diverse computing needs for executing their application algorithms
- Integration between application modules is mandatory
- Enterprise systems should run on variety of platforms and hardware
- Enterprise users demand ease of use

Here are some of the SAP’s major customers who run its enterprise resource planning software system. This list has been categorized by sector and by industry. It shows how complex SAP’s system architecture has to be so it could meet the needs of variety of customers. As we know, even companies within a same industry and sector have differing business needs and computing requirements.

Many of these companies use core SAP R/3 ERP system to meet their critical needs. To satisfy the specific needs, some of these customers buy Industry Solutions from SAP. Industry Solutions use the same SAP's standard technology platform but support industry specific business processes.

Company	Sector	Industry
BP	Basic Materials	Major Integrated Oil & Gas
General Electric	Conglomerates	Conglomerates
BMW	Consumer Goods	Auto Manufacturers
Canon	Consumer Goods	Photo. Equipment & Supplies
Coca-Cola	Consumer Goods	Beverages - Soft Drinks
Colgate-Palmolive	Consumer Goods	Personal Products
Hershey Foods	Consumer Goods	Confectioners
Home Depot	Consumer Goods	Home Improvement
Kellogg's	Consumer Goods	Food - Major Diversified
Nike	Consumer Goods	Textile, Apparel & Footwear
Procter & Gamble	Consumer Goods	Cleaning Products
Sony	Consumer Goods	Electronic Equipment
Department of Defense	Public	Defense
Kellogg Foundation	Public	Non-Profit
MIT	Public	Education
U.S. Postal Service	Public	Freight Services
Disney	Services	Entertainment - Diversified
McDonald's	Services	Restaurants
Starbucks Coffee	Services	Specialty Eateries
Target	Services	Discount, Variety Stores
Apple	Technology	Personal Computers
Nokia	Technology	Communication Equipment
Siemens	Technology	Telecom Services - Foreign
Verizon	Technology	Telecom Services - Domestic

Table 3-1: Major customers of SAP

3.1.3. Benefits & Beneficiaries

To Customers:

1. Primary Benefit: Efficient access to real-time enterprise information that enables execution of business strategies and day to day operations.
2. Non-primary Benefit: Transactional history that protects organizational information evolution. Improved employee morale and work ethics.
3. Beneficiaries: Inside an Organization: CEO, Board Members, Executive Management Team, Auditors, Middle Managers and Employees.

The following figure illustrates how Benefit flows to Beneficiaries with SAP-ERP system.

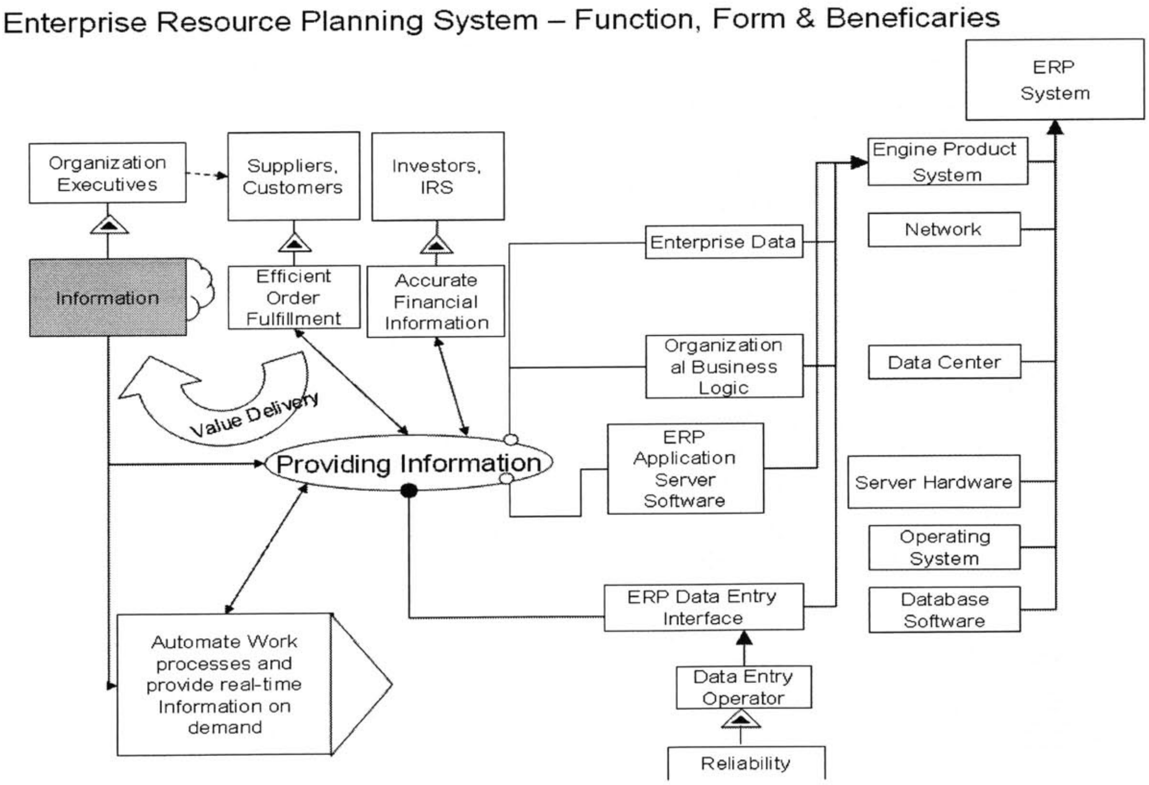


Figure 3-5: SAP value delivery

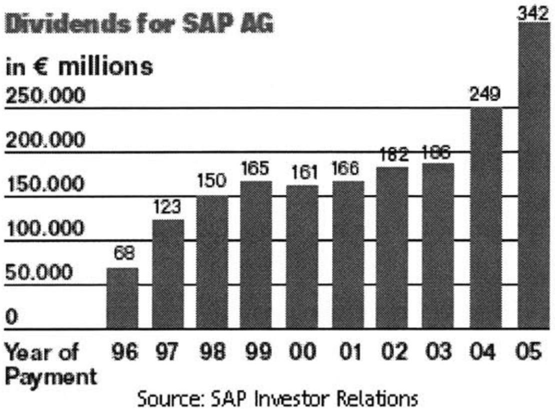
To SAP:

Benefit: Sustain leadership position in the Enterprise software industry

Beneficiaries: SAP Stockholders (Benefit is derived through Dividends and stock value appreciation)

As per [12], SAP has produced strong returns to its shareholders. Driven by a 10 year compound annual growth rate of 20% in revenues and 22% in pro forma operating income, SAP has generated a 15% compound annual growth rate in its stock price.

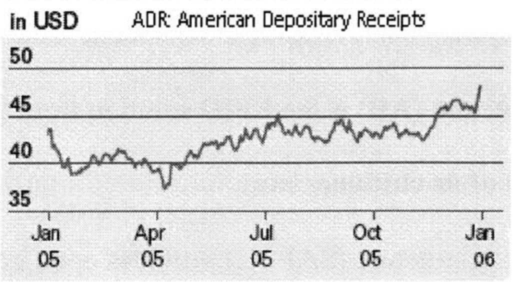
Benefit 1: Dividends to Stockholders



Source: SAP Investor Relations

Figure 3-6: SAP AG dividends

Benefit 2: Stock value appreciation



Source: SAP Investor Relations

Figure 3-7: SAP AG ADR share price

3.1.4. Overall capability & Competence of SAP AG.

SAP is the world's largest inter-enterprise software company and the world's third-largest independent software provider. SAP AG is headquartered in Walldorf, Germany. SAP software is being used by 12 million users world-wide. There are 96,400 system installations and 1,500 partners. SAP's human capital includes more than 34,000 people in more than 50 countries. The Company lists more than 32,000 customers in more than 120 countries. 75% of the Fortune Global 1,000 companies run SAP software.

3.1.5. Available technology

- SAP provides real-time transactional information access through 3-tier computing system architecture (Tier 1: Database Layer; Tier 2: Application Layer; Tier 3: Presentation Layer). That's why SAP is often called R/3. (R/3: Real Time/3-tier)
- SAP's Enterprise Services Architecture provides Service Oriented Architecture (SOA) support. This means, SAP system applications could be accessed as services.
- Web Application Server is the state of the art service providing product and Netweaver is the integrated platform.
- SAP manages the complexity through its sophisticated layered approach and it is a Database, Operating System and Platform Independent solution.
- SAP Auto-ID Infrastructure (AII) is the RFID solution from SAP. This will meet the emerging RFID needs of its customer base.
- The SAP Exchange Infrastructure (SAP XI) provides open integration technologies that enable process-centric collaboration among SAP and non-SAP systems.
- **And much more ....**

3.1.6. Regulatory Environment (Supported Standards, Principles and Laws [13])

1. International Financial Reporting Standards (IFRS): IFRS is often known as International Accounting Standards (IAS) is a set of accounting standards and issued by the International Accounting Standards Board (IASB). The European Union (EU) is requiring most publicly traded companies to issue consolidated financial statements according to IFRS.
2. Generally Accepted Accounting Principles (GAAP): A standard established by the Accounting Practices Board of the American Institute of Certified Public Accountants (AICPA).
3. US Sarbanes-Oxley (SOX): The Sarbanes-Oxley Act has broad implications for the ways organizations manage, monitor, and report on assets and operations.
4. Segregation of duties analysis: Sound corporate governance and compliance management begins with effective security controls and real-time prevention of user-rights violations.
5. Basel II: Basel II has important implications for corporate borrowers around the world. Members of the Basel Committee include the Group of Ten, or G10 countries. Basel II establishes requirements for banks to manage the risks of issuing loans.
6. Human Resources: Local employment, payroll, tax reporting, safety and other regulations place growing demands on global enterprises.
7. USA PATRIOT Act (Public Law 107-56): US Federal law designed to curb terrorist activities. Because of the disclosure requirements, the law impacts SAP financial transactions.

SAP offers processes and tools to support all of these standards, principles and laws.

Regulatory Authorities: (Who may directly or indirectly impact SAP information)

1. US Department of Homeland Security (DHS)
2. US Regulatory Authorities
3. US Internal Revenue Service (IRS)
4. European Union (EU)
5. Group of Ten (G10) countries
6. World Trade Organization (WTO)
7. International Organization for Standardization (ISO)

3.1.7 Intellectual property and R&D

Inventions and innovations are the way of life at SAP in order to meet the competitive enterprise software environment. SAP has thousands of patents approved and filed with United States Patent and Trademark (USPTO).

SAP Labs [13] represents a dynamic community within SAP's global research and development (R&D) organization, sharing insights and promoting creativity on a worldwide basis. With operations in Bulgaria, France, India, Israel, Japan, and North America, SAP Labs integrates ideas and leading-edge technologies that address the needs of specific industries and geographic regions, and keep SAP and its customers at the forefront of business success. SAP Ventures [13] is SAP's venture capital arm. Since 1996, SAP Ventures has been investing in companies that offer exciting new technologies and applications. It invests selectively in emerging technology that combines large market potential with high growth opportunity.

3.2. Analysis of SAP system goals

3.2.1 Precise terms of Product Development

SAP software should be developed using the best methodologies and models that are available in software product development arena. Waterfall Model, Spiral Model and combination of both waterfall and spiral software development models are used to develop or modify SAP software.

3.2.2 Derived from user Needs

System goals are often derived from the customer requirements. Since SAP has variety in its customer base, it is very difficult to decide the overall system goals. SAP uses its resources and organizational capabilities to manage customer requirements.

3.2.3 Corporate strategy, regulations, competitive analysis

SAP AG's strategic direction often has an impact on SAP's system architecture. Also, regulators often impose regulations and laws that have impact on architecture. Sometimes, emergence of new technology from competitor makes SAP to change its system goals.

3.2.4 Statement of goals

- Provide sophisticated enterprise system environment for businesses around the world
- Help organizations to meet their important IT needs
- Provide organizations an ability to integrate functional silos and help achieve one single enterprise

- Constantly provide software patches to meet the changing regulatory environments and address software security concerns
- Provide companies with reliable, robust and scalable information management system
- Relentlessly provide new information subsystems that work seamlessly with SAP core system to meet the emerging technological complexities
- Provide system methodologies, rapid SAP implementation drivers and training to the customer base

3.2.5 Goals defined by the Chief Architect

In SAP's case, the chief company architect has defined some ambitious goals for SAP.

Enterprise Services Architecture (ESA):

As per [13], ESA is SAP's blueprint for service-based, enterprise-scale business solutions that offer the increased levels of adaptability, flexibility, and openness required to reduce total cost of ownership (TOC). It combines SAP's experience in enterprise applications with the flexibility of web services and other open standards. The SAP NetWeaver platform is the technical foundation for ESA.

SAP NetWeaver:

According to [13], SAP NetWeaver is a comprehensive integration and application platform and it works with the existing IT infrastructure to enable and manage change. With SAP NetWeaver, organizations can flexibly and rapidly design, build, implement, and execute new business strategies and processes.

Mendocino:

Project Mendocino is enterprise software collaboration between SAP and Microsoft. Mendocino is the first joint product and is designed to transform how information workers access enterprise applications.

3.3. Analysis of the SAP system concept

3.3.1 Options and opportunities

Alternative system concepts are continuously proposed, piloted, and critically examined along with business frameworks. New opportunities are explored to provide solutions to old or new customers by SAP.

3.3.2 Key metrics and drivers

SAP makes sure that the key system metrics such as system response time, database response, average CPU load, average memory use and number of concurrent users are monitored and linked properly to customer service level.

3.3.3 Product/System life cycle

SAP Product/System life cycle is revised annually or monthly or weekly as appropriate by the Product Management teams. System bugs are identified year around by the development team and customers. SAP supplies the bug fixes to the customers as early as possible. This helps customers to maintain their system stability.

3.3.4 Failure modes

Failure modes are identified and documented properly. System redundancy tools are incorporated into the SAP system. The information is written on multiple disks. Proper backup procedures are documented. Many of the SAP's systems and subsystems have built-in fault-tolerant capability. In the SAP eco-system, even when some components or subsystems fail, other components of the overall eco-system continue to operate and provide services to the users.

3.3.5 Trade-offs and optimization

In the SAP system, trade-offs always exist among the different subsystem components. However, managing the trade-offs and optimizing the complete system are going to be key in the efficient system management. Also, 'SAP defined' best practices exist to help the customers.

3.4. Allocation of SAP functionality and definition of interfaces and abstractions

3.4.1 Decompose form and function

Decomposition is done using logical work units. SAP system includes multiple subsystems, application layers and functional modules. Within the software, there are many application architectures that make the SAP more complex. However, through a common transport system, SAP manages the management of different architectures and sub systems. Therefore, SAP system could be called as 'System of Systems' because the elements within SAP have managerial and operational independence.

3.4.2 Allocate functionality to elements

Allocating functionality to different elements is very carefully orchestrated. User needs of the SAP constantly changes. Therefore, allocation of functionalities is carefully determined with the use of proper trade-off analysis and thorough business analysis.

3.4.3 Define interfaces between subsystems

Interfaces are the most complex system components in SAP. Establishment of interfaces is based on the underlying business needs. Different technologies and protocols are employed in creating interfaces. The relationships that are created using interfaces are carefully defined and documented.

3.4.4 Configure the subsystems

Configurations of the system and subsystems are done in a static or dynamic system environment. Specific transactions exist at the system level to help create and maintain the configuration. Configuration specialists accomplish their goals by considering the user and business needs. SAP offers both Component level reconfiguration [11] and Architectural level reconfiguration [11]. These reconfigurations could be accomplished during the design time and runtime as well. This is one of the important properties of dependable system architecture.

3.4.5 Create the structure of the system

Finally, when including and packaging all the system components, a complex but complete architecture of an enterprise system emerges. The complete structure of the system is properly documented using system tools.

3.5. Evaluation for dependable system architecture

Having discussed and covered many components of SAP's architectural framework, let's focus on constructing an architectural interrelationship diagram that displays the 'Dependability' of SAP's system architecture to businesses. This step by step approach augments elements and makes an attempt to connect the SAP system architecture, SAP AG's business case and customer value proposition.

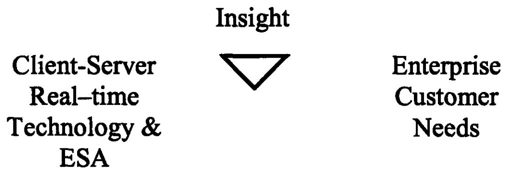

Insight
Client-Server Real–time Technology & ESA
Enterprise Customer Needs

Figure 3-8: Architectural interrelationship diagram I

Product innovation usually starts at SAP to exploit a new technology or by having understood the burning needs of enterprise customers across the globe. When look with an 'Insight', apparently 'Client-Server Real-time Technology & ESA' and 'Enterprise customer needs' are the most visible components.

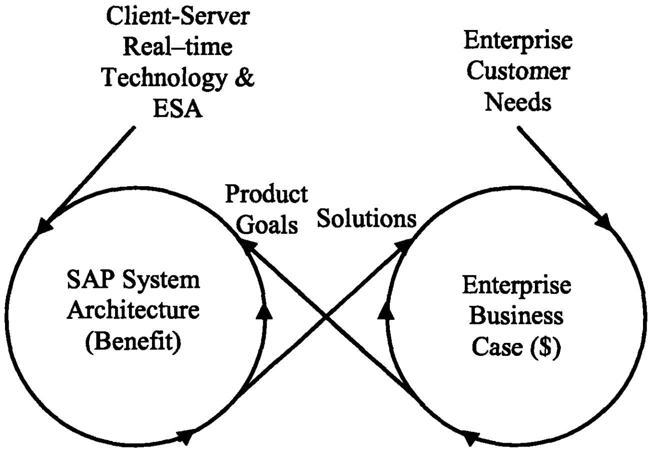

Client-Server Real–time Technology & ESA
Enterprise Customer Needs
Product Goals
Solutions
SAP System Architecture (Benefit)
Enterprise Business Case ($)

Figure 3-9: Architectural interrelationship diagram II

During the 'Conceive' phase of product development process, system architecture and the business case proceeds in an interactive manner. Either the available technology or the customer need could be the primary driver as long as they move parallel with a rhythm. Business case creates the 'product/system goals' and that become the basis for the system architecture. Therefore, system architecture becomes the answer to the business case.

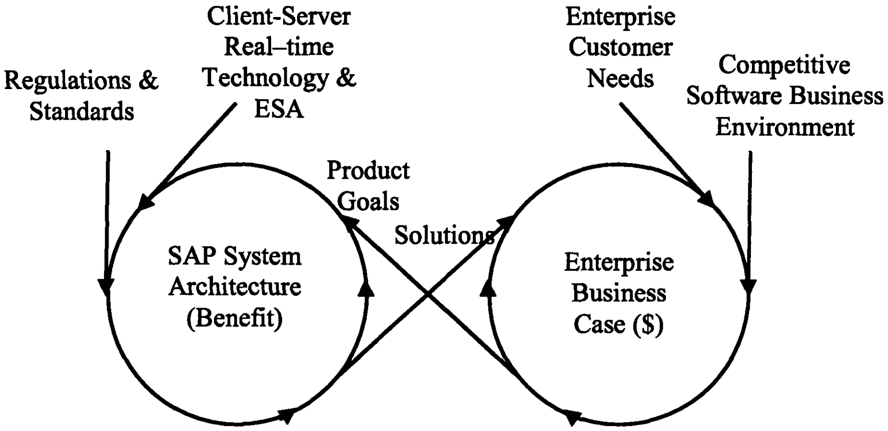


Figure 3-10: Architectural interrelationship diagram III

In SAP's case, regulations & standards set by their respective body impact the system architecture. For example, SAP cannot provide enterprise software solutions in the US without complying with Sarbanes-Oxley (SOX). Also, SAP cannot be a provider of enterprise software systems without meeting the software standard requirements of International Organization for Standardization (ISO).

On the other hand, competitive enterprise software business environment impacts SAP's business plans. For example, SAP has to meet the challenge posed by Oracle

Corporation's buy out of PeopleSoft and Siebel. Factors such as regulations, standards and competition are typically external drivers that SAP has no control over.

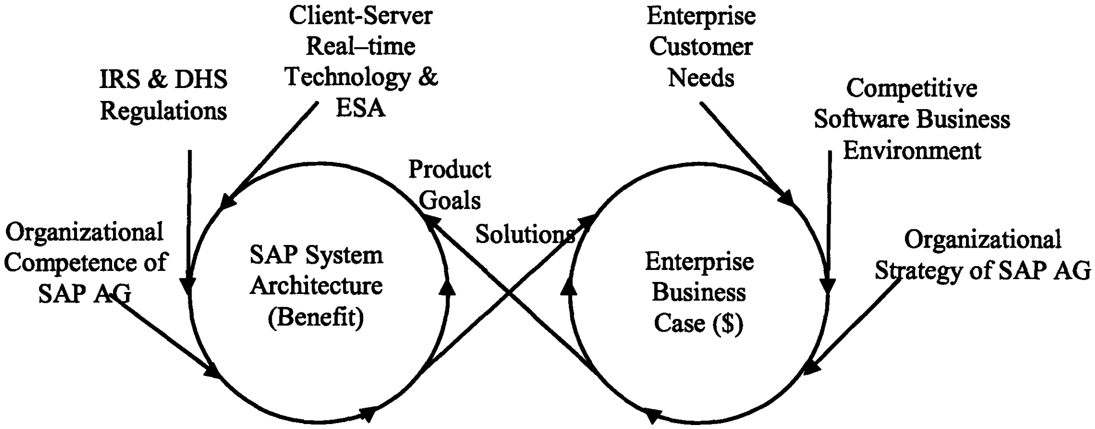


Figure 3-11: Architectural interrelationship diagram IV

To conceive, design, develop and implement the product / system, organizational competence of SAP AG is critical. SAP's organizational competence includes physical capital, human capital, intellectual capital and organizational capital possessed by SAP. On the other side, mission and goals of the product/system should align with the respective components of the organizational strategy.

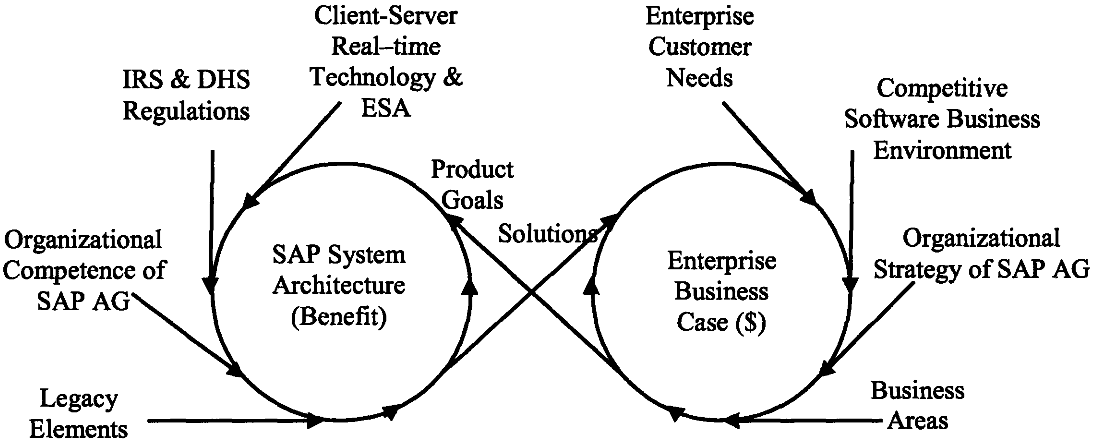


Figure 3-12: Architectural interrelationship diagram V

Legacy products related concerns need to be addressed in the new or enhanced system architecture. Once the new product is developed, it needs to be marketed and sold to the customers with the help of SAP AG's other business areas such as marketing and sales.

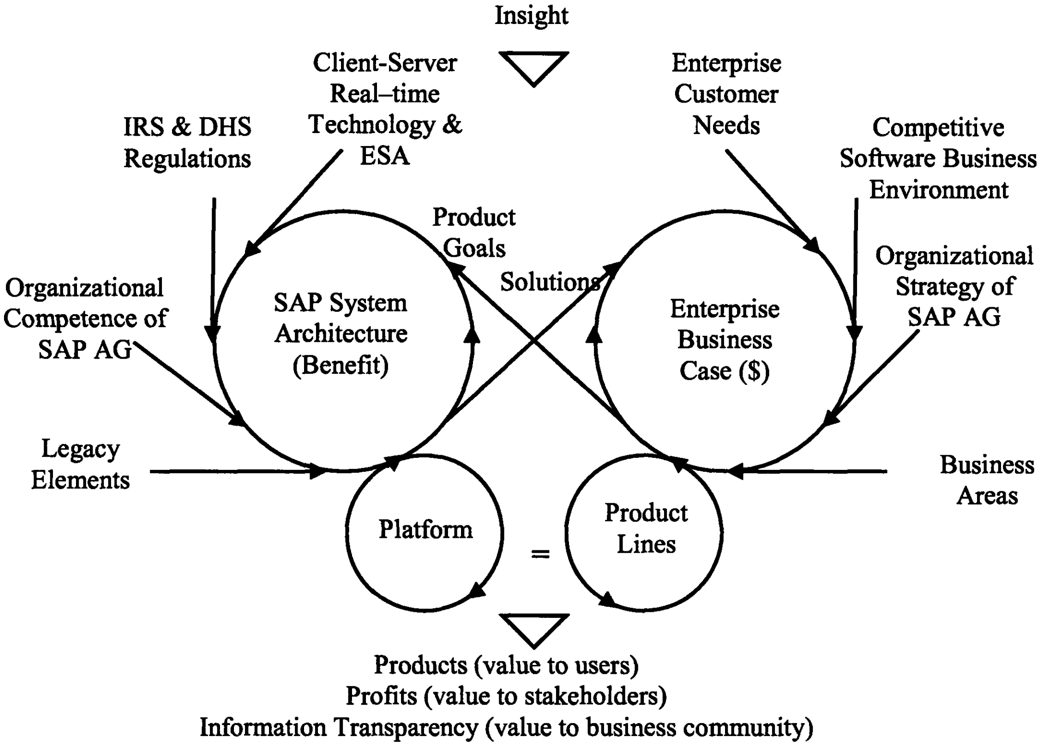

Insight
Client-Server Real–time Technology & ESA
IRS & DHS Regulations
Enterprise Customer Needs
Competitive Software Business Environment
Product Goals
Solutions
Organizational Strategy of SAP AG
Organizational Competence of SAP AG
SAP System Architecture (Benefit)
Enterprise Business Case ($)
Legacy Elements
Business Areas
Platform
=
Product Lines
Products (value to users)
Profits (value to stakeholders)
Information Transparency (value to business community)

Figure 3-13: Architectural interrelationship diagram VI

In the last step, the system architecture process creates the common dependable software platform for SAP. This platform is common to many products. On the other side, creation of a product line happens and it moves through the business channels and reaches the customers. Now it is evident that both the platform and product line are very well aligned in many aspects.

Finally, the architectural cycle closes and so does the business case cycle. Now, the commitment of SAP AG to its customers moves from product/system development to product service and support. As a result, the SAP product/system provides value to the users and the customers. Information access and transparency benefit the business community. In return, SAP AG receives revenue and its shareholders enjoy the stock value appreciation.

3.6 Analysis for Platform Leadership

According to Gawer and Cusumano [30], to become platform leaders, firms want their products to become the foundation on which other companies build their products or offer their services. The following paragraphs evaluate SAP with the use of 'Four Levers of Platform Leadership', a framework developed by Gawer and Cusumano [30].

Lever 1: Scope of the firm [30]: This deals with what the firm does inside and what it encourages others to do outside.

SAP possesses extensive in-house capability to create its own complements. Initially, SAP started with the core ERP system. With opportunities presented over a period of time, SAP expanded its complements offerings from Business Intelligence (BW) to Enterprise Portal (EP). Many of its industry-specific applications (e.g. Automotive, Chemicals) can work as an independent solution or a 'compliment' solution to an organization's mySAP ERP system. This depends on a company's SAP eco-system setup. SAP tried to promote its own open source database system to run SAP system but didn't achieve any major success. SAP relies on companies such as Microsoft and Oracle for the database component that it requires for the SAP system. SAP also relies on companies such as Microsoft and IBM for the operating system component that it requires for the SAP system.

Lever 2: Product Technology [30]: This deals with overall product architecture, interfaces, intellectual property etc.

SAP systems in SAP eco-system are highly modular. However, the degree of openness of the interfaces is very low. Most of the interfaces use SAP's proprietary algorithms and interface components. This makes creating interfaces between SAP and external systems very difficult. Outside firms have limited information about SAP interfaces. This makes potential complementors job tougher. However, many companies use this limited SAP information and have successfully developed complements (e.g. Imaging, Workflow vendors) for SAP.

Lever 3: Relationship with external complementors [30]: This lever centers on determining how collaborative versus competitive should the relationship be between the platform leader or wannabe and complementors. There is not enough information publicly available for evaluating SAP against this lever.

Lever 4: Internal organization [30]: This allows platform leaders and wannabes to use their internal organizational structure to manage external and internal conflicts of interest more effectively. There is not enough information publicly available for evaluating SAP against this lever.

The following critique is from author's personal knowledge and professional work experience. Reading of the book [30] was very helpful in coming up with some of the following conclusions or recommendations.

- SAP had a tradition of competing with its complementors. It made many of its complementors to abandon their idea of developing product complements for SAP.

- There were accusations in the ERP industry that SAP obtained lot of technical expertise and ideas from its complementors, later turned around, and became their competitor.
- In the SAP economy, other companies made more money than SAP. For example, at one instance, SAP sold more Oracle database licenses (for its system) than Oracle Company itself.
- Regardless of SAP's actions, many complementors offer compliments to the SAP eco-system. Most of the complementors, however, use their own platform instead of using SAP's WAS platform. This is mainly due to the proprietary nature of SAP system and interfaces.
- Based on author's own experience, SAP has been not successful or sophisticated in its approach to platform leadership. SAP has driven innovation in the ERP industry. But, it didn't steer the direction of complimentary innovation that was useful to them. Actually, SAP's business strategy was more useful in making money to many other companies than SAP itself.
- Based on author's professional experience, SAP is very weak with Lever 3 and Lever 4. SAP has to do lot more work in improving Lever 3 and Lever 4 to become a platform leader.
- The author suggests the following recommendations with the help of [30].
    - SAP should open its interfaces to its platform without giving away to much information or advantage to its complementors.
    - SAP should consider 'availability of complements' as an important strategy to improve its overall offerings.

- SAP needs to create a partner alliance with leading enterprise software companies, database vendors and hardware manufacturers.
- SAP should encourage its complementors use its WAS platform to develop products. In order to do this, SAP has to share lot more proprietary information with its trusted partners. By doing so, SAP can make its compliments that run on WAS perfectly fit into the SAP eco-system thus reducing TCO to its customers.
- SAP should invest in new technologies through niche start-up companies through SAP ventures. Later, when the technology matures, SAP can either buy or spin-off these startups. This mitigates the risk in developing products for a new technology. Also this way, SAP can acquire new capabilities from the market place.
- With Oracle, Peoplesoft and JD Edwards mergers, it is high time for SAP to think about strategies for 'platform leadership'. Because of the complexities involved in SAP, existing customers will have hard time moving to Oracle. If Oracle can integrate its acquired components (Peoplesoft and JD Edwards) into one single package, it could open up possibilities and incentives for the existing SAP customers to move to Oracle eco-system. But, Oracle will have lot more work to do to accomplish this task. In the next couple of years, SAP has to strive to become a platform leader so that many firms in the industry use SAP WAS as its standard platform to develop their products and services.
- SAP should phase out its old programming environment ABAP (Advanced Business Application Programming) and fully adapt to Java as its universal

programming environment. At the same time, provide .Net connectors so that applications can be developed using Microsoft technologies as well.

- There is nothing wrong with SAP's business strategy of being one stop shop for all enterprise computing software. But, it needs to cultivate more complementors who use SAP's standard platform but offer solutions to reduce SAP's complexity and improve business user experience.
- Finally, SAP needs to evolve its platform while rallying other firms around these changes.

4. Technical Architecture of SAP

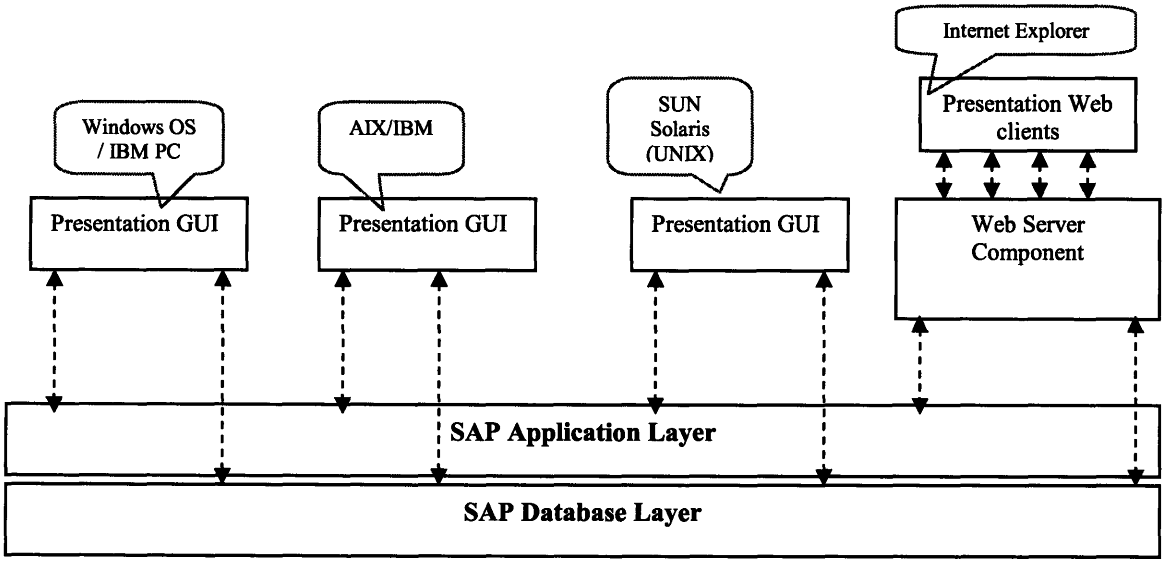
SAP uses three-tier system architecture to reduce its system complexity and manage risk. The three tiers are presentation, application and database layers. This chapter analyzes the technical architecture of SAP from a higher abstraction level to a detailed level.

4.1 SAP presentation layer

SAP Thick client approach: This client could run on any personal computer or work station or server platform. Regardless of the end-user environment, SAP thick client seamlessly accesses the SAP core system. The SAP core system could run on any SAP supported database, operating system or hardware platform.

Thin (web) client approach: An integrated or stand-alone web application server is available to access the SAP Application server. Now, only an internet browser is required to access the SAP core system. Currently, SAP bundles a web server component with its application servers, therefore; a separate web server is no longer necessary. Integrated web server approach improves the performance of the overall SAP system.

Figure 4-1: SAP 3-Tier Architecture – A Higher Level of Abstraction



4.2 SAP application layer

SAP application layer is where all the application logic is kept and run. This application logic directly connects to both the SAP database layer and the presentation layer. This application layer has capability to interface with external systems and receive information from even an external logic as well. In a local or wide area network, multiple application layer components could be run on multiple hardware machines to support variety of SAP application logic. This application layer not only reduces the load on the database layer but also helps to improve overall system response time. Apart from processing business critical application logic, the application layer could also be used to handle non-critical business services such as 'printing', 'background jobs' etc. This application layer acts as a buffer for database layer when a request is received from a user. The user request doesn't reach the database layer but instead it gets processed at the application layer level. The application layer retrieves information from its own data buffers and supports the processing of the user request. Buffer validation cycles are built-in between the application and database layers in order to maintain data validity and integrity.

4.3 SAP database layer

SAP database layer is where a supported relational database management system (RDBMS) is kept and run. The complete SAP system is stored in the RDBMS as a component. The subsystem components and data are distributed across multiple database tables. SAP utilizes 'one single database' concept to ensure consistency and maintain reliability of its core real time 3-tier (R/3) ERP system.

4.4 SAP communications

In the SAP system environment, the systems and subsystems need to communicate with each other effectively. SAP uses Common Programming Interface - Communications (CPI-C) standard call interface for applications, which perform direct program-to-program communications. As per [14], CPI-C was first defined as a standardized communications interface by IBM in 1987, as part of the SAA standard.

The main advantage of CPI-C is the easy portability of programs to various system platforms made possible by the common interface. As per [14], the CPI-C communications interface essentially fulfills the following requirements of program-to-program communication:

1. Communication setup, 2. Data exchange, 3. Data conversion, 4.Communication control, 5. Communication close.

The CPI-C interface can be split into two function groups.

- CPI-C Starter Set (Establishing a connection, Data exchange, Closing a connection)
- Advanced Function Calls (Data Conversion, Synchronization and control, Changes in communication characteristics, Checking of communication characteristics, Security functions)

The CPI-C interface is available for both C and ABAP (Advanced Business Application Programming) programming languages.

4.5 Solution independence

The major advantage of having 3 layers in SAP is to enable software solution independence. This means SAP's 3-tier could be deployed using majority of RDBMS software, operating system software and platforms that are available in the market place. SAP supports all of the following operating systems, RDBMS and platforms.

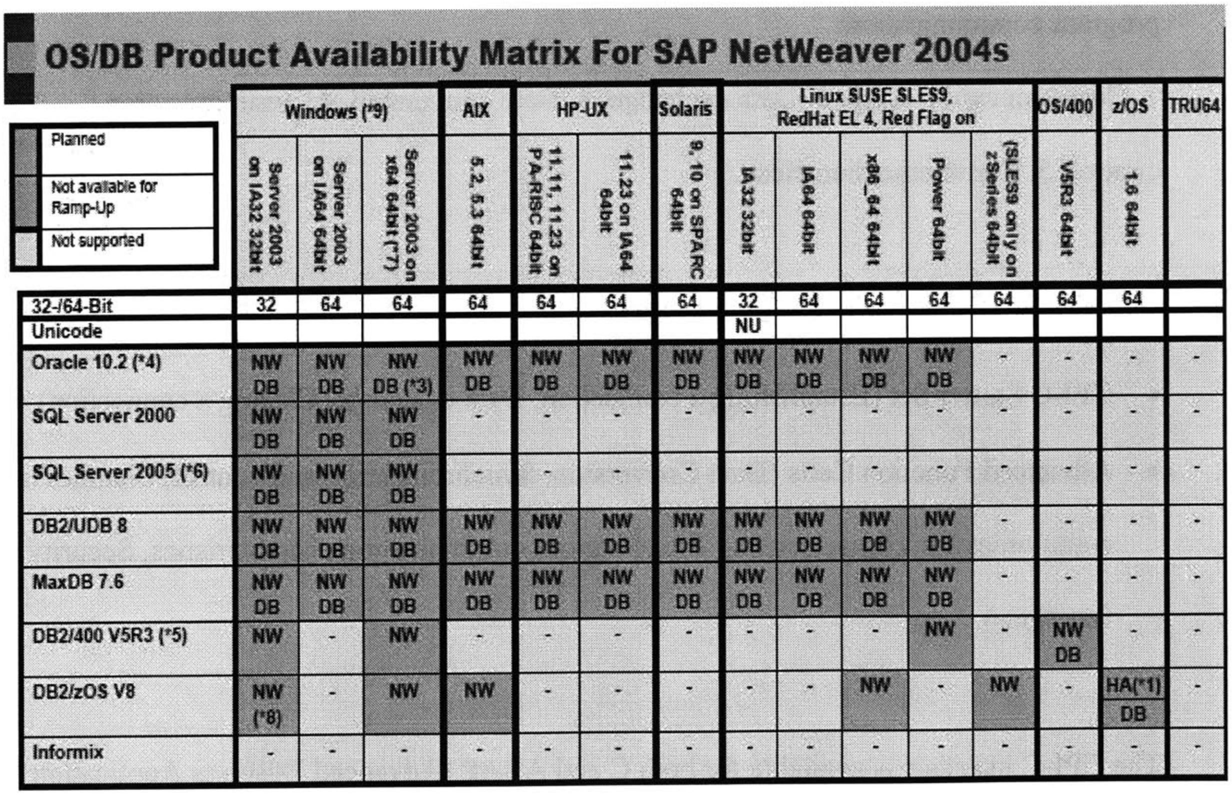
Operating Systems:

Microsoft Windows Servers, Linux, Solaris, AIX, HP-UX, OS400, z/OS, TRU64

RDBMS:

Microsoft SQL Server, ORACLE, Informix, DB2

Figure 4-2 shows the possible combination of installations for SAP Netweaver.



OS/DB Product Availability Matrix For SAP NetWeaver 2004s

Planned
Not available for Ramp-Up
Not supported

	Windows (*9)			AIX	HP-UX		Solaris	Linux SUSE SLES9, RedHat EL 4, Red Flag on					OS/400	z/OS	TRU64
	Server 2003 on IA32 32bit	Server 2003 on IA64 64bit	Server 2003 on x64 64bit (*7)	5.2, 5.3 64bit	11.11, 11.23 on PA-RISC 64bit	11.23 on IA64 64bit	9, 10 on SPARC 64bit	IA32 32bit	IA64 64bit	x86_64 64bit	Power 64bit	(SLES9 only on zSeries 64bit	V5R3 64bit	1.6 64bit	
32-/64-Bit	32	64	64	64	64	64	64	32	64	64	64	64	64	64	
Unicode								NU							
Oracle 10.2 (*4)	NW DB	NW DB	NW DB (*3)	NW DB	NW DB	NW DB	NW DB	NW DB	NW DB	NW DB	NW DB	-	-	-	-
SQL Server 2000	NW DB	NW DB	NW DB	-	-	-	-	-	-	-	-	-	-	-	-
SQL Server 2005 (*6)	NW DB	NW DB	NW DB	-	-	-	-	-	-	-	-	-	-	-	-
DB2/UDB 8	NW DB	NW DB	NW DB	NW DB	NW DB	NW DB	NW DB	NW DB	NW DB	NW DB	NW DB	-	-	-	-
MaxDB 7.6	NW DB	NW DB	NW DB	NW DB	NW DB	NW DB	NW DB	NW DB	NW DB	NW DB	NW DB	-	-	-	-
DB2/400 V5R3 (*5)	NW	-	NW	-	-	-	-	-	-	-	NW	-	NW DB	-	-
DB2/zOS V8	NW (*8)	-	NW	NW	-	-	-	-	-	NW	-	NW	-	HA(*1) DB	-
Informix	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Figure 4-2: OS/DB Product availability matrix

4.6 Technical architecture of SAP application server [15]

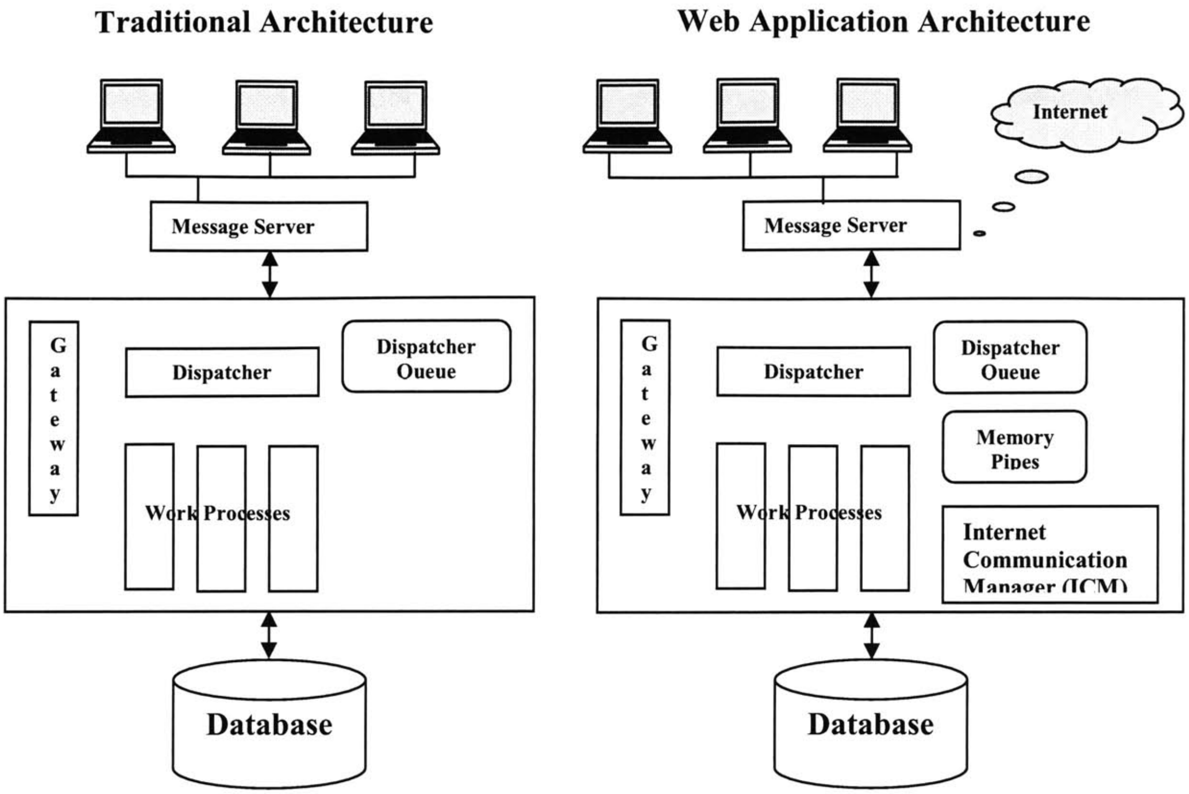


Figure 4-3: Comparison of architectures

Typically, a business user initiates a request through SAP presentation server. The user request reaches the Message Server where it gets routed and forwarded to an available application server. A logon load balancing algorithm facilitates the user request forwarding process. The request reaches the Dispatcher where it gets assigned an available work process for business processing. The work-process processes the request by accessing the application data buffer or the underlying database. The processed request is then sent back to the user for further business tasks or value addition.

SAP Web Application Server (WAS or Web AS) is a further development of the traditional SAP Application Server technology [15]. This WAS is an integral part of SAP's NetWeaver technology platform. This new WAS infrastructure is capable of receiving, processing and sending **http** or **https** requests from internet or intranet users. To allow this additional internet capability, the SAP kernel has been given an additional process called 'Internet Communication Manager' (ICM). ICM is clearly depicted in the architectural diagram and uses threads to communicate on the internet as a client or a server. Memory Pipes are used to make the data transfer possible here. ICM could be accessed through a transaction in the SAP system where it could be administered by a trained system administrator.

4.7 Installation architecture of WAS [14]

SAP Web Application Server technology supports SAP's own Advanced Business Application Programming (ABAP) and Java as well. WAS can be installed in different system variants. Selection of a system variant depends upon the type of system implementation that meets the underlying business need.

1. SAP Web AS ABAP + Java System

With this system installation variant, both the ABAP engine and Java (J2EE) engine operates under one system identification. This means one SAP system.

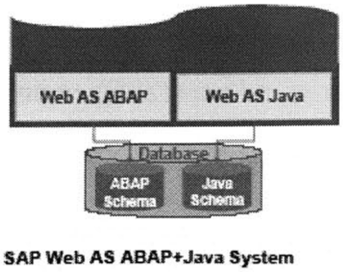


Figure 4-4: SAP Web AS ABAP+Java installation variant

2. SAP Web AS ABAP System + SAP Web AS Java System

With this system installation variant, ABAP engine and Java (J2EE) engine operate under two different system identifications. This means two different SAP systems.

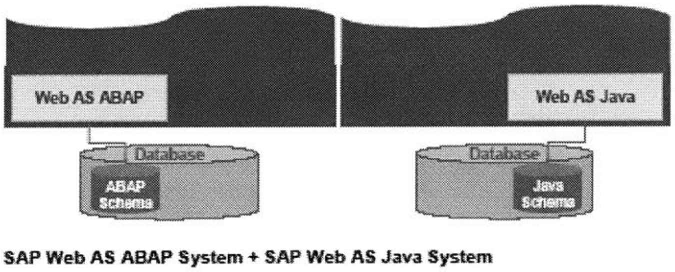


Figure 4-5: Web AS ABAP+Web AS Java installation variant

3. SAP Web AS ABAP System

With this system installation variant, an ABAP engine operates under one system identification. This means one SAP system.

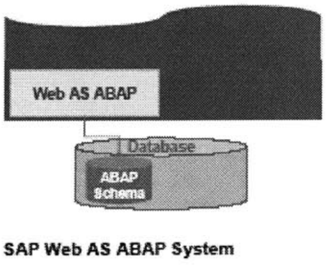


Figure 4-6: Web AS ABAP installation variant

4. SAP Web AS Java System

With this system installation variant, a Java (J2EE) engine operates under one system identification. This means one SAP system.

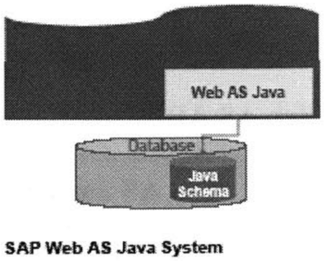
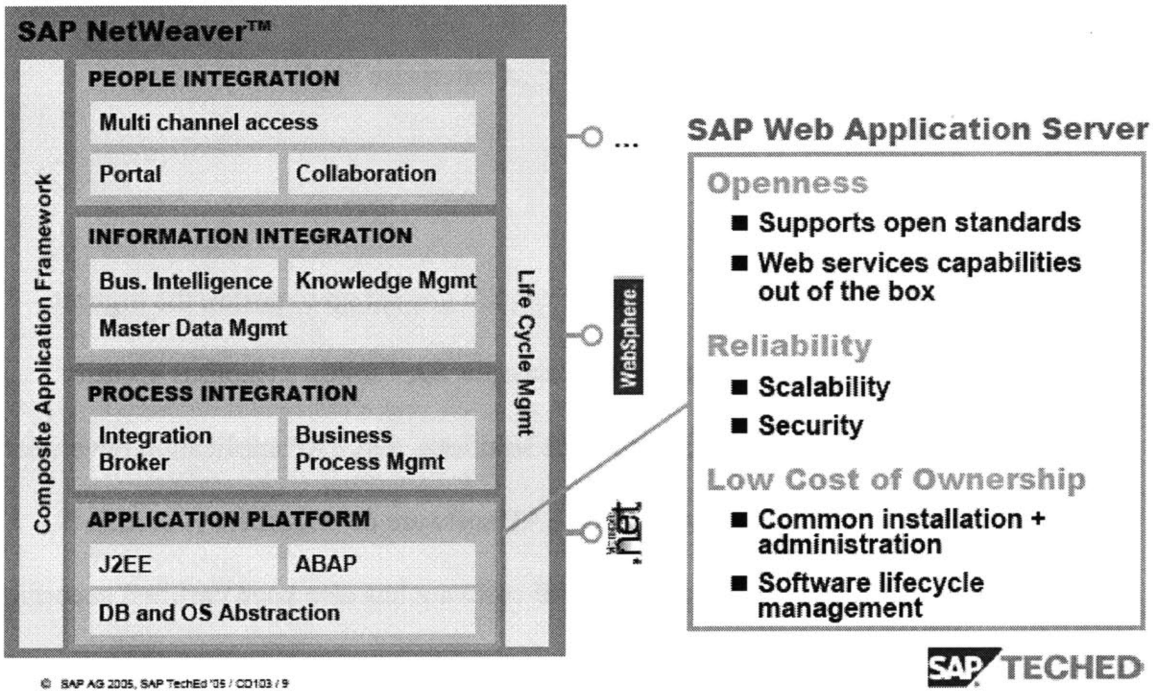


Figure 4-7: Web AS Java installation variant

4.8 SAP NetWeaver with WAS

Figure4-8 depicts the relationship between the SAP NetWeaver framework and SAP Web Application Server (WAS). In this chapter, the focus is on the 'Application Platform' component of NetWeaver. That component is nothing but the SAP WAS. From the earlier discussions it is clear that WAS has a robust but dependable architecture. J2EE and ABAP are the programming environments in WAS that help to connect applications with the database. Therefore, WAS is the technology foundation of NetWeaver.



© SAP AG 2005, SAP TechEd '05 / CD103 / 9

Figure 4-8: SAP NetWeaver and WAS relationship

WAS supports many open standards and provides web services capability to SAP. Web services will be discussed in detail in one of the forthcoming chapters. The diagram has a connector to .Net. This connection enables NetWeaver to make use of the rich functionalities of .Net. Also, in one of the later chapters, we will discuss about how SAP

Auto-ID Infrastructure (AII) seamlessly integrate with the other components of the NetWeaver.

4.9 Reliability of WAS

- Scalability

Scalability indicates the capability of a system to increase performance under an increased load when resources (typically hardware) are added. SAP WAS offers near linear scalability (with respect to CPUs/ blades) while offering excellent response times and is thus well suited for the most demanding of enterprise implementations.

- Security

As per [13], SAP solutions are built from the ground up to ensure the highest levels of security in the most sensitive environments. SAP follows rigorous security standards in the design and development of its solutions, and SAP application developers receive extensive software security training. SAP software development is certified according to the ISO 9001:2000 standard. SAP software has also been certified according to Information Technology Security Evaluation Criteria (ITSEC) Level E2 Medium.

In addition, SAP follows a thorough security response process, which includes a security bulletin to keep SAP customers up-to-date as new security threats and vulnerabilities are uncovered and addressed.

4.10 Dependable SAP WAS

It is important to conclude that software architecture of WAS satisfies many of the computing requirements of businesses. Its 3-tier, by enabling 'solution independence', makes SAP a preferred choice in enterprise computing. For example, businesses won't have to choose a different operating system or RDBMS or platform to implement SAP. Instead businesses could choose the WAS combination in which they have expertise in. This helps businesses to align SAP with their enterprise technology architecture. Many companies often find SAP a 'strategic fit' to their businesses.

Companies can rely on SAP WAS because it is the standard software platform for most of the software offerings from SAP. Many companies, in order to reduce the Total Cost of Ownership (TCO), tend to buy solutions from the same vendor. If SAP's other software offerings meet the business needs, companies would prefer to buy other software products from SAP. In such a scenario, the company already possesses the core SAP technology expertise. So the Implementation of the new software becomes easy. More importantly, integrating the new product with the existing SAP product becomes even simpler. This is mainly due to the standard WAS platform, common communication protocols and standard business interfaces. System maintenance, monitoring and trouble shooting activities of the new system are mere replication of the known system activities.

Survivability specifications are clearly defined for the SAP WAS. Many mechanisms exist in SAP software to ensure survivability. Many Fortune companies choose SAP because of SAP's reputation to meet the defined survivability specifications.

4.11 Strengths and Weaknesses of SAP architecture –Business perspective

Many of the strengths of SAP are weaknesses as well. This is a challenge when dealing with a SAP eco-system.

SAP provides dependable system architecture for businesses in terms of standard software platform, flexible adaptation, reliable survivability architecture and fault-tolerance. This thesis details these points in the later chapters.

These strengths require highly trained resources to maintain the SAP systems. This may work against the interest of small or mid-sized businesses. These businesses cannot afford to have these resources on-site. Often businesses are afraid to choose SAP systems mainly because of the perceived complexity and the TCO.

SAP's clear separation of architectural layers provides scalability for the applications and improves the experience of business-users by shortening system response time.

The same separation of layers makes even small changes to the system very difficult and expensive. Because of the higher TCO, even requests for small changes to system need to go through an elaborate approval process.

SAP system supports 'solution independence'. This enables companies to choose their own hardware, operating system (OS) and database system (DB) based on their own company policies to run SAP system.

'Solution independence' offers some benefits to the technology operations. However, during a SAP system crisis, even OS or DB related issues need to be dealt with SAP and not with the OS or DB vendor. Regardless of the nature of the issue, businesses need to

rely on SAP for everything. Due to inherent system complexity, only SAP can provide solution to any SAP related system issue. This reliance puts businesses under huge risk. Businesses cannot avoid paying SAP for system maintenance, usually 15%, year after year.

SAP's system installation variants allow companies to use either SAP's proprietary ABAP (Advanced Business Application Programming) or standard Java based infrastructure.
This poses a huge challenge to all size of businesses. Most of the existing systems are installed using ABAP. Some of the new installations are on Java. Even in the Java environment, many ABAP related system objects need to be accessed which requires Java professionals trained in ABAP. Still, most of the SAP applications are in ABAP. This forces businesses to rely too much on ABAP, a 4GL programming language, which is hardly used by anybody except SAP. This puts a huge constraint on required technical resources for SAP.

SAP's business functions have many functionalities which gives businesses many ways to configure the systems to be suitable for the business purposes.
Small and mid-size businesses don't need many of these functionalities but are forced to configure and hide these functionalities. This results in complexity and higher TCO.

SAP system provides standard business processes that meet many of the standards such as ISO and regulatory requirements such as WTO. For mature businesses, SAP offers many of the needed BPs and provides a complete enterprise solution.
For many growing companies, SAP business processes may be inflexible. They can even become barriers to the business growth.

SAP makes use of multiple application architectures and data models to run different functional modules such as Finance, Material Management and Sales & Distribution. This difference in architecture shows that different function modules were added to the SAP core systems at different times. This is how an evolution of an ERP system took place.
Multiple architectures and data models increase the complexity of SAP software management. System development becomes harder. System analysts and developers have to have a complete understanding of the SAP system making 'system development' work hard and expensive.

5. SAP Eco-System / Solution Portfolio

The dictionary definition of 'eco-system' is the following. "A system formed by the interaction of a community of organisms with their physical environment". In our case, the organisms are different SAP software solutions within a single enterprise environment. Based on the business need, the elements of the SAP eco-system vary among companies. Since its inception SAP has been offering many software solutions. Initially, they may be delivered as independent solutions but later they may be consolidated and delivered as a combined single solution. On the other hand, a single solution may also get converted into multiple variant independent solutions. Regardless, the global businesses and their changing needs really dictate these mergers and splits.

5.1 General-purpose applications

SAP General-Purpose Applications [13] cover many aspects of business operations. These systems provide support to standardized business processes of an enterprise. The business processes within a system are very integrated in nature and they are highly customized during the system implementation. SAP calls the following list as mySAP Business Suite.

1. mySAP ERP: This is a complete, scalable, and effective software for enterprise resource planning (ERP) with a flexible, open technology platform that leverages and integrates SAP and non-SAP systems. This is the traditional ERP software solution that many companies use around the world.

2. mySAP Customer Relationship Management (mySAP CRM): Enables customer-centric, end-to-end business processes; supports customer-facing departments such as marketing, sales, and service.

3. mySAP Product Lifecycle Management (mySAP PLM): Provides an integrated, single source of all product-related information needed for collaborating with business partners and supporting processes including product innovation, design and engineering, quality and maintenance management, and control of environmental issues.

4. mySAP Supply Chain Management (mySAP SCM): Helps to improve business and operations processes; Integrates collaboration, planning, execution, and coordination of the supply chain network.

5. mySAP Supplier Relationship Management (mySAP SRM): Simplifies and automates the procurement processes; Integrates strategic practices for supplier qualification, negotiation, and contract management through an analytical framework; Supports multi-channel supplier enablement.

5.2 Industry-specific applications

Industry-Specific Applications [13] are add-ons to the mySAP Business Suite applications or to SAP NetWeaver. They perform industry-specific business functions. For example, SAP Apparel and Footwear application is for the consumer products industry. These industry specific solutions are built on the standard SAP WAS platform and delivered and optimized for NetWeaver.

Aerospace & Defense	Automotive	Banking
Chemicals	Consumer Products	Defense & Security
Engineering, Construction & Operations	Healthcare	High Tech
Higher Education & Research	Industrial Machinery & Components	Insurance
Life Sciences	Logistics Service Providers	Media
Mill Products	Mining	Oil & Gas
Postal Services	Professional Services	Public Sector
Railways	Retail	Telecommunications
Utilities	Wholesale Distribution	Add-ons to mySAP Business Suite Applications or to SAP NetWeaver

Table 5-1: Available SAP industry solutions

5.3 SAP xApps composite applications

SAP xApps Composite Applications [13] perform functions spanning multiple applications, departments, and organizations. They deliver flexible business processes focused on improving user experience. Examples are SAP xApps for compliance and SAP xApps for analytics. The portfolio of SAP xApps applications can also come from SAP certified partners.

5.4 Solutions for small and mid-size enterprises

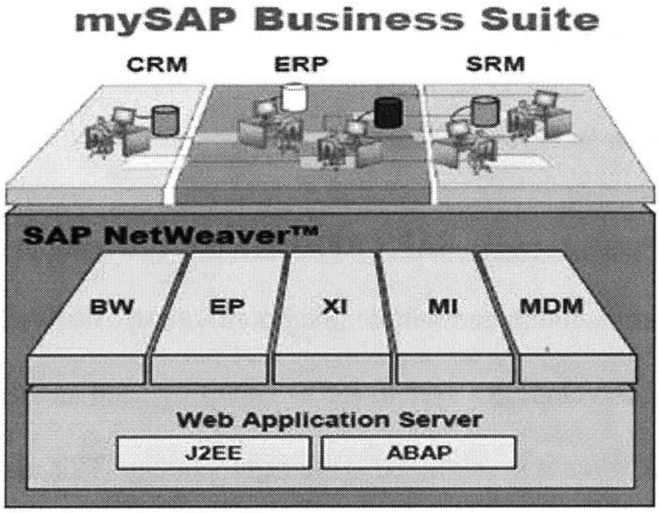
These solutions provide enterprise functionality to small and mid-size businesses.

SAP Business One [13]: SAP Business One is an easy-to-use business and operational management solution for emerging and dynamic businesses ranging in size from 10 to several hundred employees.

mySAP All-in-One [13]: Each qualified mySAP All-in-One partner solution is a prepackaged, industry-specific version of mySAP Business Suite with built-in content, tools, and methodologies for a cost-effective, turnkey implementation.

5.5 The NetWeaver stack

Recently, SAP combined many of its offerings into the NetWeaver stack [15]. To examine the SAP Eco-system, the NetWeaver stack is a good starting point. The mySAP Business Suite sits on top of the NetWeaver stack. Adjacent is a group of solutions and tools that can help integrate and optimize the transactional and master data from the mySAP Business Suite of applications. All these components are built on SAP Web Application Server (WAS) technology. As we know from our previous discussions (in chapter 4) that WAS can support J2EE or ABAP or both. Based on the available resources, organizations choose to install different variants of this combination.



Source: SAP AG (Taken from SAP NetWeaver Master Guide)

Figure 5-1: SAP NetWeaver stack

As per [15], apart from mySAP CRM, mySAP ERP and mySAP SRM, the NetWeaver product stack includes the following components.

SAP NetWeaver Business Intelligence (**BW**): Helps to integrate data from across the enterprise and transform the data to drive sound decision-making

SAP NetWeaver Portal (**EP**): Unifies critical information and applications across the enterprise to give users role-based views.

SAP NetWeaver Exchange Infrastructure (XI): Delivers open integration technologies to support process-centric collaboration across the enterprise value chain.

SAP NetWeaver Mobile Infrastructure (MI): Provides a mobile run-time environment based on open technology standards for building integrated mobile solutions.

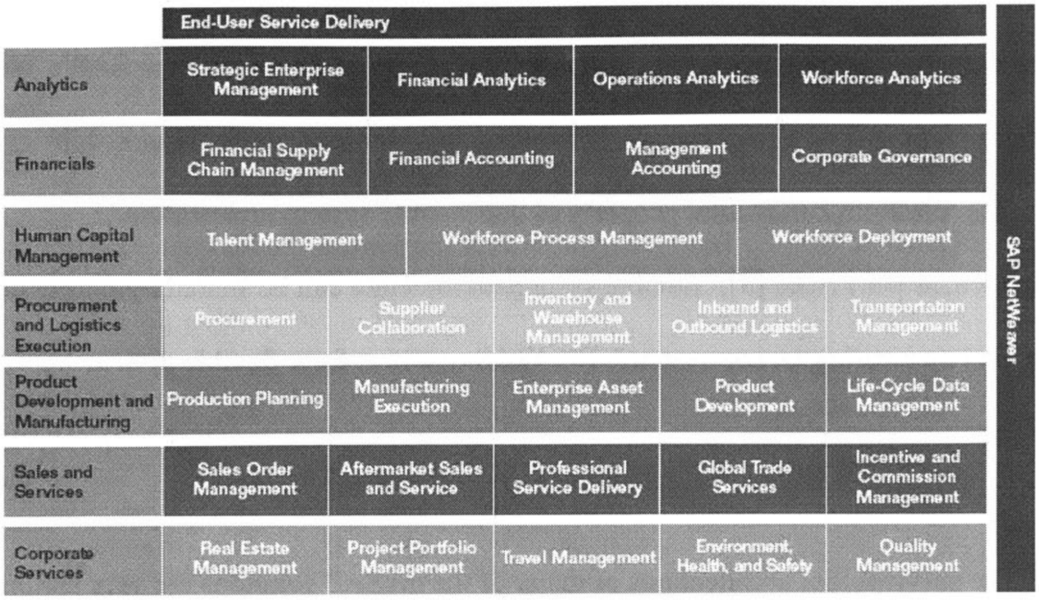
SAP NetWeaver Master Data Management (MDM): Ensures cross-system data consistency and helps to integrate business processes across the enterprise value chain.

5.6 Future addition to NetWeaver stack

SAP Auto-ID Infrastructure (AII): SAP's RFID Solution (Covered in a later chapter). This is an independent solution and will be integrated into the NetWeaver stack in the forth coming future. Developing a system for an emerging need and then integrating it to the NetWeaver stack when it is stabilized is the right strategy. This also ensures 'dependability' of the established software stack.

5.7 mySAP ERP is SAP R/3 (in a traditional sense)

mySAP ERP supports many of the critical business processes an enterprise need.



	End-User Service Delivery				
Analytics	Strategic Enterprise Management	Financial Analytics	Operations Analytics	Workforce Analytics	
Financials	Financial Supply Chain Management	Financial Accounting	Management Accounting	Corporate Governance	
Human Capital Management	Talent Management	Workforce Process Management	Workforce Deployment		
Procurement and Logistics Execution	Procurement	Supplier Collaboration	Inventory and Warehouse Management	Inbound and Outbound Logistics	Transportation Management
Product Development and Manufacturing	Production Planning	Manufacturing Execution	Enterprise Asset Management	Product Development	Life-Cycle Data Management
Sales and Services	Sales Order Management	Aftermarket Sales and Service	Professional Service Delivery	Global Trade Services	Incentive and Commission Management
Corporate Services	Real Estate Management	Project Portfolio Management	Travel Management	Environment, Health, and Safety	Quality Management

Figure 5-2: mySAP ERP solution map

As per [13], Figure 5-2 is known as mySAP ERP solution map. This map shows how various processes are supported in mySAP ERP. It is a depiction of multilevel blueprint of processes. This map helps to visualize, plan, and implement the mySAP ERP solution. It also helps to clarify the business solutions available and the business value that they can bring.

5.8 Overall Eco-System offers dependability to business

Previous sections of this chapter covered the complete SAP solution portfolio. The portfolio includes core business applications, industry dependent offerings and specialty cross functional solutions. These solutions are designed to meet the goals and objectives of the stakeholders of business within stakeholder's surrounding business context.

The biggest advantage of creating an eco-system with these solutions is that they all use a common software platform. That common platform is derived from SAP's Web Application Server (WAS) technology. Therefore, this common platform approach exhibits the 'platform bandwidth' property of dependable system architecture. Furthermore, as per Fricke [5], 'flexible system architecture can be shared by family of system; end up creating an eco-system'. So, WAS qualifies for a flexible system architecture.

The survivability architectures of many of the mySAP solutions are very similar to each other. However, there may be variations in the survivability specifications among solutions. Most of these solutions offer component level reconfiguration and architectural level reconfiguration. This shows 'flexible adaptation to change' another important characteristic of dependable system architecture.

5.9 Strengths of SAP eco-system – Business perspective

The biggest benefit of a SAP eco-system is 'integration of applications' and the use of the 'single standard software platform-WAS'.

- Single standard platform allows similar system administration activities, maintenance tasks, hardware needs and backup procedures. This reduces the cost of maintaining and running the eco-system infrastructure. Therefore, day to day technology operations become easier.
- Typically, organizations spend lot of money and resources on integrating applications. Integrating applications requires system level interfaces. These interfaces increase the

system complexity. To manage this complexity, more resources are necessary and this increases the TCO. However, the SAP eco-system enables and facilitates easy integration of solutions by creating standard interfaces. This brings the overall TCO down.

- Seamless integration of the components at the system level enables complete application integration at the business process level. This improves efficiency and effectiveness of organizational business processes. This enhances productivity of the business users and system stakeholders. As a result, employee morale improves and the customer service level goes up.
- Integration at various levels of an eco-system improves the ability to offer new functionalities to the business users and smoothes the change management process in an organization.
- SAP eco-system reduces the overall system risk and minimizes the disruptions to the ongoing business processes because it is dependable system architecture.

5.10 Weaknesses of SAP eco-system – Business perspective

The previous section 5.9 is applicable to an ideal eco-system. In the real business world, ideal situations never exist because the SAP eco-system is created over a period of time. This creates SAP System-of-Systems (SOS) instead of a perfect SAP eco-system. The reason to introduce SAP SOS here is to emphasize of the managerial and operational independence of SAP eco-system components. For example, mySAP ERP is typically owned by the Finance Department whereas mySAP CRM is owned by the Marketing and Sales. On the other hand, IT department looses control of how an IT (SAP) system should be run but is forced to maintain the SAP technical infrastructure such as hardware, and operating system software. Also, these stakeholders make decisions based on their own organizational understanding and department capabilities. This converts the perfect SAP eco-system into SAP SOS. Given the SOS nature, variety of weaknesses exists in maintaining and running a heterogeneous SAP eco-system.

- The notion of single standard platform appears to be true only at the surface. The eco-system solution uses different versions of platform WAS. Often, huge architectural differences exist among versions of WAS. Just managing this incompatibility could become a nightmare for IT departments.
- SAP keeps releasing patches for its products in the form packages. (SAP calls them 'Support Packages'). Different versions of WAS require different level of support packages. These support packages have unique characteristics and they could easily break applications. Major strength of SAP eco-system is application integration. However, during the application of support packages, the same 'application integration', due to heavy dependency between components, causes the application to break. This can be fixed by rigorous testing. However, this may convert a small patch application in to a 'system project'. These projects require participation of resources from a variety of departments. Overall, this increases the TCO.
- This 'software update' complexity and costs, many times, force organizations to stay on with their older SAP versions. After some time, the SAP eco-system could become heterogeneous complex System-of-System setup.

To manage these pitfalls, companies should adapt the following best business practices.

- Considering SAP Program Management as a separate department is useful. The goals and objectives of this department should be set by stakeholders from departments such as Finance, IT, Marketing and Sales and Human Resources.
- Resources (technical/business/change management/project managers) should be dedicated to maintain the SAP eco-system year around
- An elaborate communication strategy is required to notify the business users about the SAP updates and happenings.
- Organizational SAP budget should be allocated from different departments instated from one single department. This distributes the control over decision-making.
- Implementation of SAP, most of the time, changes the way organizations do business. So, it is important to have a strong change management team involved in communicating and implementing polices and procedures.
- SAP Program Management organizational structure should be created in a way to meet the needs and requirements of SAP eco-system and its stakeholders.
- Some of the important goals of SAP Program Management are the following:
    - Keeping the 'change' in SAP eco-system always in control
    - Reducing the system complexity with the use of best business practices, tools, and system methodologies.

5.11 Stages of SAP system implementation

5.11.1 Initial implementation

Overall Complexity: High; Resources Required: Large (Consultants and system experts are needed temporarily); Cost: High.

This is the most crucial stage and the system success is based on this stage. Initial implementation costs are quite high. An analysis phase should include identifying crucial business processes (BP), BP flow mapping to SAP BPs and GAP Analysis. GAP analysis helps to identify the process gaps. (Organization's AS-IS and TO-BE). Typical recommendations from the GAP analysis and their implications are as follows.

Recommendations	**Organization's Willingness to change**	**Short-term/Long-term Implications**
Adapt to SAP recommended BP	Very low; High resistance may come from BP owners	Complexity: High / Low TCO: Low / Low
Custom Interface	Very high; low resistance from BP owners	Complexity: High / High TCO: High/High
3rd party software	Medium; resistance may arise from BP owners	Complexity: Low/ High TCO: High/High

Table 5-2: Stages of SAP Implementations

The analysis phase is followed by design, construction and implementation phases of a typical system implementation methodology.

5.11.2 System stabilization stage

Overall Complexity: Medium; Resources Required: Medium (Some of the consultants may be rolled-off); Cost: Medium.

This stage comes right after the initial implementation stage. Some of the external resources are still retained.

5.11.3 Operational stage

Overall Complexity: Low; Resources Required: Few (only the trained organizational resources); Cost: Low.

This stage comes right after the system stabilization stage. All external resources are rolled-off. This stage continues until the organization decides to go for a support pack upgrade or for a major system upgrade.

6. Stakeholder Analysis

As per [19], stakeholder analysis is a process used to catalog, rank, and assess the positions of the individuals, groups, and organizations (internal and external to an organization) affected by or interested in the organization, issue, program or policy being considered. In our case, the program refers to the SAP Project/Program. Performing stakeholder analysis helps to improve the implementation and operation architectures of the SAP system. As we discussed in the earlier chapters, both system implementation and operation have major impacts on the system dependability.

6.1 Stakeholders

Here is the list of stakeholders who contribute to SAP eco-system's dependability. This list comes from author's own professional experience.

External to an organization or Business:

SAP AG, System Integrators, 3rd Party Software Providers, Hardware Vendors, Component Suppliers.

Internal to an organization or Business:

SAP Software Owner, Executive Committee, Project Sponsor, Project Manager, Technical Team Lead, Functional Team Lead, System Developer, System Administrator, Change Manager, System Integrator, Internal consultants, Security Specialist, Internal Auditor, End-Users, Business Analysts, System Analyst, and Trainer.

Following sections will describe the relationship between these different stakeholders.

6.2 Hierarchical relationship

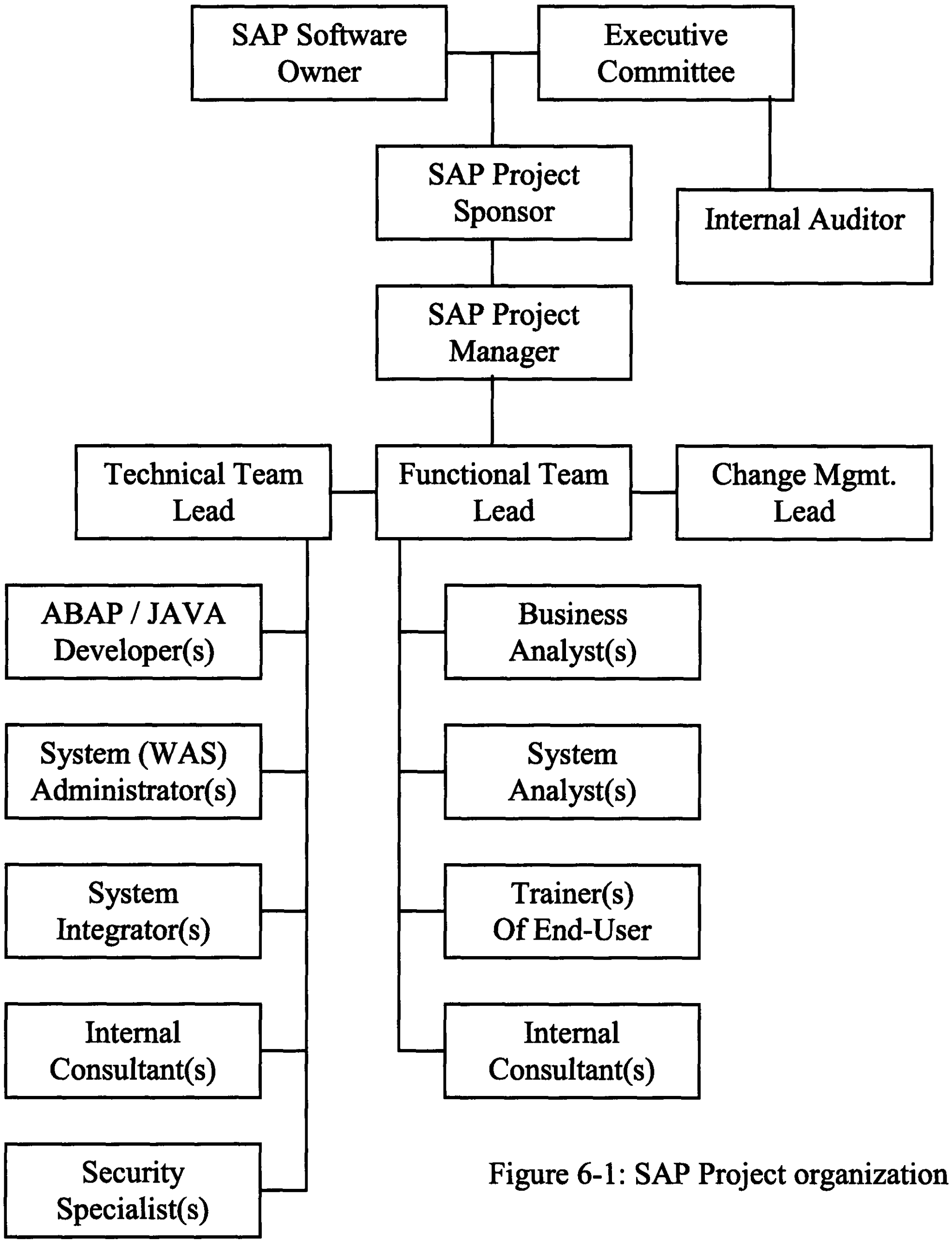


Figure 6-1: SAP Project organization

This hierarchical relationship shows the typical reporting relationships within a SAP project/program management organization. From author's professional experience, this hierarchy seems typical; however, slight variations may be possible in some organizations.

6.3 Value flow mapping

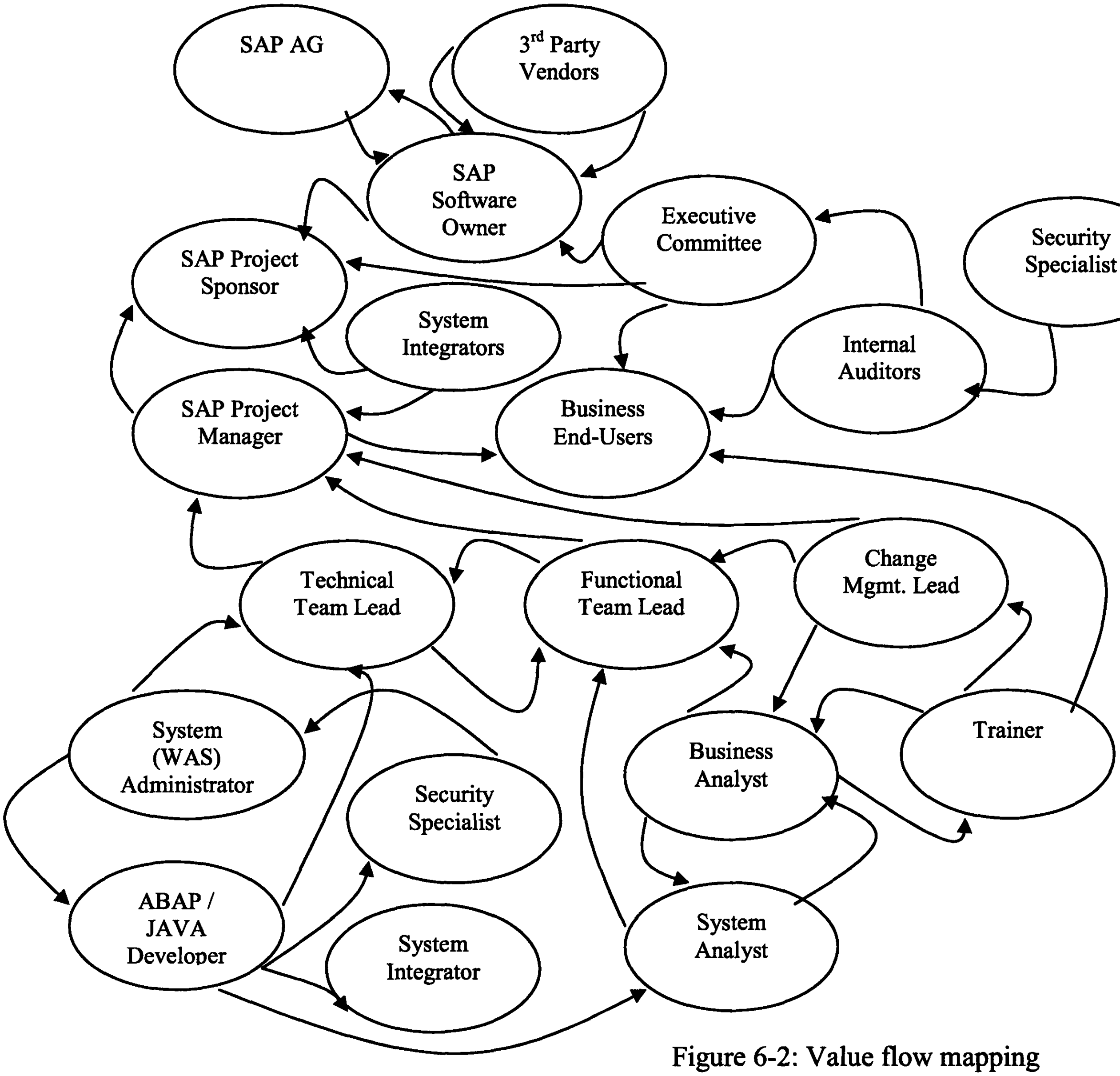


Figure 6-2: Value flow mapping

The arrows indicate how value flows from one stakeholder to another. The stakeholder at the tail end of the arrow is provider of a certain value. On the other hand, the stakeholder at the end of the arrow receives the value. In many occasions, a stakeholder may provide and receive value from another stakeholder as well.

Let's exemplify the value flow map through the eyes of SAP Project Manager (PM).

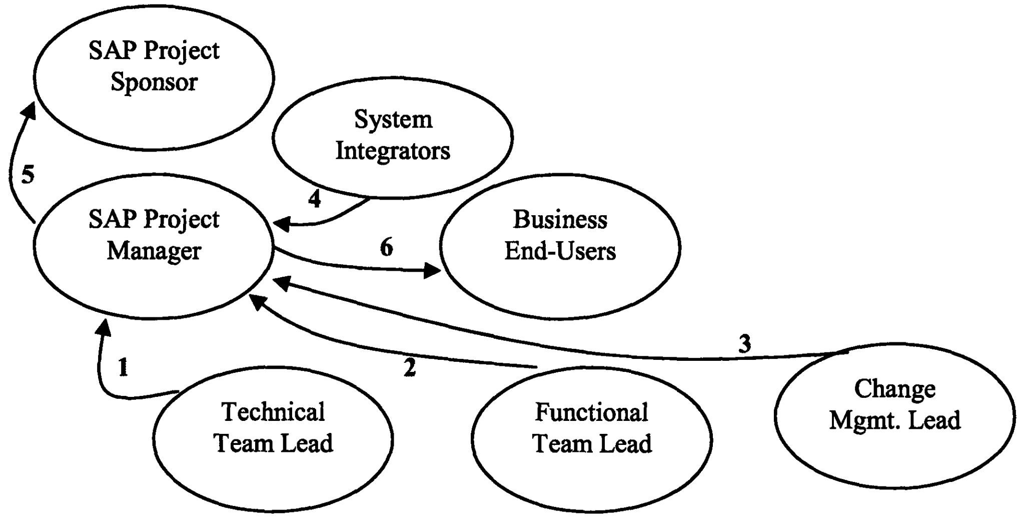


Figure 6-3: Example of value flow

1. Technical Team Lead manages the technical team and provides 'Infrastructure and system development support' value to PM.
2. Functional Team Lead manages the business team and provides 'Business process management and implementation support' value to PM.
3. Change Management Lead provides 'Change management assistance' value to PM.
4. System Integrators provide 'Implementation partner support' value to PM.
5. The PM provides 'SAP system and project management' value to SAP Project Sponsor.
6. The PM provides 'SAP real-time data, processes, and applications' value to the Business End-Users.

6.4 SAP System: An enterprise context

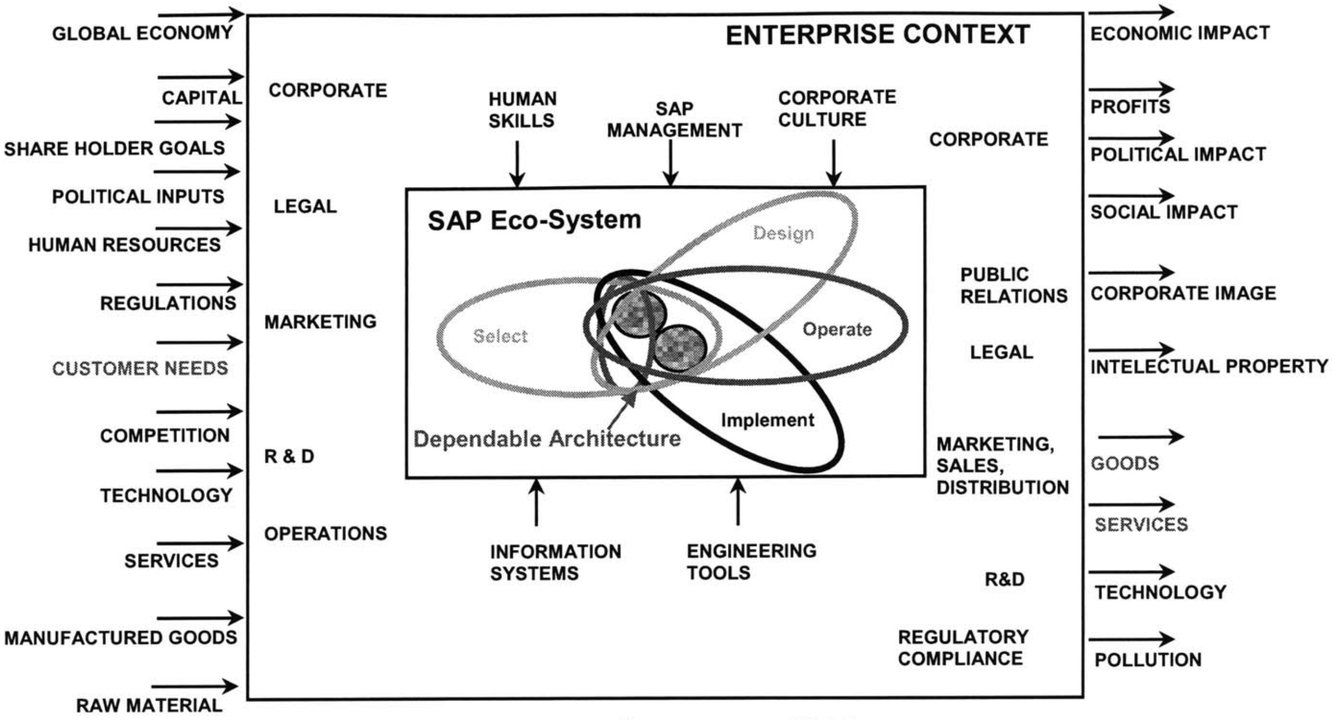


Figure 6-4: Enterprise context of SAP system

Figure 6-4 is adopted from MIT course ESD.34 and originally authored by Professor Edward Crawley. As per the picture, the SAP eco-system sits within an enterprise. In some cases, there is a possibility that the SAP eco-system may sit in between two or more enterprises. The enterprise brings many of its internal strengths together to select, design, operate and implement a SAP eco-system. Important internal strengths include SAP project/program organization, human skills, other information systems and engineering tools. Obviously, the corporate culture has a strong impact on the SAP eco-system as well. Change management is the tool that helps mitigate the risks posed by the corporate culture. In sum, the way the enterprise operates has a strong influence on making the SAP eco-system as dependable system architecture for an enterprise.

6.5 Societal context

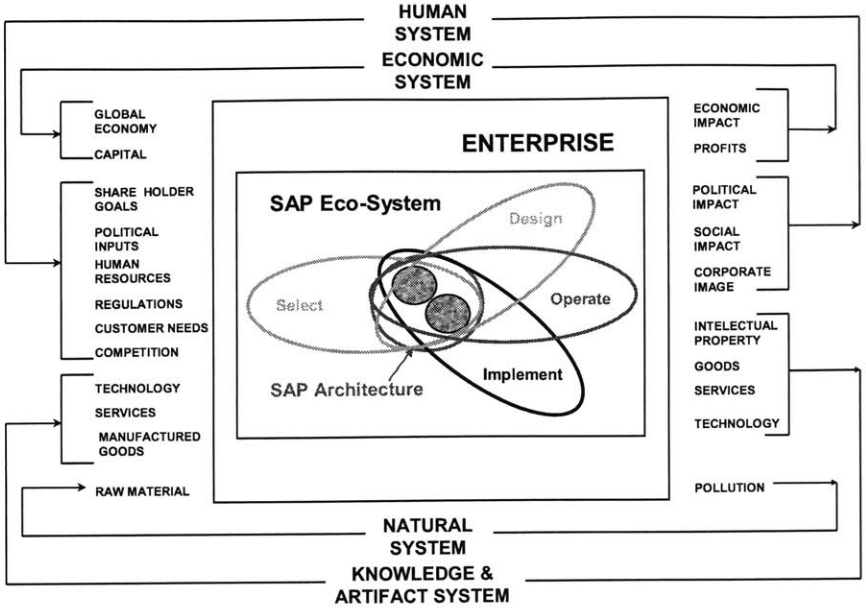


Figure 6-5: Societal context of SAP system

Figure 6-5 is adopted from MIT course ESD.34 and originally authored by Professor Edward Crawley. As per the picture, influences on SAP eco-system architecture often come from outside of the enterprise boundary. For example, global economy, regulations and technology may impact the architecture. Therefore, the elements outside the enterprise are often critical to the architectural success. It is safe to conclude that 'dependable system architecture' must be capable of handling the impact posed by the variables that are external to an enterprise. SAP eco-system includes components such as mySAP ERP, mySAP PLM etc which are designed to meet the challenges posed by the external elements. As a result, from a societal perspective, SAP eco-system meets many of the criteria that are needed to make it 'dependable system architecture'.

6.6 Analysis into SAP stakeholders

SAP project organization requires highly skilled professionals to manage and run the SAP eco-system. Special technical skills are often required to implement new processes in the system. Traditionally, SAP program organizations hire external consultants to perform some of these activities. If the business is growing, then an organization needs an annual budget to spend on special consultants who offer critical services. The following paragraph explains the major difference between system analysts in the United States and Germany where SAP originated and developed.

In the US, system analysts are split into technical analysts and business analysts. Business analysts have limited knowledge of SAP programming and technical analysts have limited knowledge about business processes. This poses a major challenge in implementation of SAP in the US. In Germany, there is not much difference between business and technical analysts. System analysts know both business processes and SAP programming.

In the US, segregation of duties between both business and technical analysts contribute to major communication gaps that result in inefficient SAP implementations and cost over runs. In US organizations, major politics emerge between these analyst groups. This becomes a big road block in the smooth implementation of SAP. Many German companies directly adapt SAP's BPs as their organization's BPs thereby reducing the TCO over the long run. US companies resist adapting to SAP's BPs. This creates major GAPS in the overall organization's processes and results in a higher TCO. An exception to this is the SAP implementation of Microsoft Corporation. Microsoft completely adapted to SAP's BPs and runs its complete business operations using SAP and it has realized efficiency and lower TCO in the use of SAP.

As per chapter 4, SAP uses its proprietary programming language ABAP for the development of its application components. However, programmers can use ABAP or Java to develop external interfaces or reports. ABAP is more like business programming where as Java is a system and application programming environment. The aging ABAP community possesses deep knowledge of the SAP system but resists migrating to Java. This is causing big problems for the industry. When SAP talks about Java, it also faces a huge hurdle of convincing a huge ABAP developer community to make use of Java so that their previous SAP system knowledge could be exploited. On the other hand, it is difficult for SAP to attract new programmers into SAP ABAP and attract new programmers into SAP Java that requires an in depth knowledge of ABAP components as well. This is going to be a huge challenge for SAP moving forward. This is one of the reasons why many US companies outsource their SAP programming to countries such as India and China.

6.7. SAP Business value proposition to stakeholders

While the earlier chapters covered many of the strengths of SAP, this chapter focuses on a very high level on business value proposition that SAP brings to its stakeholders. Use of SAP eco-system brings the following benefits [13].

Better decision-making:

- Access the right information in real time in order to identify business concerns early: SAP offers real time access to business information. There are thousands of standard business reports available real time in the SAP system. For example, General Ledger balance sheet can be run at any time of the day.
- Pursue potential market opportunities: Business reports and transactions often combine together information from many business areas. This helps companies to pursue market opportunities.

Improved productivity, efficiency, and responsiveness:

- Extend the reach of business processes to connect more people in real time: Connecting people, information and process real time is a key to the success of a business. SAP helps accomplish this inside and outside the enterprise.
- Provide easy access to consolidated views of business processes: The business processes in SAP are tightly integrated between functional areas such as Finance, Materials Management and Sales & Distribution. This facilitates consolidated views of business processes.

Reduced costs through increased flexibility:

- Deploy additional business functionality when needs evolve: SAP offers different configuration options at the system level and at the architectural level. This helps to reduce costs and facilitate change. Additional functionalities can be easily delivered to business-users with minimal effort.

- Protect and leverage the existing IT investments by integrating SAP with non-SAP solutions: If the systems are integrated appropriately, it will help protect the investment that was made in SAP and non-SAP solutions.

Adaptability to business change:

- Integrate end-to-end business processes: By integrating end-to-end BPs, the cycle time between processes decreases. This improves adaptability to business change.
- Take advantage of the Web-based technologies: When appropriately developed, most of the SAP application functionalities can be offered over the internet. This will help in connecting with business partners, vendors and customers.

Reduced risk:

- Implement additional mySAP ERP components with minimal disruption to ongoing business processes: If the mySAP ERP is successfully implemented and running, it is easy to add additional mySAP ERP components to the eco-system. During the addition, on-going business processes are not impacted.

Higher staff morale and productivity:

Boost staff morale and increase productivity: Business users can access the information they want and when they want it. This reduces frustration and improves the workplace morale. This improves individual productivity of the users and the overall productivity of the business as well.

7. SAP Support Solution

As we discussed in Chapter 2, survivability architecture and self-healing architecture are vital in establishing dependable system architectures. These architectures have strong relationships with system and service level management. Maintenance of a SAP eco-system requires complex processes, sophisticated tools and well-trained people.

7.1 SAP Solution manager

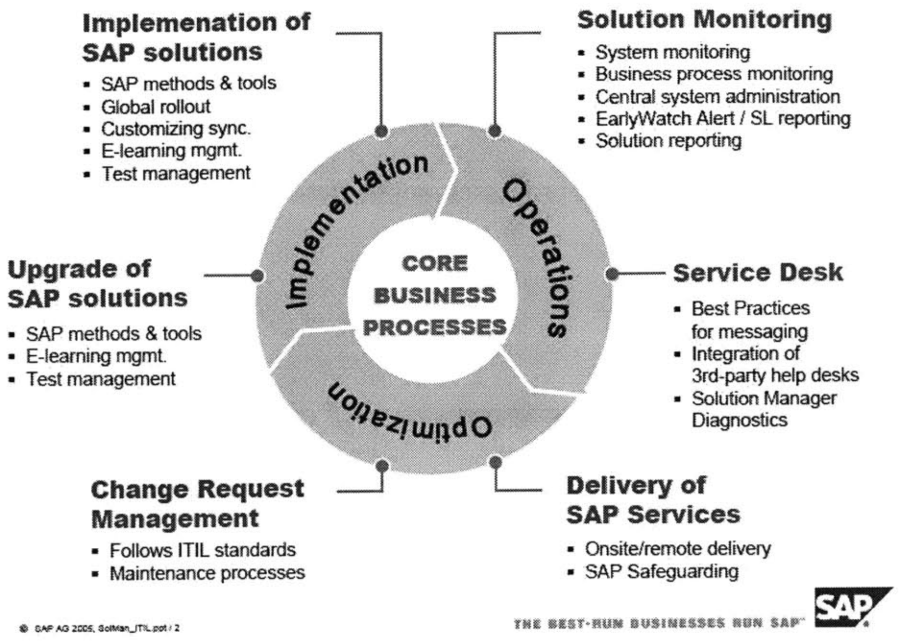
To simplify the eco-system management, SAP offers a system called 'SAP Solution Manager'. This is built on SAP WAS technology platform. As per [25], SAP Solution Manager is a centralized solution management platform that provides the tools, the integrated content, and the gateway to SAP that an enterprise needs to implement, support, operate, and monitor its SAP solutions. SAP Solution Manager runs in the SAP eco-system landscape and facilitates the technical support of distributed SAP systems.

As per [25], SAP Solution Manager:

- Minimizes the risk and increases the reliability (survivability) of SAP eco-system solutions.
- Helps to reduce TCO throughout an SAP solution's life cycle.
- Helps companies to manage their core business processes and link business processes to the underlying IT infrastructure.
- Supports both SAP and non-SAP software and helps companies to get more from their existing SAP and IT investments.

7.2 SAP Solution manager scenarios

The SAP Solution Manager supports the following scenarios [26]:



Implemenation of SAP solutions
- SAP methods & tools
- Global rollout
- Customizing sync.
- E-learning mgmt.
- Test management

Solution Monitoring
- System monitoring
- Business process monitoring
- Central system administration
- EarlyWatch Alert / SL reporting
- Solution reporting

Upgrade of SAP solutions
- SAP methods & tools
- E-learning mgmt.
- Test management

Service Desk
- Best Practices for messaging
- Integration of 3rd-party help desks
- Solution Manager Diagnostics

Change Request Management
- Follows ITIL standards
- Maintenance processes

Delivery of SAP Services
- Onsite/remote delivery
- SAP Safeguarding

© SAP AG 2005, SolMan_ITIL.ppt / 2

THE BEST-RUN BUSINESSES RUN SAP™

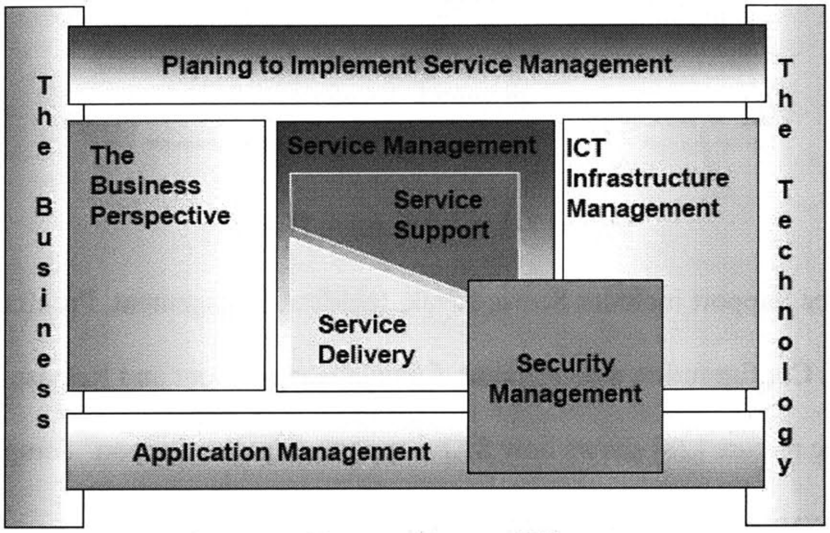
Figure 7-1: Scenarios of SAP solution manager

1. Service Desk: Provides infrastructure for organizing and operating a solution-wide support organization.
2. Implementation of SAP Solutions: Supports the implementation of projects and the definition of template projects.
3. Upgrade of SAP Solutions: Supports the implementation of projects and the definition of upgrade projects.
4. Change Request Management: Provides processes for managing change requests, project management, and change logistics. (Completely aligned with ITIL)
5. Solution Monitoring Provides support for functionality such as Service-Level Reporting, Early Watch Alert, Business Process Monitoring and System Monitoring.

6. Delivery of SAP Services: Point of access to SAP remote services, on-site services and self services.

7.3 IT Infrastructure Library (ITIL) support

ITIL [29] is the most widely accepted approach to IT service management in the world. ITIL provides a cohesive set of best practice, drawn from the public and private sectors internationally.

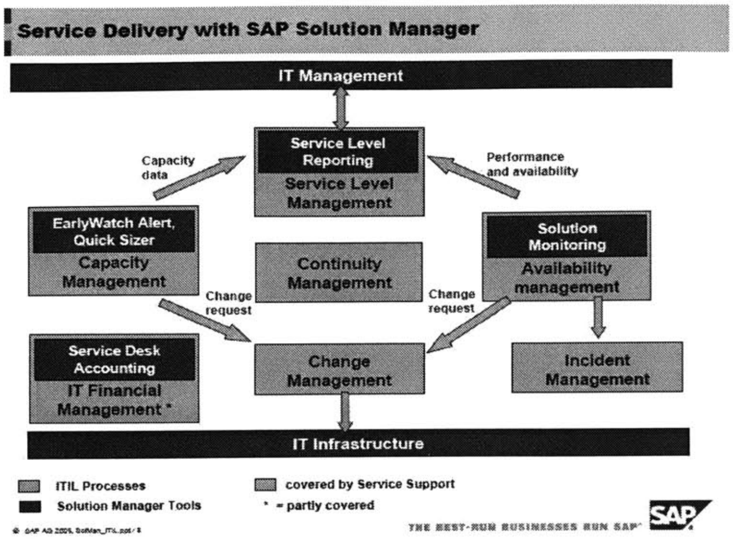


Source: Office of Government Commerce (OGC)

Figure 7-2: ITIL Framework

Figure 7-2 shows the ITIL framework [27] from OGC. As indicated above, Service Support and Service Delivery are considered to be the heart of the ITIL framework for IT Service Management.

Service Delivery includes Service level management, Financial management for IT services, Capacity management, IT service continuity management and Availability management. The following picture [28] shows how SAP supports 'Service Delivery' component of ITIL with its various proprietary tools.



Service Delivery with SAP Solution Manager

IT Management

Capacity data

Service Level Reporting
Service Level Management

Performance and availability

EarlyWatch Alert, Quick Sizer
Capacity Management

Continuity Management

Solution Monitoring
Availability management

Change request

Change request

Service Desk Accounting
IT Financial Management *

Change Management

Incident Management

IT Infrastructure

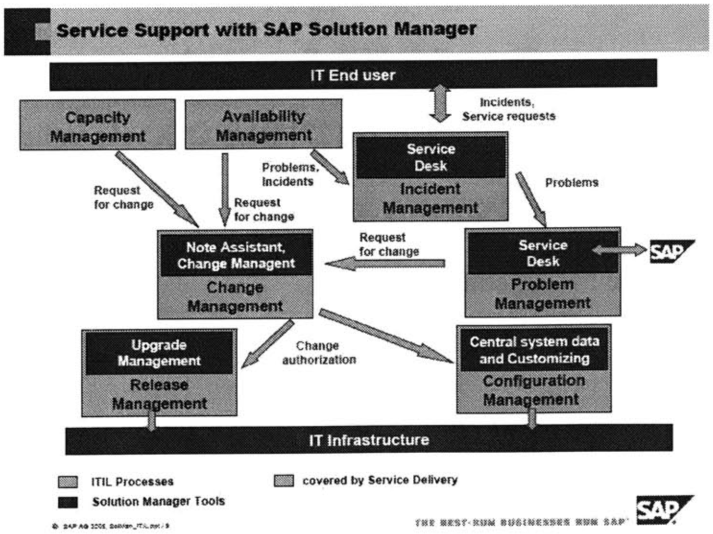
▨ ITIL Processes
■ Solution Manager Tools
▨ covered by Service Support
* = partly covered

© SAP AG 2005, SolMan_ITIL.ppt / 8

THE BEST-RUN BUSINESSES RUN SAP

Figure 7-3: SAP Service Delivery

Service Support includes Service desk, Incident management, Problem management, Configuration management, Change management and Release management. The following picture [28] shows how SAP supports 'Service Support' component of ITIL with its various proprietary tools.



Service Support with SAP Solution Manager

IT End user

Incidents, Service requests

Capacity Management

Availability Management

Service Desk
Incident Management

Problems, Incidents

Request for change

Request for change

Problems

Note Assistant, Change Managent
Change Management

Request for change

Service Desk
Problem Management

Upgrade Management
Release Management

Change authorization

Central system data and Customizing
Configuration Management

IT Infrastructure

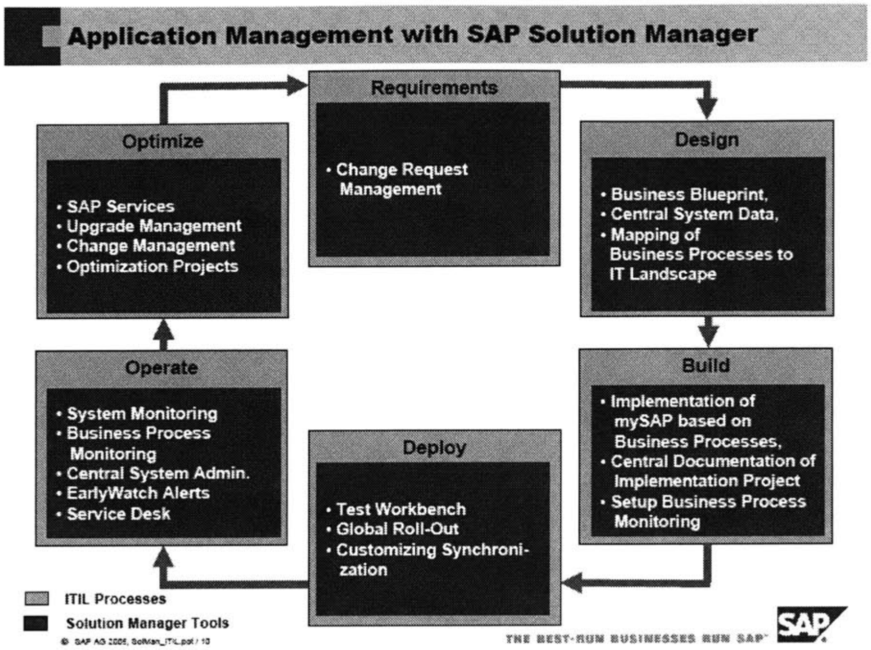
▨ ITIL Processes
■ Solution Manager Tools
▨ covered by Service Delivery

© SAP AG 2005, SolMan_ITIL.ppt / 9

THE BEST-RUN BUSINESSES RUN SAP

Figure 7-4: SAP Service Support

Application management [27] provides an outline of the Application management life cycle and is a guide for business users, developers and service managers of how application can be managed from a service management perspective.



Application Management with SAP Solution Manager

Requirements
- Change Request Management

Design
- Business Blueprint,
- Central System Data,
- Mapping of Business Processes to IT Landscape

Build
- Implementation of mySAP based on Business Processes,
- Central Documentation of Implementation Project
- Setup Business Process Monitoring

Deploy
- Test Workbench
- Global Roll-Out
- Customizing Synchronization

Operate
- System Monitoring
- Business Process Monitoring
- Central System Admin.
- EarlyWatch Alerts
- Service Desk

Optimize
- SAP Services
- Upgrade Management
- Change Management
- Optimization Projects

▨ ITIL Processes

■ Solution Manager Tools

© SAP AG 2006, SolMan_ITIL.ppt / 13

THE BEST-RUN BUSINESSES RUN SAP™

Figure 7-5: SAP Application Management

The picture [28] shows how SAP supports ‘Application Management’ component of ITIL with its various proprietary management techniques, processes and tools.

7.4 Support solution improves dependability

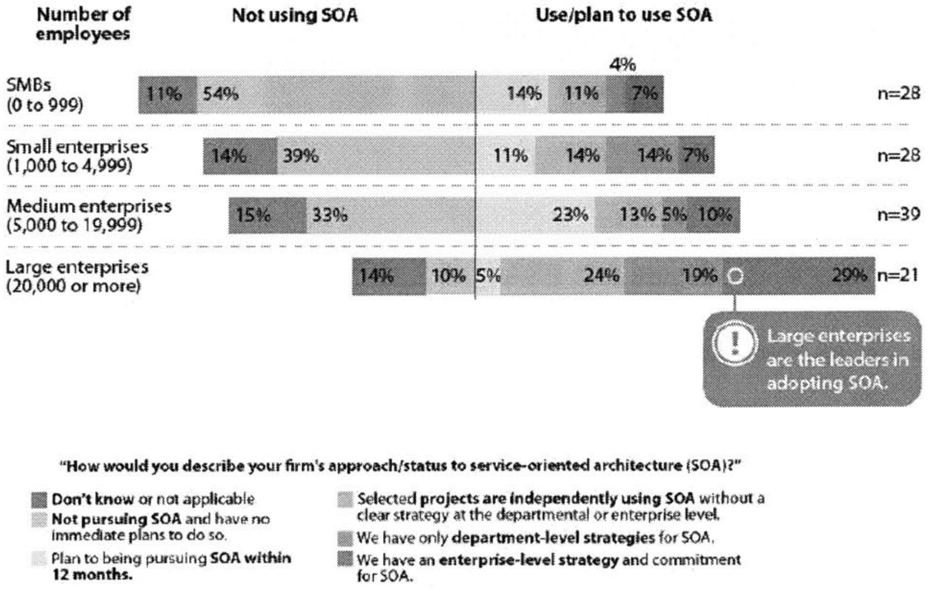
Survivability specifications of systems decide the level of system support required in an eco-system. SAP eco-system includes systems that perform a variety of functions. Their survivability specifications are very different. To manage this complexity, SAP offers SAP Solution Manager which is also part of the SAP eco-system. This improves the dependability of the whole eco-system.

8. Emerging Trend: Enterprise Services Architecture (ESA)

As per [21], SAP defines Service Oriented Architecture (SOA) as the following. "SOA is an architectural style to achieve loose coupling among interacting software agents. A service is a unit of work done by a service provider to achieve desired results for a service consumer." One of the key ingredients [21] in SOA is Web Services (which is a set of protocols and standards). Composite applications, event-driven, and process-driven are the different styles of building applications within SOA. According to SAP, Enterprise Services Architecture (ESA) is a business driven approach to SOA and SAP's open architecture for adaptive business solutions.

8.1 Service oriented architecture (SOA)

Large enterprises are adopting SOA to improve the efficiency and effectiveness of their underlying business processes. In the following figure, Forrester Research shows that 67% of large enterprises are considering and/or pursuing SOA.



Large enterprises are the leaders in adopting SOA.

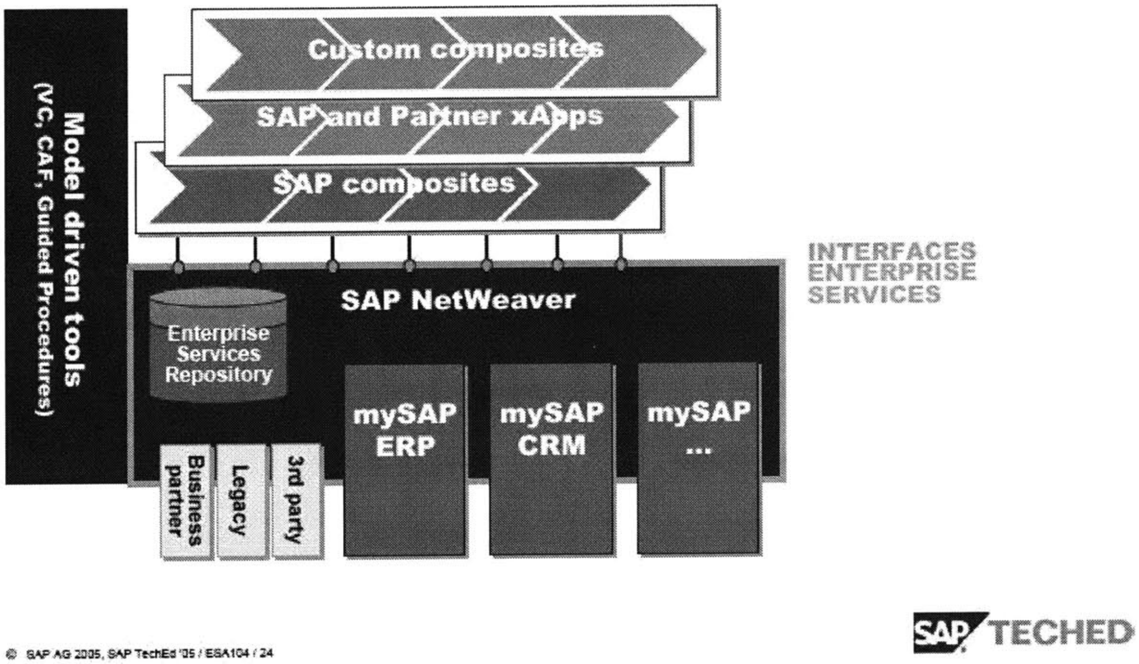
Base: 116 North American decision-makers familiar with programming technologies, application software architecture, and application platforms (percentages may not total 100 because of rounding)

Figure 8-1: Use of SOA

In response to the emerging trend, SAP has come up with ESA as its answer to SOA. The following paragraphs explain the ESA in detail.

8.2 ESA and SAP NetWeaver

According to SAP, ESA and SAP NetWeaver are not synonymous. ESA is the blueprint to define SOA in SAP, where as SAP NetWeaver provides the tools that can be used to follow this new architecture concept.



© SAP AG 2005, SAP TechEd '05 / ESA104 / 24

Figure 8-2: ESA with SAP NetWeaver

As per the above figure, SAP NetWeaver includes Enterprise services repository and Model driven tools. These two components are essential to implement ESA. Enterprise services repository catalogs all the available SAP services and the model driven tools help to create ESA based applications. As a result, building composite applications and service enabling SAP applications are possible.

8.3 Key elements of ESA

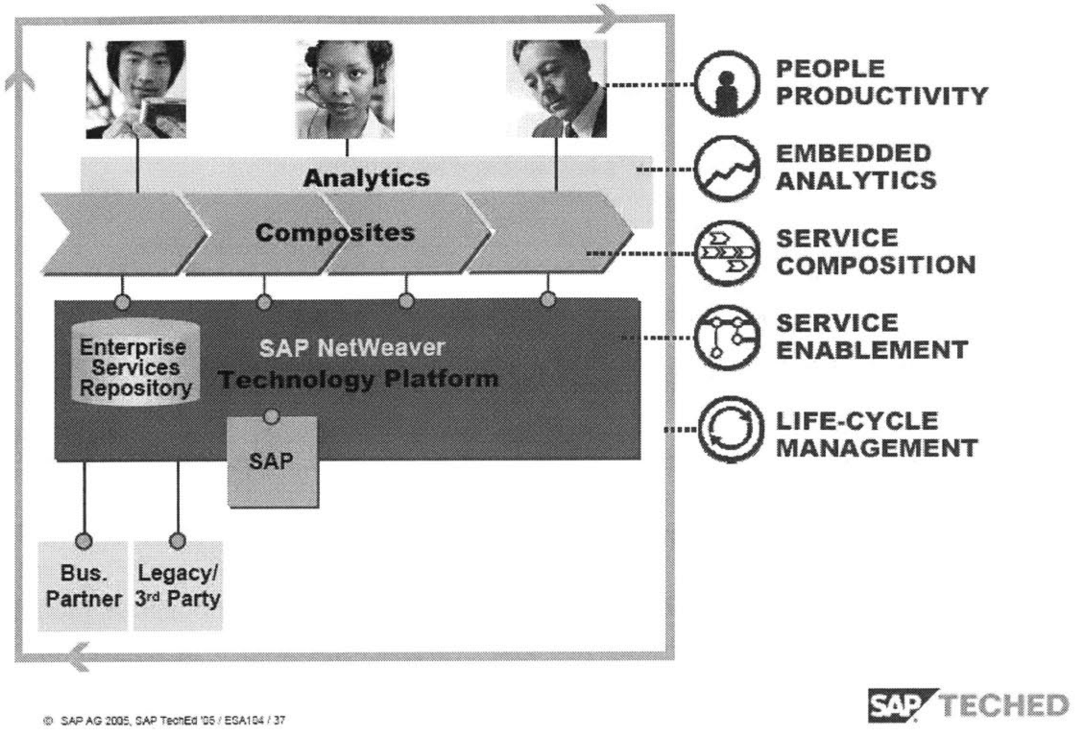


Figure 8-3: Key elements of ESA with SAP NetWeaver

As per [21], the key elements that support adaptation of Enterprise Services Architecture are the following.

- People productivity: Improves because people and processes are linked.
- Embedded Analytics: Business intelligence receives real-time data. Milestone monitoring and dashboards are possible.
- Service Composition: Common services repository to a whole enterprise.
- Service Enablement: Business, configuration, and execution models are available.
- Life-Cycle Management: Configuration through SAP Solution Manager is possible. SAP Solution Manager concept has been discussed in chapter 10.

8.4 ESA architectural dependability

Having discussed about ESA in detail, it is obvious that SAP addresses the emerging trend effectively. Especially, as per [24], SAP's ESA is considered to be one of the best in the industry. This shows SAP's technology leadership.

At the same time, ESA has been delivered with SAP's flagship product NetWeaver. Thus SAP customers don't have to rely on outside software vendors to get SOA delivered. To expose its application components as services, SAP uses 'Services Repository'. Then to combine the services for creating complex business processes, SAP offers development tools. This helps businesses in enabling flexibility of business model, organization and technology. As a result, flexibility improves readiness to change and ability to change rapidly. In sum, ESA enhances flexible adaptation to change which ultimately improves the 'dependability of the SAP system architecture'.

8.5 Competition to provide SOA

While the ultimate platform leader remains to be seen, this research examines the three major players emerging in the SOA standards: SAP NetWeaver, Microsoft's .NET, and IBM WebSphere.

ESA via SAP NetWeaver

SAP NetWeaver is a comprehensive application integration platform. It provides an enterprise services repository and a set of development tools for selecting appropriate services to create new or composite business applications. SAP NetWeaver unifies integration technologies into a single platform and is pre-integrated into business applications, thus reducing the need for custom solutions. The platform is based on industry standards and can be extended with commonly used development tools such as Java 2 Platform, J2EE, Microsoft .NET, and IBM WebSphere.

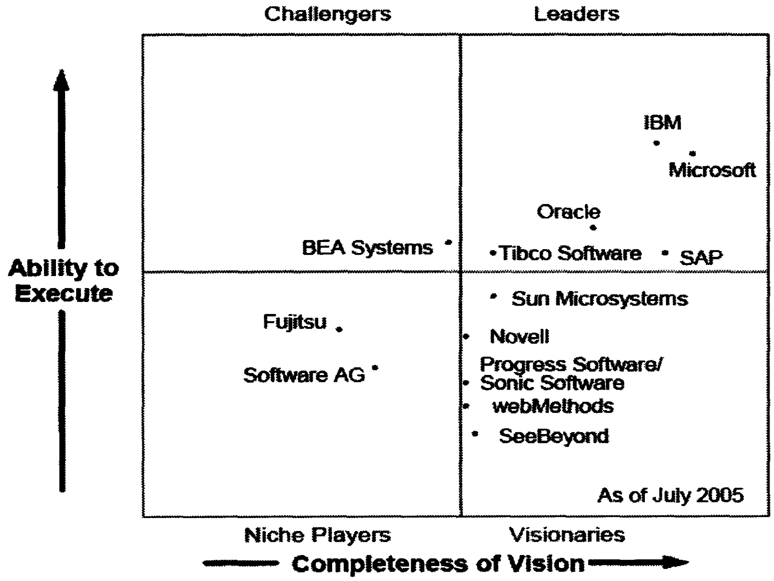
SOA via Microsoft .NET

As per Microsoft [22], .NET is Microsoft's web services strategy to connect information, people, systems, and devices through software. Integrated across the Microsoft platform, .NET technology provides the ability to quickly build, deploy, manage, and use connected, security-enhanced solutions with web services. These web services form the basis for the much wider SOA framework. Apart from web services, .NET based SOA includes policies, practices, and frameworks by which the right services are provided and consumed.

SOA via IBM WebSphere

As per IBM website [23], IBM WebSphere software provides businesses with integration capabilities designed to maximize both flexibility and responsiveness. WebSphere Application Server, the Java 2 Enterprise Edition (J2EE) and Web services-based application server are currently available on number of platforms and provide a robust, proven environment for Java applications. The WebSphere Application Server is the foundation of IBM's Service-Oriented Architecture and offers a comprehensive set of software tools supporting the full application development lifecycle for traditional mainframes, Java and composite applications with sophisticated connectors.

Gartner Research [24] places SAP, Microsoft and IBM in the Leaders Quadrant for Web Services Platforms.



Source: Gartner (July 2005)

Figure 8-4: Gartner's Magic Quadrant: Web services platform

Gartner [24] defines 'Leaders in Web Services Platforms' as "high-viability vendors with proven track records in Web services, as well as vision and business investment that indicate they are well-positioned for the future". Leaders do not necessarily offer the best products for every customer project; however, they provide solutions that offer relatively lower risk.

8.6 Recommendation to businesses

ESA is highly recommended for companies with an existing SAP eco-system. ESA uses SAP's core technology and it is completely integrated and delivered on NetWeaver. So, only existing SAP customers can reap most benefits out of ESA. On the other hand, ESA is not recommended to non-SAP user companies. Since ESA exploits many features of the Java or .Net based technologies, it is better for non-SAP companies to choose either IBM WebSphere or Microsoft .Net based SOA.

9. Case Study: SAP Auto ID Infrastructure (AII)

9.1 Introduction

Radio Frequency Identification (RFID) is one of the most discussed technologies today. It is an evolving technology and going through continual changes. Businesses are, however, taking serious note of RFID and its impact on the bottom line. The biggest industry concern is how to make use of this technology in a constantly changing competitive business environment. One way is to integrate the RFID technology and data to the existing enterprise systems seamlessly. This will ultimately lead to business process improvement which can be translated into business process transparency and flexibility. In this chapter, we will focus on the software company SAP AG's long-term RFID strategy to meet the emerging needs of its enterprise customers. Also, we will study, how the RFID strategy makes SAP eco-system more dependable for businesses. There are already more than 35,000 customers who use core SAP as their enterprise system. Therefore, SAP's Auto-ID strategy is critical to many businesses around the world.

SAP started its RFID corporate research in 1998 and began the development its RFID solution in year 2001. In the year 2004, SAP delivered its RFID solution called 'SAP Auto-ID Infrastructure' or simply called as 'AII'. The recent release of AII is version 2.1. So far 19 customers have signed up for this solution and two customers SC Johnson and Pacific Cycle have gone live with AII successfully. While AII was getting developed, SAP geared all its technology solutions towards Enterprise Services Architecture, an offshoot of Service Oriented Architecture (SOA) concept. SAP offers complete business process level integration to AII with other SAP solutions using its Netweaver platform.

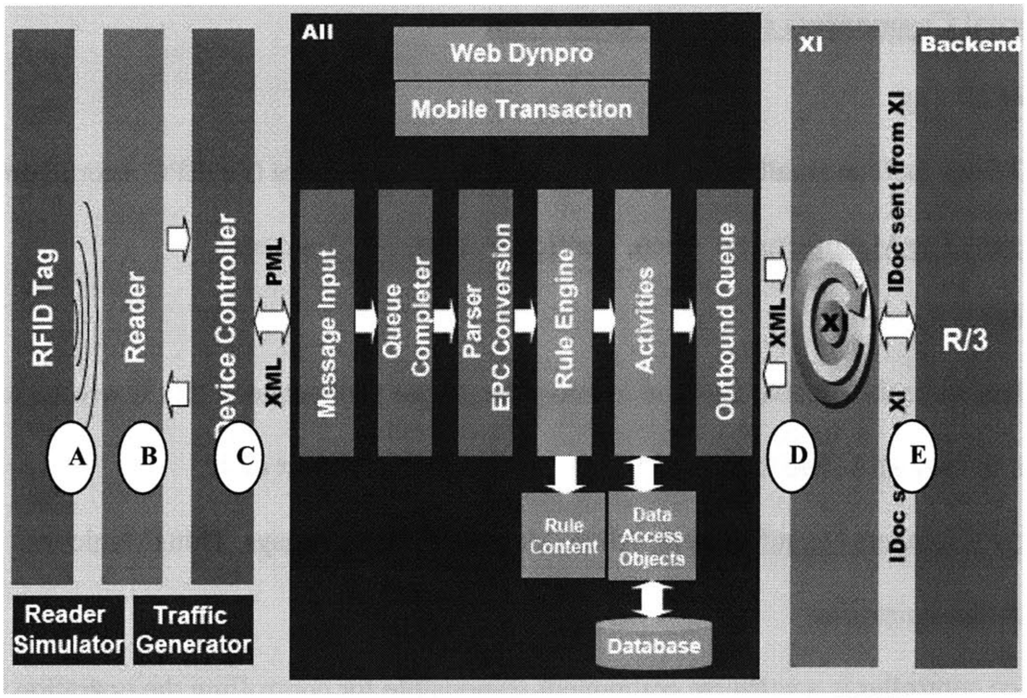
9.2 SAP RFID solution – architectural requirements

SAP predicts that RFID will not only replace some existing software, hardware or technologies but also drive complex business processes. Therefore, SAP is developing its Auto-ID Infrastructure (AII) solution as a way to accomplish RFID enabled business processes. Given that SAP has more than 35,000 customers worldwide, compliance to different customer requirements is going to be difficult. It will necessitate SAP AII to be a highly configurable solution.

The next major challenge is going to be managing large amount of RFID data that might come out of different RF devices. This mandates AII solution to be highly reliable and scalable. Also, AII solution should be able to filter and analyze large amount of data and provide only the value added data to underlying applications. Common internet protocols could be used to satisfy the data communication requirements. Also, AII should be compatible with other SAP systems. This may require SAP to develop its AII solution using its ABAP programming language. SAP AII needs to be built on the SAP's own Web Application Server (WAS) technology. Use of the above mentioned technologies will also help SAP to comply with its own Netweaver Integration platform. Therefore, SAP not only looks at its huge customer base but also looks at the system components of its own eco-system for requirements. This is a major challenge but an Enterprise Software company like SAP AG has the ways and means to develop and deliver such a solution. On the RFID end, the tag and reader technologies are constantly changing as no industry standard has emerged so far. As a result, SAP cannot build on hardware compliance and integration at this time. So, it makes sense for SAP to get a RFID integration middleware component from some established hardware vendors.

9.3 Technical architecture (SAP AII)

Recently, SAP has delivered an AII solution. As per [17], the following diagram illustrates its high level technical architecture. This picture is nowhere close to complete. Substantial improvements are expected in the next few years.



© SAP AG 2005, SAP TechEd '05 / SPI 207 / 19

Figure 9-1: Technical Architecture of SAP AII

<u>Critical components [17] of AII</u>

1. Web Dynpro

SAP's Web Dynpro offers a design-time environment that allows its customers to model and create browser-based user interfaces. It also provides a standards-based runtime environment that is device independent and bridges the gap between different platforms like J2EE, ABAP, and .NET, and different Web browsers.

2. *Rule engine & Activities*

XML/PML is translated into appropriate ABAP (SAP's Programming Language) context. Rules are assigned to business processes and activities are attached to the rules. The *Rule: Activity* can be defined as *1:n*. This means one activity can be used by many rules. This part of AII contains the application logic which reads and writes data from and to the database.

External Components of the AII eco-system

A. RFID Tag

RFID Tags include small chips that carry unique product codes (i.e. EPC information)

Different Tag Manufactures: Alien, Matrics, Philips, ST Micro etc.

B. RFID Reader

Readers wake the passive tags and identify the unique product code that is written on them. In case of Active tags, readers just recognize the product code.

Different Readers Manufacturers: Alien, Matrics, Philips, Tagsys, ThingMagic etc.

C. Device controller

Device controller is a software component responsible for controlling the operation of specific physical RFID readers. It enables operating and integrating of single RFID device groups (logically grouped devices) in a location. This component includes proprietary software code of RFID Middleware providers.

Device controller Software: Sun, Oat Systems, Microsoft, Intel, ConnecTerra etc.

D. SAP Exchange Infrastructure

A powerful integration broker [13] that works well with various connectivity, format, and protocol requirements of an IT landscape. It is a component of the SAP NetWeaver platform and runs on the SAP Web Application Server (SAP Web AS).

E. SAP R/3 System

This is SAP's traditional core Enterprise Resource Planning system.

Network Protocols

i. PML

Physical Mark-Up Language (PML) is used as a common language in the EPC global Network to define data on physical objects.

ii. XML

XML (Extensible Markup Language) allows information and services to be encoded with meaningful structure and semantics

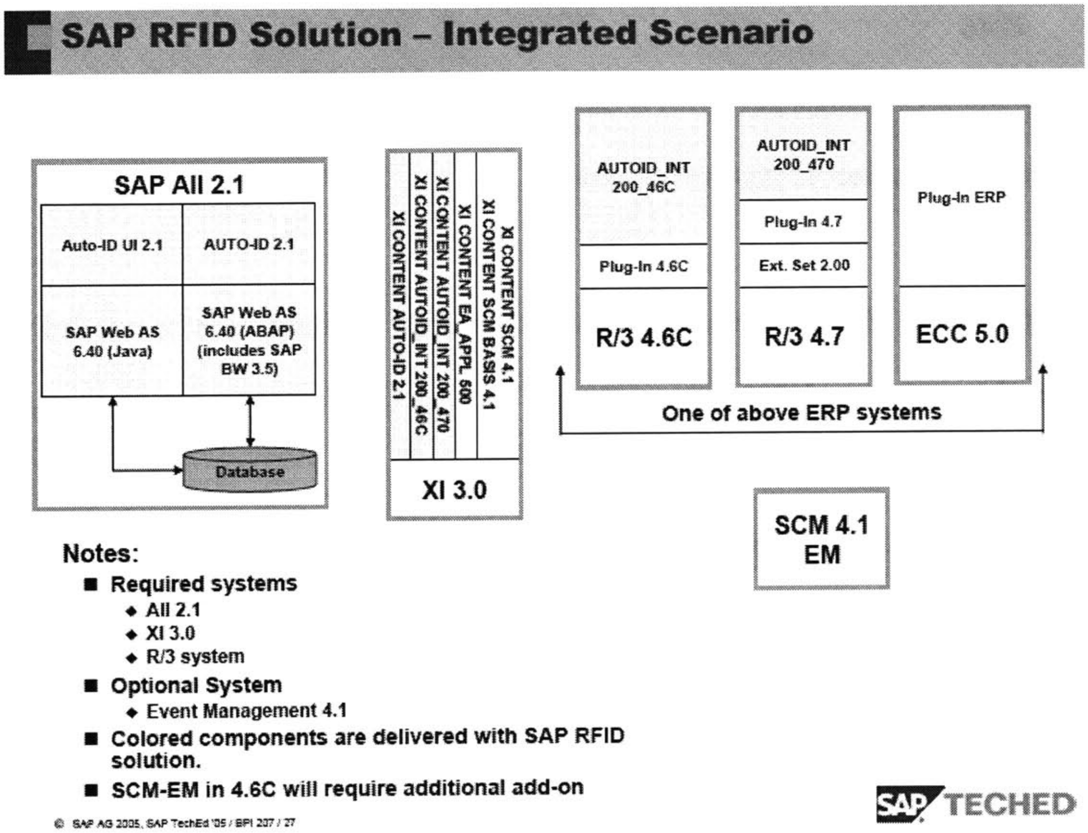
iii. IDoc

Intermediate Document (IDoc) is a data container for data exchange between SAP systems or between an SAP system and an external system.

AII Operation

An RFID reader reads a tag and sends the information to the device controller. Device controller monitors the performance of the entire hardware environment from a central console location and has interfaces to all hardware devices (readers, printers and others). Device controller software integrates devices to the application software by passing XML/PML data streams. The data streams go through different layers as shown in the architecture diagram. After the AII data processing, the data is sent in XML format to SAP's Exchange Infrastructure (XI). XI manages and passes the RFID data to different SAP systems in an IDoc format. The backend systems are capable of receiving information from XI through special *Plug-Ins*. Then, the data is processed by the application logic of the corresponding R/3 system.

9.4 SAP RFID Solution – Integrated scenario



SAP RFID Solution – Integrated Scenario

SAP AII 2.1

Auto-ID UI 2.1

AUTO-ID 2.1

SAP Web AS 6.40 (Java)

SAP Web AS 6.40 (ABAP) (includes SAP BW 3.5)

Database

XI CONTENT SCM 4.1

XI CONTENT SCM BASIS 4.1

XI CONTENT EA_APPL 600

XI CONTENT AUTOID_INT 200_470

XI CONTENT AUTOID_INT 200_46C

XI CONTENT AUTO-ID 2.1

XI 3.0

AUTOID_INT 200_46C

Plug-In 4.6C

R/3 4.6C

AUTOID_INT 200_470

Plug-In 4.7

Ext. Set 2.00

R/3 4.7

Plug-In ERP

ECC 5.0

One of above ERP systems

SCM 4.1 EM

Notes:

- Required systems
  - AII 2.1
  - XI 3.0
  - R/3 system
- Optional System
  - Event Management 4.1
- Colored components are delivered with SAP RFID solution.
- SCM-EM in 4.6C will require additional add-on

© SAP AG 2005, SAP TechEd '05 / BPI 207 / 27

SAP TECHED

Figure 9-2: SAP RFID –Integrated scenario

SAP AII 2.1

The above diagram [17] depicts an SAP/RFID integration scenario for an enterprise. The AII 2.1 (the latest available version) includes two logical support environments. The Java based SAP Web As provides services to Java based application components. The ABAP based SAP Web As provides services to ABAP (traditional SAP development environment) based application components. The power of common SAP Web AS goes beyond just supporting Java and ABAP. Java and ABAP can share a basic SAP work process (worker thread) and work seamlessly. Therefore, a SAP logical work unit (LUW) could consist of both Java and ABAP requests. The common SAP work process then connects to the database for data inert, change/update and delete. Thus, all these components make the complete and robust SAP AII system.

SAP Exchange Infrastructure (XI 3.0)

This information broker sits in between SAP AII and an organization's SAP eco-system. This infrastructure includes plug-ins for different supported SAP systems. The diagram above shows multiple versions of the same SAP systems that are available for information exchange. Therefore, this middle layer ensures integration of different system components across an enterprise landscape.

Supported SAP R/3 systems

In an organization, it is highly likely that multiple versions core SAP software are being run to support different business areas and business processes. These systems are customized significantly to meet the business needs. This type of business configuration constantly goes through changes as the business need changes. These configurations are applied to the application logics of the underlying SAP systems. These SAP systems are capable of providing services to their independent application logics and the database. However, one of the application logics supports the integration of external systems. This logic relies on the XI related plug-ins to accomplish efficient and seamless data transfer between systems.

By using these different system components, an ordinary business process becomes an RF-enabled business process thereby creating real business value. Also creating composite applications with the use of SAP Netweaver platform is possible with this setup. The next chapter details the integration of AII with SAP Netweaver.

9.5 SAP NetWeaver platform & Auto-ID

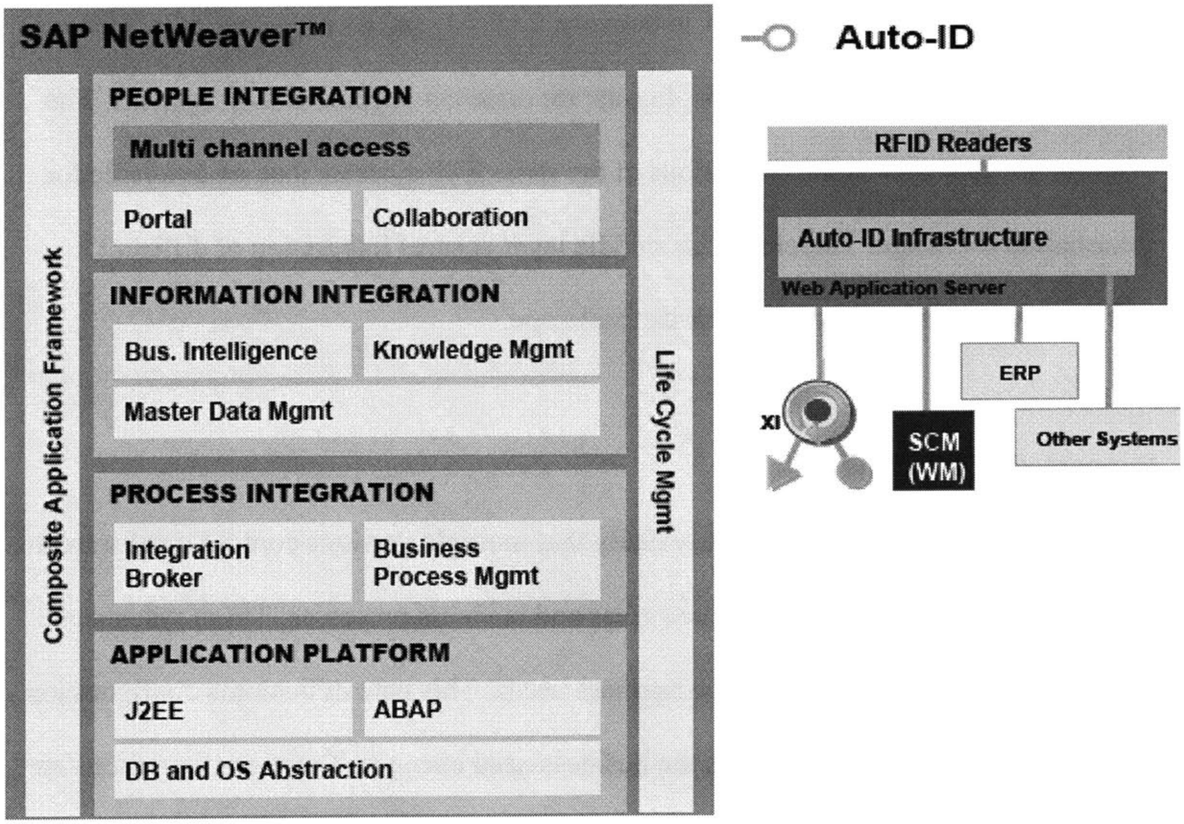


Figure 9-3: SAP RFID & NetWeaver

SAP Enterprise Services Architecture (ESA) is a business-driven approach to Service-Oriented Architecture (SOA) [21]. SAP NetWeaver product provides tools to implement the ESA concept. SAP NetWeaver also provides the Enterprise Services Repository so that appropriate services can be published to create system independent or composite applications. As we have seen before, AII connects to the SAP Eco-system through Exchange Infrastructure (XI), a NetWeaver component. With the combined power of AII and NetWeaver, RF-enabled business processes become reality. Potentially, RFID related services could be published in Enterprise Repository that will make 'never possible' complex business processes possible. Also, it is important to note that all of the technology components described here are written and developed using SAP's native language.

9.6 Future State: AII with NetWeaver

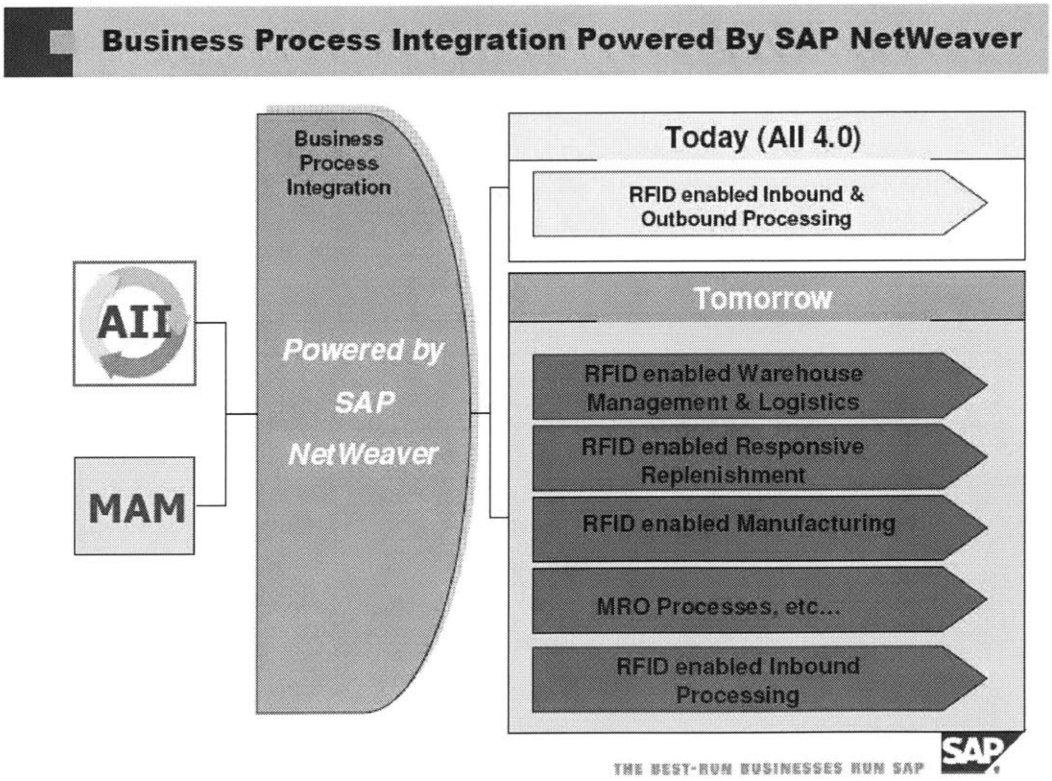
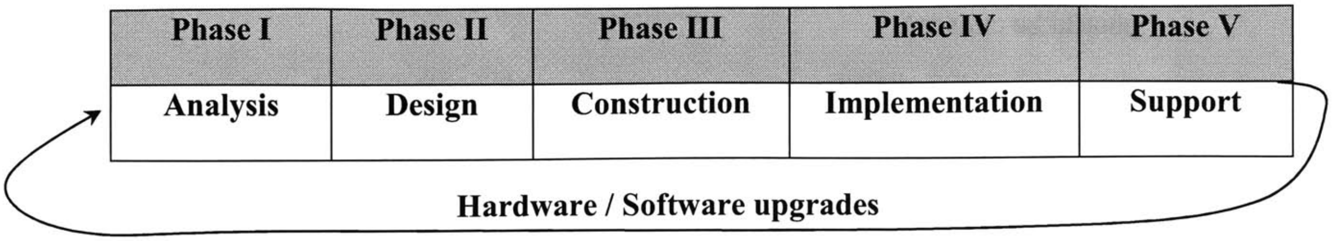

Business Process Integration Powered By SAP NetWeaver
AII
MAM
Business Process Integration
Powered by SAP NetWeaver
Today (AII 4.0)
RFID enabled Inbound & Outbound Processing
Tomorrow
RFID enabled Warehouse Management & Logistics
RFID enabled Responsive Replenishment
RFID enabled Manufacturing
MRO Processes, etc...
RFID enabled Inbound Processing
THE BEST-RUN BUSINESSES RUN SAP
SAP

Figure 9-4: Future: SAP RFID & NetWeaver

In the future, from AII 4.0 release [17], SAP is aiming to power AII with its sophisticated NetWeaver platform. Combining AII with NetWeaver adds agility to the responsiveness of business processes which will immediately translate in to real business value. This integration will allow AII to make use of other Service Oriented Architecture (SOA) features of NetWeaver. Actually, some of the AII services can be published in to services repository of NetWeaver. This will enable external systems to access the SAP Auto-ID information seamlessly. Ad-hoc workflows could be built using Netweaver platform and they could integrate people, processes and tools across different enterprises.

9.7 AII Implementation methodology

The author has developed an AII Implementation methodology from his years of experience with SAP system implementations.



Phase I	Phase II	Phase III	Phase IV	Phase V
Analysis	Design	Construction	Implementation	Support

Phase I. Analysis

- Requirements gathering
    - Internal (Departments, warehouses, facilities etc.) This is internal to an enterprise.
    - External (Transporters, Distributors, Customers, Partners etc.) This is external to an enterprise.
- RFID hardware selection (This process involves evaluating different hardware vendors)
    - Selection of tags (Tag price, tag performance, and tag vendor matters for long-term planning)
    - Selection of tag testing equipments (Use this equipment to test the tag every step of the way)
    - Selection of readers (Reader performance, compatibility with tags, adherence to standards)
- Software selection

  - Selection of middleware software (SAP certified RFID middleware vendor is recommended)
  - Selection of AII device controller (SAP certified AII device controller vendor should be chosen)
- SAP AII
  - Hardware selection (Identifying SAP certified hardware vendor is important for long-term system support)
    - Sizing (Make use of SAP AII sizing document for multiple hardwares)
  - Software selection (Different AII versions available. E.g. 2.0 and 2.1; Different SAP application servers are available aswell)
    - AII version analysis & selection (This is critical for reducing future maintenance and support costs)
  - Operating system selection (This should be based on the IT environment of the enterprise)
  - Database selection (This should be based on the IT environment of the enterprise)
  - SAP Exchange Infrastructure (XI) version selection (This should be based on the SAP environment of the enterprise)
- SAP AII landscape preparation
  - Documentation of different system components with technical specifications
- SAP Eco-System Landscape preparation
  - Documentation of different SAP systems with technical specifications
- Finalize complete system specification

- Identify business processes (BP) to be RF enabled (An ordinary BP becomes an RF-Enabled BP)
- Perform GAP Analysis (Identify and document the process gaps between existing BP and RF-BP)
- Documentation

Phase II. Design

- Prototyping
    - Small scale system demo of SAP AII and non-SAP AII. i.e. RFID systems
- Build test systems
    - Develop and Install SAP-AII and RFID test systems
- Translate requirements
    - Converting requirements into functional and system forms
- Create RF enabled business process flow
    - Develop and organize the new RF enabled BPs
- Design RF enabled business processes
- Create RF enabled business process review (BPRs) documents
- Document the Design

Phase III. Construction

- Develop and Install system components
- Develop and Install subsystems

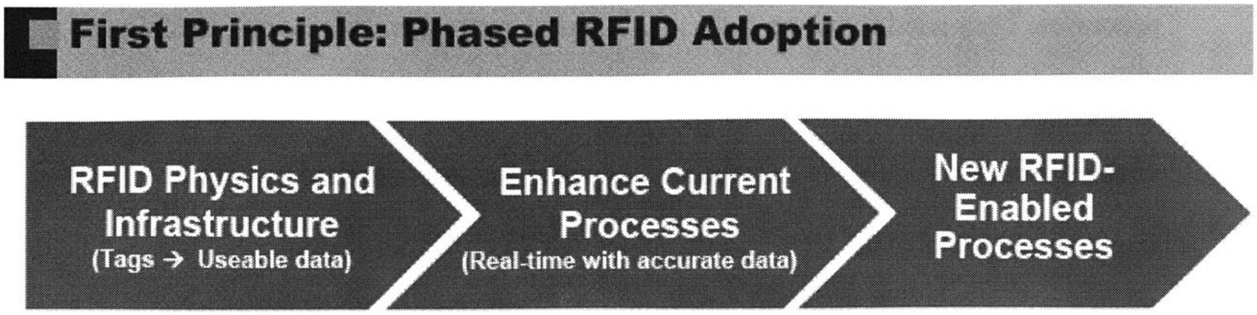
- Prepare test scripts (To facilitate the acceptance testing process across the enterprise)
- Perform unit testing
  - Perform testing on individual system components to flush out glitches
- System Integration Testing
  - Perform testing on the complete system to flush out glitches
- Perform stress testing (To check the system load handling or System throughput)
- Design performance metrics (To continually improve system performance)
- Documentation

Phase IV. Implementation

- Prepare training manuals
- End-user training (Train-the-trainer approach is recommended)
- System Go-live
- System performance tuning (To continually improve system performance)

Phase V. Support

- Execute SAP-AII technical support
- Execute RF enabled business process support
- Disseminate enterprise-wide communication



First Principle: Phased RFID Adoption

RFID Physics and Infrastructure
(Tags → Useable data)

Enhance Current Processes
(Real-time with accurate data)

New RFID-Enabled Processes

RFID Physics and Infrastructure

SAP recommends [17] its customers to adopt a phased approach while implementing RFID or Auto-ID infrastructure. This is critical because RFID is an evolving technology. The hardware/software standards are slowly emerging. There are many compatibility issues between RFID tags and readers. However, SAP handles these incompatibility issues through the additional architectural layer called 'Device Manager' [17]. This device manager component is also slowly being designed and implemented. This component will be integrated into AII 4.0 as a core infrastructure.

Enhancing current processes

Current processes will receive more up-to-date information with the help of AII. For example, shipment level product tracking will expand to pallet level tracking and item level (SKU level) tracking. With the advent of new data, the current processes may create gaps within themselves. These gaps need to be identified and fixed. Then the next step is going to be seamlessly integrating RFID data into the business processes.

RFID Enabled Processes

When processes get integrated with RFID, they truly become powerful RFID enabled business processes. They are very different from traditional stand alone business processes. They need to be created using business process re-engineering techniques. Also, documenting the difference between the traditional business processes and RFID enabled business process is an important key step for any organization that is implementing SAP AII.

9.8 Business value proposition

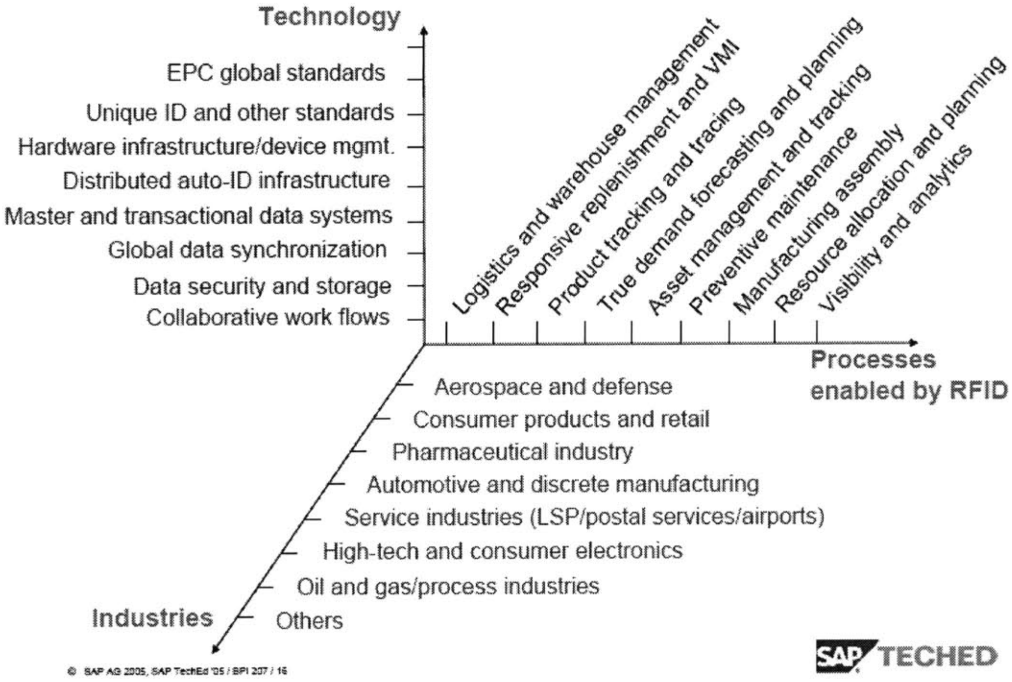

Technology
EPC global standards
Unique ID and other standards
Hardware infrastructure/device mgmt.
Distributed auto-ID infrastructure
Master and transactional data systems
Global data synchronization
Data security and storage
Collaborative work flows
Logistics and warehouse management
Responsive replenishment and VMI
Product tracking and tracing
True demand forecasting and planning
Asset management and tracking
Preventive maintenance
Manufacturing assembly
Resource allocation and planning
Visibility and analytics
Processes enabled by RFID
Aerospace and defense
Consumer products and retail
Pharmaceutical industry
Automotive and discrete manufacturing
Service industries (LSP/postal services/airports)
High-tech and consumer electronics
Oil and gas/process industries
Others
Industries
© SAP AG 2005, SAP TechEd '05 / BPI 207 / 16
SAP TECHED

Figure 9-5: SAP RFID: Business value proposition

This picture [17] depicts Technology, RFID enabled processes and respective industries in three different perpendicular axes. However, when we look at the enabled processes and industries the worth of underlying business value emerges.

- SAP customers could bring down the Total Cost of Ownership (TCO) by using SAP AII. This is because most of the integration issues will be taken care by SAP.

- While RFID technology keeps evolving, a software company like SAP should be able to cope up with changes and deliver patches and value added software solutions to its customers consistently.
- SAP is a well known Enterprise Resource Planning system. By integrating the RFID data, SAP's standard business processes become even more valuable. This will help to improve the efficiency of the whole enterprise system and the enterprise that it serves.

9.9 AII enhances dependability of SAP eco-system

From the elaborate discussion in this chapter, it is evident that organizations that use SAP eco-system have many benefits to reap from implementing SAP AII. AII Implementation is going to be complex; however, the use of RIFD data with organization's business processes will immediately help improve the business' bottom line.

SAP AII is evolving and it will be integrated later into the SAP NetWeaver stack. Since AII is built on SAP's standard technology platform WAS, stability of the NetWeaver stack won't be threatened. While responding to the emerging business needs, AII along with standard platform, protocols and technology will improve the 'dependability' of the SAP eco-system architecture.

10. Case Study: On-Demand Customer Relationship Management (CRM)

10.1 Architectural requirements

As per Gartner, major businesses worldwide will boost marketing and advertising in the next few years. Following are some of the reasons.

- Understanding of the strategic value that marketing brings to the rest of the company.
- To leverage technology and process automation to empower Marketing and Sales to cultivate growth opportunities.
- IT departments are slow to react to the growth agendas of Sales and Marketing departments.
- Mid and small size companies can't afford to implement an enterprise CRM system and maintain competitive edge in the marketplace at the same time.

Therefore, this case study primarily analyzes three solutions that are developed to meet these architectural requirements. (Salesforce.com, SAP on-Demand CRM and Oracle-Siebel CRM OnDemand)

10.2 On-Demand CRM solutions

10.2.1 Salesforce.com

As per [31], AppExchange OS, salesforce.com's new on-demand operating system, allows companies to manage and share CRM and custom business applications in a single environment — with one data model, one sharing model, and one user interface. This software model may be easy to implement and use. This simplified architecture is very suitable and manageable for small and mid sized businesses. With salesforce.com, when the system or user base grows, there is no concept of moving to a traditional software environment. The CRM environment will still operate from the hosted environment.

10.2.2 SAP on-Demand CRM

SAP recently offered SAP on-Demand CRM to its customers. This on-demand CRM uses the same architecture, underlying technology and data model of its standard mySAP CRM solution. This on-demand version is available from SAP for a price. When the organizations grow SAP will offer smooth transition from On-demand version to mySAP CRM. Analysis shows that on-demand CRM from SAP is not different from mySAP CRM but only offers only hosting operational advantages. Still, this may not be a perfect solution for small and mid-sized businesses because this solution by SAP's own definition will bring all the complexities of mySAP CRM. So, small and midsized companies cannot implement this complex solution and maintain competitive edge in the marketplace at the same time.

10.2.3 Oracle-Siebel CRM OnDemand

As per [32], Siebel CRM OnDemand is available as hosted and on-premise software version. It also offers a web services toolkit that enables customers to meet their unique integration needs with legacy and third-party applications using web services standards. This is very similar to SAP on-demand CRM model.

10.3 On-Demand CRM Conclusions

While comparing these three solutions, salesforce.com clearly stands out as unique. Its architecture and data model are much simpler than that of two other solutions. Based on the simplicity, salesforce.com is highly recommended for small and mid sized businesses. Organizations with existing mySAP CRM and Siebel CRM implementations can go for the respective on-demand hosted CRM solutions if that model reduces the TCO.

11. Conclusion

This thesis established a definition for Dependable System Architecture and criteria to evaluate such architecture. SAP, a leading ERP system, was selected for the evaluation. The evaluation criteria employed system frameworks, concepts, tools, components and methodologies. During the process of evaluation, the thesis touched upon many different architectural evaluations such as technical, operational, implementation and service and support.

In the process of identifying dependable system architecture, it is critical to analyze the strengths and weaknesses of architecture. So, this thesis critiqued the strengths and weaknesses of SAP architecture and its eco-system. The critique has lead to a conclusion that high level of diligence is required to run the SAP system.

While offering some important benefits such as 'application integration' and 'standard platform', SAP also comes with lot of complexity. So, it is clear that a dedicated SAP support team is required to reap the real business benefits out of SAP. The support team with the help of SAP Program Management should proactively keep the software of the SAP eco-system up to date and 'software change' in control.

Analysis of past and present trends in SAP technology shows SAP's technology leadership in the enterprise software industry. SAP's solution offerings are well aligned with the goals and objectives of SAP Company. SAP's solutions are also focused on meeting the goals and objectives of its more than 35,000 customers. This indicates the business value proposition of SAP to the business world.

After multiple analysis and evaluations, this thesis concludes that the SAP eco-system does provide Dependable System Architecture for businesses. Furthermore, based on the very definition, Dependable System Architecture emerges in order to meet the goals and objectives of the stakeholders within SAP's surrounding business context.

Strengths and weaknesses of SAP and learning from the case studies reveal that SAP's technologies are developed to meet the emerging needs of businesses. However, some of SAP's technologies are still evolving. Especially, SAP's ESA may not bring immediate ROI to SAP AG but will cost SAP AG lot of resources. SAP's RFID solution is not nearly complete and its on-demand CRM solution model may not succeed with its customer base. Therefore, SAP has lot of obstacles and challenges in the years ahead.

To meet the challenges and respond to the competition from Oracle, this thesis recommends SAP to adapt a clear strategy to become a platform leader. Becoming a platform leader is one of the ways to maintain competitive advantage in the enterprise software industry.

12. Bibliography

1. David Garlan and Mary Shaw. An Introduction to Software Architecture. School of Computer Science, Carnegie Mellon University. January 1994.
2. Michael Jackson. Problem Structure and Dependable Architecture. Available at http://mcs.open.ac.uk/mj665/.
3. Professor Daniel Jackson. ESD.34: Lecture: Software Architecture. MIT, 2005.
4. Professor Olivier L. de Weck. ESD.34: Lecture: Product Platform Extent and Flexibility. MIT, 2005.
5. Dr. Ernst Fricke. ESD.34: Lecture: Flexibility and Robustness – Major Characteristics for Changeable System Architectures. MIT, 2005.
6. Victor Tang, ESD.34: Lecture: Nagano Winter Olympics 1998: IT Architecture Case Study. MIT, 2005.
7. Wermelinger, M., Koutsoukos, G., Lourenco, H., Avillez, R., Gouveia, J., Andrade, L., Fiadeiro,J.L. Enhancing Dependability Through Flexible Adaptation to Changing Requirements. IEEE Computer Society Press, 2003.
8. John C. Knight and Elisabeth A. Strunk. Achieving Critical System Survivability through Software Architectures. Department of Computer Science, University of Virginia. 2004.
9. U.S. Department of Commerce, National Telecommunications and Information Administration, Institute for Telecommunications Services, Federal Std. 1037C.
10. Knight, J. C., E. A. Strunk and K. J. Sullivan. "Towards a Rigorous Definition of Information System Survivability". IEEE, April 2003.

11. Porcarelli, S., Castaldi, M., Giandomenico, F., Bondavalli, A., and Inverardi, P. A Framework for Reconfiguration-based Fault-Tolerance in Distributed Systems. Springer-Verlag Berlin Heidelberg, 2004.

12. Report: January 2006 SAP FACT SHEET. SAP Investor Relations, 2006. Available at: http://www.sap.com/investor.

13. SAP corporate website. Available at http://www.sap.com/

14. SAP Online Help. Available at http://help.sap.com

15. Master Guide - SAP NetWeaver'04, Version 1.54, June 2005, SAP AG.

16. Master Guide - SAP NetWeaver 2004s, Version 1.02, December 2005, SAP AG.

17. Fundamental Aspects of SAP RFID Technology. Session BPI207. SAP TECHED '05, Boston, 2005.

18. Enterprise Services Architecture. SAP TECHED '05, Boston, 2005.

19. Randall S. Clemons, Mark K. McBeth. Public policy praxis theory and pragmatism: A case approach. Prentice Hall, 2000.

20. Professor Edward Crawley. Course: ESD.34 System Architecture. MIT, 2005.

21. Enterprise Services Architecture and SAP NetWeaver. Session ESA104. SAP TECHED '05, Boston, 2005.

22. Basics of .Net. 2005. Available at: http://www.microsoft.com/net/basics.mspx

23. IBM Service Oriented Architecture. 2005. Available at: http://www.ibm.com/software/solutions/soa/

24. Gartner, Inc. Magic Quadrant for Web Services Platforms, July 2005.

25. SAP Solution Manager. 2006. Available at: http://service.sap.com/solutionmanager

26. Master Guide - SAP Solution Manager 4.0, Version: Ramp-Up, 2005, SAP AG.

27. Foundations of IT Service Management based on ITIL by itSMF. Van Haren Publishing, 2005.

28. SAP Solution Manager - ITIL Support. Product Management SAP Solution Manager. SAP AG, 2005.

29. Office of Government Commerce - IT Infrastructure Library (ITIL). Available at: http://www.itil.co.uk/

30. Annabelle Gawer, Michael A. Cusumano. Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation. Harvard Business School Press. 2002.

31. Salesforce.com corporate website. Available at http://www.salesforce.com/

32. Siebel OnDemand solution website. Available at http://www.crmondemand.com/