Managing Agile Information Technology Infrastructure

By:
Biju Kalissery

Submitted to the System Design and Management Program in Partial Fulfillment of the Requirements for the Degrees of

Master of Science in Engineering and Management

at the

Massachusetts Institute of Technology

June 2007

Author: .....
Biju Kalissery
System Design and Management Program

Certified by: .....
Thesis Supervisor: Prof. Michael Cusumano
Sloan Management Review Distinguished Professor, MIT Sloan School of Management

Certified by: .....
Pat Hale
Director, System Design and Management

Table of Contents

List of Figures

List of Tables

Abstract

Information technology (IT) can be a key contributor for the successful implementation of business strategies. However, companies normally find it hard to synchronize their evolving business strategies with the capabilities in IT. This thesis analyzes the key contributors to the problems in synchronizing business strategy and Information Technology and suggests both management and technical frameworks for an agile IT infrastructure that can stay in sync with the evolving business strategy. Agility in IT infrastructure means the ability for the infrastructure to accommodate evolving needs and business strategies without significant re-architecture. There is no magic bullet that could induce agility into an IT infrastructure and its management. But, this thesis studies the best practices in management and technology which are being used by industry leaders successfully.

Thesis Supervisor: Michael Cusumano

Title: Professor, MIT Sloan School of Management

Acknowledgements

I would like to thank Professor Michael Cusumano for being a patient advisor while I was trying to juggle the academic responsibilities along with the challenges of a new job in a startup company. Professor Cusumano's class on Software Business had opened my eyes into the issues of managing technology and aligning the technical decisions with business strategy. Thanks to Professor Starling Hunter for introducing me to the concepts of Strategic Management and motivating me to learn more about it. Thanks to Pat Hale, Director of SDM, and the extremely helpful staff at the SDM office. Pat has been an eternally patient listener and constantly worked with students to give us the maximum benefits of MIT academic life while trying to accommodate the special needs of our professional careers. I would also like to thank my colleagues at Marcam and IdentiCert for all the intellectually stimulating conversations and guidance throughout my career. Above all, I would like to thank my parents, sisters and their families for the all support they provided for my pursuits in personal, professional and academic goals.

1 Introduction

Today's firms are constantly under competitive pressures from globalization, deregulation, advances and convergence of industries and technologies. The ability of a firm to respond quickly depends on the existing infrastructure and its ability to adapt to new challenges. Many business leaders find that their Information Technology (IT) infrastructure is not able to quickly adapt to the new changes [4] despite the fact that their investments towards IT is a significant part of the capital expenditure.

Information Technology (IT) is pervasive in today's organizations to enable business electronically and to connect with customers, suppliers and other strategic partners. For the discussion in this thesis, firm's total investment in hardware, software, telecommunications, electronic data and people dedicated to providing the information services are considered part of the IT infrastructure. The foundation for a firm's IT infrastructure is on the public IT infrastructure (examples: internet, telecommunication service providers). A firm also requires its own internal infrastructure (examples: common email servers, large database servers, public web sites, application servers for processing information). Within a firm, each business unit may build its on business applications (examples: customer order processing, purchasing, manufacturing management, business intelligence) on top of firm's common infrastructure. Market leaders in the industry create a unique combination of the IT infrastructure and applications to enable their strategic business goals.

Agility in the IT infrastructure is one of the key ingredients to a firm's ability to respond to the changing environment. The purpose of this thesis is to analyze the reasons for why it is hard to build an IT infrastructure that is agile enough to align and keep up with the evolving business strategy and to recommend a solution including frameworks for management and technical architecture.

2 The purpose of IT

In many firms, Information Technology is integral to achieving the business goals and can be the single largest capital expense. The purpose of investments in IT is to successfully implement current strategies and to use the technologies to enable new strategies (strategic intent). The management objectives for IT can be categorized into four groups: infrastructure, transactional, informational and strategic [2].

Infrastructure systems form the foundation for the IT capability in the firm. It is normally shared across business units to reduce the marginal cost of business unit's IT. For example: network services and centralized email servers are infrastructure services that are normally facilitated by the central IT departments. There is uniformity in the requirements for the common infrastructure components, no matter which business unit application is built on top of it.

Transactional IT systems are built on top of the common infrastructure. Transactional systems execute the basic, repetitive business processes efficiently. For example, systems that support inventory control, customer order processing and accounts receivables. Transactional systems are primarily used to reduce the cost of running the

business by executing them as efficiently as possible. The purpose of transactional applications is to support the business strategy. They are not part of the core strategy and usually tend to be stable compared to the strategic applications.

The informational IT systems provide the insights for managing and controlling the firm. They support the planning and decision making processes. These systems collect and summarize data from many transactional systems and present high-level views to senior management. Executive Information Systems (EIS) or Dashboards are examples of Informational IT systems.

Strategic IT systems are built for gaining competitive advantage in the market and thereby to increase the market share or sales. For example, Citibank started using Automatic Teller Machines (ATMs) in the 1980s successfully to increase their market share from 4 percent to 13 percent [2]. Strategic IT systems may evolve into transactional systems in the long term, once their strategic value erodes in the market. In case of Citibank ATMs, once competitors caught up to the new technology, ATMs became transactional IT systems primarily used for driving costs down.

3 Problems in aligning IT and business strategy

Strategic implementations of IT can provide significant competitive advantage to enterprises. However, IT departments within enterprises have to constantly refine the internal systems to meet the evolving needs of the organization. The challenges for IT to be in sync with strategy can be because of business environment changes, progress in technology or other organizational issues [4].

3.1 Business environment changes

Dynamic collaboration between partners in the value network is increasing, creating more need for information integrations between enterprises. For example, the integration between Wal-Mart and Procter & Gamble resulted in more efficient sharing of data, reduced need for inventories and increased sales [17]. Globalization and increased trade between firms across national boundaries have amplified the importance of electronic communications and data exchange. Firms now have to strategize for competition from across the border and even from firms on the other side of the globe. Changing business environments and competitive demands shifts the underlying strategic assumptions of the firm. These changes in strategic assumptions have significant implications for IT.

3.2 Progress in technology

The fast evolution of the hardware and software industries have created ever more powerful computer hardware and higher and higher levels of software abstractions, making it possible to build complex systems with relative ease. With the new service oriented architecture and XML/SOAP standards based interfaces, business partners can seamlessly integrate their disparate information systems. With the advent of Web services and application mashups, users expect simple, easy to use, consolidated user interfaces, even if the systems behind the scenes are complex and hard to integrate. Continuous evolution of the software and hardware systems can create the need for IT departments to keep up or be left behind in the market. However, changing an established enterprise software system is a non-trivial project. Large enterprise software

systems are normally rigid in their business rules and can prevent IT departments from quickly responding to the business needs. Extensive customizations of ERP systems can also create problems with migrations to future versions of the application.

3.3 Organizational Context

The operational management in the firm may not have a clear understanding and commitment to firm's strategic focus. This may be caused by lack of focus by senior management, frequently changing strategic intents or insufficient understanding of the senior management about IT's need for strategic awareness. It is also possible that the business and technology strategies were defined in isolation and were not synchronized. Technical, political, or financial constraints could also cause the IT systems to include numerous legacy systems that are not flexible enough to keep up with the changes in business strategy.

Leaving the technical strategy decisions to IT staff without senior management involvement can cripple the alignment between business and IT. Without being aware of the business strategy, IT staff will normally be interested in the latest technological innovations. Senior managers have to take an active role in understanding the strategic decisions made by the IT and continuously communicate with the line managers to synchronize business and IT strategies. Another common problem is the wrong tools used for measuring the performance of IT. IT departments are often assessed for their ability to contain costs and not for their ability to respond flexibly to strategy [4]. Thus, the expectations of senior managers in a firm are at odds with those of IT managers.

4 The solution

Business leaders and IT staff are all normally dedicated to the success of the company. However, lack of communication between the senior management and the IT staff often creates the problems in agility. Lack of a common vocabulary between the business leaders and IT staff adds to the problem. As shown by the example of Microsoft realigning itself to plunge into the browser wars, clear and compelling visions from the senior management with immediate tactical ramifications can focus the whole organization on to meeting the goals [5]. The sections below presents some of the frameworks that have found to reduce or eliminate the misalignment between strategy and IT infrastructure.

5 Aligning IT and business strategy

A firm's strategic context combines the long and short term strategies and business goals [2]. The strategic context is made up of the strategic intent (where the firm wants to go in the long term), current strategy (how the business is getting done today) and the business goals (quantitative and qualitative business targets). For example, a car-rental company's strategic intent could be to become the lowest-cost car rental. It's currently strategy could be to provide low rate cars near airports and cheaper hotels. The business goal of this company could be to reduce transaction costs compared to its competitors.

For successful alignment of strategy and IT, firms have to define a companywide strategic intent and business objectives, understand the strategic context of the company, develop business and IT objectives matching the corporate strategic objectives and develop a portfolio of business and IT investments to support the strategic objective [2]. For example, in 1995, Johnson and Johnson, world's largest manufacturer of health care products, established Health Care Systems (HCS) to market its existing products to large customers [6]. This was done in response to the need for an integrated delivery system, where the doctor, hospital and insurance company were are becoming more connected. With the new strategy, J&J needed to form long term relationships with large customers. It required close cooperation and coordination to deliver customer information across multiple J&J companies. This shift in strategy resulted in more focus on shared IT infrastructure investments across the business units.

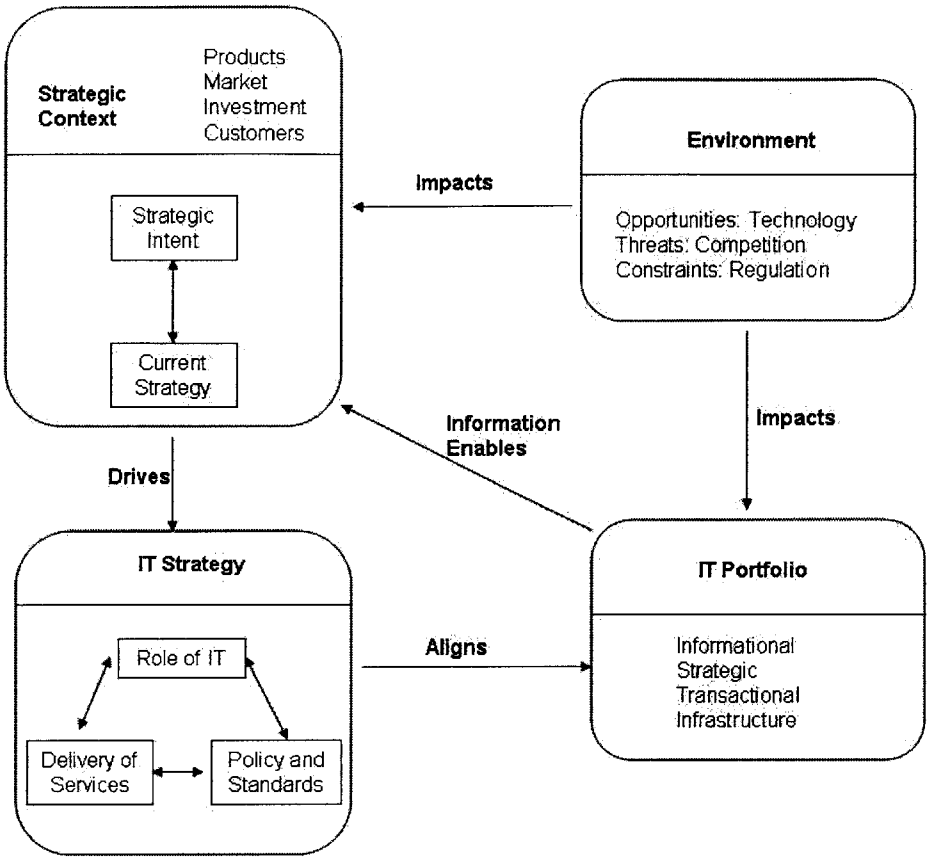

Figure 1 Aligning Business Strategy and Information Technology (source [2])

While choosing IT investments, IT leaders have to consider the following issues to ensure alignment of IT and business strategy [17]:

- Does the infrastructure and application architecture align with the corporate strategy?
- What are the inherent risks and impediments to change in the existing infrastructure?
- What does a viable IT portfolio look like?

5.1 Applications-Infrastructure Scorecard

Flexibility in strategy and IT results from a shared understanding of the agenda by the business and IT managers. Applications-Infrastructure Scorecard is a tool that can be used to facilitate dialog between business and IT [4]. In this paper, C.K. Prahlad and M.S. Krishnan describe how, by dynamically synchronizing the IT capabilities with the demands from strategies, companies like GE Lighting is able to stay ahead in their industry. Their research suggests focusing on the following topics during the discussions between IT and business leaders.

- Application's role in strategy
- Stability of the business process
- Degree of change in the application
- Application sourcing
- Data privacy level
- Level of quality

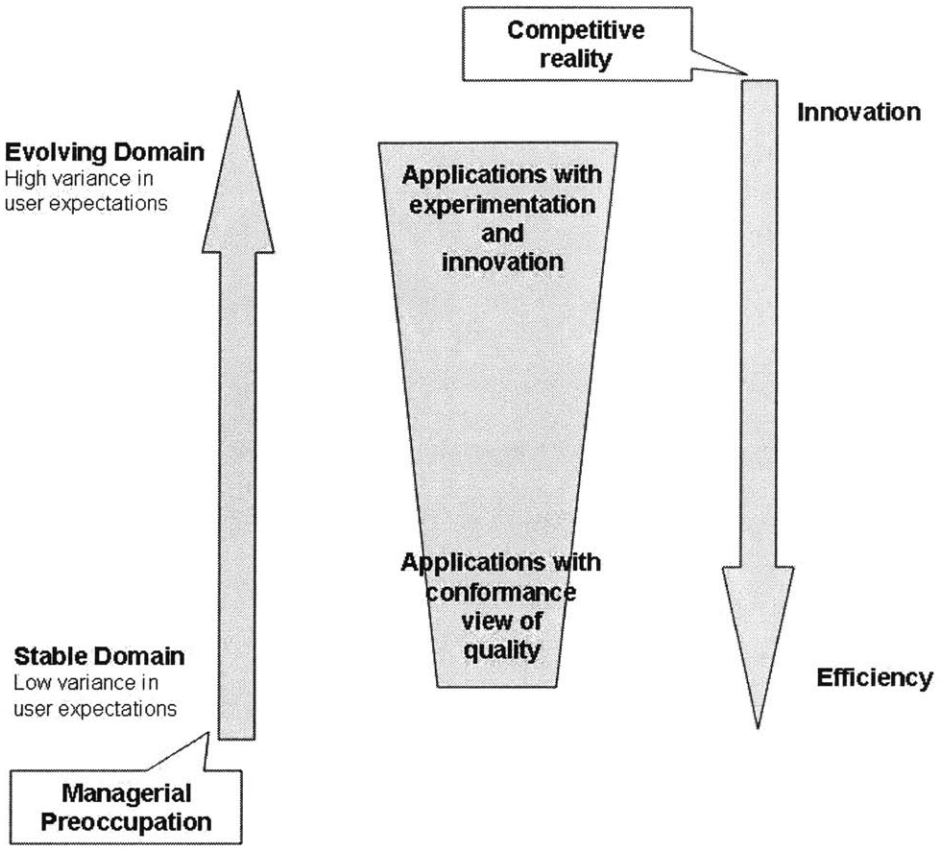

Figure 2 Dynamics of IT application; Efficiency vs. Innovation (Source [4])

The table below shows an example of Application-Infrastructure Scorecard of a fictitious e-commerce company where the customer-facing web site is key to the business strategy. Auditing the existing IT applications and categorizing them using the scorecard would be a good way to start dialog between IT and business leaders. In this simplified format, it would be easier to see each application within the context of the overall business strategy and identify any misalignments. A shared agenda derived from the scorecard can identify the current infrastructure capabilities and potential weaknesses in the system. For example, developing or doing extensive customization to the ERP

application in purchasing would be a bad idea in the organization depicted in the table below because purchasing is not in the strategic focus of the company. Higher number of changes in such applications should alert the senior management of misaligned IT priorities. Similarly, very low rate of change in the strategic applications would indicate stagnated business strategy or IT strategy that may need reviving.

Table 1 Example of Application-Infrastructure Scorecard (adapted from source [4])

IT Application	Role in Strategy	Stability of the business process	Degree of change	Sourcing	Privacy Needs	Level of quality
Finance	Efficiency	Stable	Low	Packaged ERP Application	Low	High
Human Resources	Efficiency	Stable	Low	Packaged Application	High	High
Purchasing	Efficiency	Stable	Low	Packaged ERP Application	Low	High
Customer-facing web application	Innovation	Evolving	High	Custom built	Medium	Moderate

The sections below describe the expected categories for applications for each criteria listed in the application scorecard.

5.1.1 Application's role in strategy

IT applications within the enterprise can be core to the strategy or may be playing supportive roles. The strategic role of an application is between efficiency and innovation. An application that is core to the strategy should support innovation and be flexible. On the other hand, supporting applications should help the company to be more

efficient. For example, for a company manufacturing consumer electronic device, product design and life-cycle management would be part of the core strategy. IT applications supporting the product design and life-cycle management would need maximum flexibility and should support innovation. However, the same company may not need a flexible Human Resources application. Once a suitable pre-packaged HR application is selected and implemented, the company will have a stable system for the foreseeable future.

5.1.2 Stability of the business process

Applications in the IT portfolio consist of stable-domain applications that support efficiency and consistency and evolving-domain applications like support innovation and strategic experimentation. Stable-domain applications normally have lower risk profiles compared to the evolving-domain applications. Businesses processes (example: product design) tied to the evolving strategy of the company tend to be less stable compared to the processes within stable business domains (example: Human Resources or Financial Accounting). In stable domains, there is more clarity on what is expected out of the IT systems. In evolving domains, there is less clarity on what the future direction of the applications would be, because a number of internal and external influences can alter the strategy of the company. The stability of the IT business process domain would depend on the context of the company. For example, Human Resources application is likely to be part of the core strategic application for a consulting services company and may always be in the evolving status.

5.1.3 Degree of change in the application

The number of changes made to an application can be categorized as high, moderate or low. These categorizations can show if the churn in the application is in alignment with the intended strategic focus of the company. For example, do the core strategic applications have high degree of changes? Do the supporting applications have a low degree of change? If not, managers need to analyze the reasons behind changes and fine-tune the IT projects portfolio.

5.1.4 Application Sourcing

Application sourcing decisions can depend on a number of factors; company's strategy, availability of suitable third-party software applications, availability internal and external expertise and available budget. Main sourcing methods are: custom built applications, packaged systems with customizations, or third party service provides. The position of the application in the scorecard can help managers decide where to source the applications from. For example, evolving applications that are part of core strategy should ideally be custom-built in-house. Support applications with stable functions and their customizations can be outsourced to third-parties, especially if security and privacy are not and issue for the application in question. Stable-domain application functions can also be fulfilled by packaged ERP applications, which bring in efficiency and data consistency. More and more companies are also finding that the application service providers with Software as a Service (SaaS) is a good way to source supporting applications by spending low subscription fees (operating expense) instead of large license fees (capital expense).

General Electric Lighting uses the scorecard to gain insights into its IT investments. Their financial system is a stable domain and uses R/2, a version of SAP that is more than 10 years old [4]. GE doesn't see a compelling need for migrating to later version of the application. However, GE invested in building a category management application in-house to meet its evolving strategic needs.

5.1.5 Data Privacy Needs

If the data requires high degree of confidentiality, application cannot be sourced from service providers because it risks the data being transmitted over the internet to be compromised. Proprietary and confidential data is best managed in-house.

5.1.6 Level of quality

IT application's level of conformance to the requirements of the business is another measure that can indicate the alignment between IT and business strategy. For example, stable-domain applications should normally have higher level of quality. Applications in the evolving-domains which are closer to the strategy is expected to have higher number of pending conformance or quality issues.

5.2 IT Portfolio Management

Long IT projects are notoriously hard to control, especially when the managers do not use well-defined metrics to measure the progress and review the projects once they are complete. IT Portfolio Management (ITPM) is a combination of practices and techniques to generate most value out of IT investments and to reduce risk [3]. It is a way to managing IT as a portfolio assets similar to financial portfolio by balancing risk and rewards. The chart below shows the maturity model for ITPM. The stages are named as

Ad Hoc, Defined, Managed and Synchronized. The stages to the right of the chart include the factors to the left of the chart.

Table 2 IT Portfolio Management Maturity Model (source [3])

Factor	Maturity Stage		
	Defined	Managed	Synchronized
Advanced Valuation			Inclusion of qualitative option value in funding decisions: monitoring of project's earned value in deployment
Feedback Mechanism			Feedback on IT alignment with strategy – score cards evaluate each project
Benefits Measurement			Tracking of project benefits after project development is complete; measurement of IT value through the full project life cycle
Active Portfolio Management			Understanding of risk and return portfolio weighed accordingly
Strategic Alignment		Annual review sessions between business-unit heads and IT to discuss IT and strategy alignment	Frequent review sessions with business unit to discuss strategy alignment (quarterly or monthly)
Financial Metrics		Use of financial metrics in prioritizing: NPV, ROI, IRR	
Demand Management		Well-defined scheme for	

		screening, categorizing and prioritizing projects; portfolio-management approach to rank projects for investments	
Centralization	All projects in one database; all IT spending tracked centrally and rolled into one database; centralized project office monitors projects		Use of portfolio software – real-time updates on portfolio modifications, performance and health
Standardization	Applications and infrastructure are well defined and documented	IT portfolio segmented by asset classes – for example, infrastructure, strategic projects.	

5.2.1 Ad-hoc stage

Ad-hoc stage is not listed in the chart. At this stage, IT decisions are uncoordinated. For example, each business unit may start IT projects without coordinating with the central IT department. At this stage, even simple savings by vendor consolidation is hard to achieve. The chance of projects succeeding depends entirely on the team dynamics. None of the lessons learned are reused, and the project is assumed to be a success if the basic objectives were met.

5.2.2 Defined stage

In the Defined stage, companies have identified and documented the main components. Project information is recorded in a central database and monitored. Basic financial

metrics are applied while selecting projects for implementation. However, there are no feedback loops at this stage. At this stage, it is hard to coordinate the IT projects with corporate strategy because there are no common beliefs and standards. Projects are selected purely based on the short-term or small-group needs of the end-users with no coordination with the strategic leadership or senior management.

5.2.3 Managed stage

In the Managed stage, companies have standardized ITPM process to help with objective selection of projects. Metrics like Return on Investment and Net Present Value are calculated and reviewed (albeit annually) for the projects being managed.

5.2.4 Synchronized stage

In the Synchronized stage, IT investments are in close alignments with business strategy. The metrics for measuring performance evolve and the review process is used to weed out underperforming projects. Managers assess risks associated with the project: delays, cost overruns, strategic misalignment and end-user acceptance. One of the key differences between the Managed and Synchronized stages of model is the frequency of the project reviews. In Synchronized stage, projects are reviewed more frequently to stay in alignment with the corporate strategy. Typical questions asked during the review process in the Synchronized stage are: What is the business need? What do we want from the IT investment? What are the business outcomes we expect (increased customer satisfaction, increased revenues, decreased costs)? What will be the total cost of ownership (what will be the increased cost of support and infrastructure)? Higher cost projects should go through more rigorous analysis. Data

analysis from the survey conducted by Jeffery and Leliveld [3] showed that Synchronized ITPM process improved the return-on-asset (ROA) performance. The most valued benefit of ITPM is the improved business-strategy alignment.

For example: A CIO of a consumer-goods company used six criteria related to their strategic objectives for reviewing the projects [3]: financial return, consumer focus, supply-chain business benefit, technology efficiency, knowledge advantage and work-life balance. The review results showed good alignment with the corporate strategy and IT projects received wider support from senior business management. Overall IT budget was increased in the following budget period because the pending high-value IT projects was visible to the senior management through ITPM.

Quantifying IT's business value is not easy. This, combined with the typical lack of communication between business and IT leaders create significant organizations challenges for the successful implementation of ITPM. IT staff typically lack basic knowledge of financial concepts and business staff do not understand the challenges within IT projects. Jeffery and Leliveld recommends four approaches to make ITPM work [3]; staged implementation strategy, creating a process for upgrading ITPM, ensuring that the staff is trained and involve business-side people from the beginning.

6 Agile IT Architecture

Architecture is an integrated set of technical choices used to guide the organization in satisfying business needs [2]. By taking an architectural approach, businesses can create an IT infrastructure that enables integration of business processes which can

evolve over time and maintain balance of cost, performance and scalability. Architecture has to evolve based on changing needs of the business and the technology landscape. Well documented IT architecture and guidelines will help with its evolution and make the organization more efficient in the long term. A common understanding of IT architectural guidelines in the enterprise helps in achieving compatibility among the internal and external software applications. It also helps with faster designs of new applications and clearer future migration paths. Common architecture also reduces the cost of supporting the infrastructure because of the reduced cost of training the staff and the potential for volume discounts from vendors of software, hardware and services.

Architectural guidelines can be documented for the computing platforms, communication networks, data, applications and workflows. For example, enterprises may choose a hardware platform (example: x86, RISC) or specific operating system (example: Windows, Linux) for the desktop and server environments. Architectural documents can clearly define the common data structures shared among multiple applications (example: Customer, product and financial data). Common repository of business object definitions and data would make integration of disparate systems much faster. Storage of data and access control to data should also be standardized for privacy, compliance and security. Software applications developed and used across the enterprise should also be standardized to increase operational efficiencies and reduce the number of integration points. Common applications can also increase the depth of internal knowledgebase, resulting in additional productivity from the users. Documenting the standard workflow processes also can help in expediting the improvements in IT.

The technical architectural decisions on IT can be grouped into two areas, the enterprise application architecture and the enterprise integration architecture. As the firm grows, the number of software applications grows along with it. Many of these software applications are purchased or rented (Software as a service) from third party vendors. Firm cannot influence the technical architecture of these applications. However, it has complete control over the applications that are custom-built for the firm. These applications tend to be strategically most important to the company as well.

6.1 Agile Enterprise Application Architecture

In this section we will review the techniques and tools that can be used for creating agile enterprise software application architecture. I assume the size of these applications to be 100s of 1000s or millions of lines of code, with large number of expected users and Gigabytes or Terabytes of data. Building agile applications require a flexible software framework and complementary set of software design and development tools for maximum productivity. Investing in the framework and tools for building software applications may make sense only in cases where the gain in productivity over the life of the application justifies the investment in them. The software architect, along with the IT manager will have to find the right balance between the budget and required agility. For example, if the application being built is the core service that is expected to be used for the life of the company, it makes sense to invest in making the application flexible. However, if the application being built to address a peripheral functionality that could be addressed by other means, investing in agility would not be prudent.

6.1.1 Logical Software Layering

One of the primary techniques used for creating a flexible application is to use logical software layers for presentation (user interface), domain (business logic) and data source [10]. Layering is a classic concept used in numerous scenarios in the software industry (For example, the TCP/IP network stack layer). One of the biggest advantages of layering is that each layer can be replaced with an equivalent layer from a different provider with relative ease. Since the boundaries between the layers are determined by the well-defined interfaces, neighboring layers can continue to operate as if nothing has changed. Thus, layering can be used to add to or replace new presentation layers like a thin client or rich client [11]. It can also be used to replace the database with a new provider (for example: replacing IBM DB2 with Oracle or vice versa). These logical layers can be used to physically separate the software application into different servers to increase the capacity, performance or scalability of the system. Layered architecture also helps the application functionality to be exposed as a service.

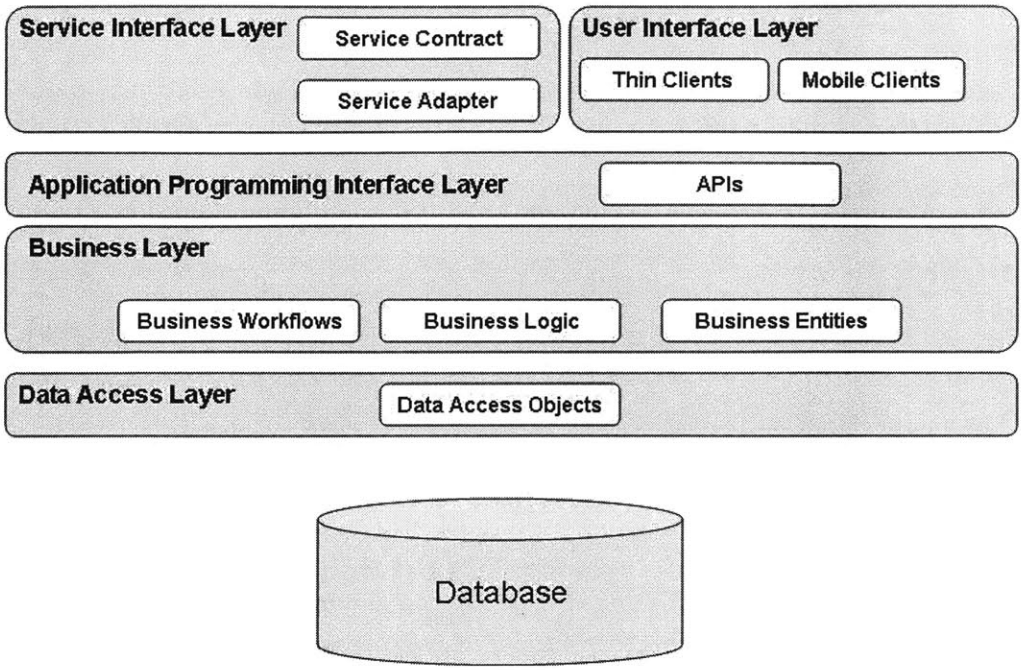

Figure 3 Software Layers

6.1.2 Component based design

One of the ways to quickly assemble software application functionality is to reuse software components [12] existing within the company or buy and assemble off-the-shelf software components. Well designed software components provide stable interfaces and implement a well-defined function that can be used in many scenarios. Modular designs will increase the potential for reuse and sharing across product lines, resulting in faster delivery [5]. System level components are more suitable for this type of reuse. For example, the WININET component made it easier to reuse HTTP, FTP and Gopher input/output. Re-architecting Internet Explorer 3.0 to use component based

was a critical design decision that helped Microsoft in closing the gap with Netscape [5] and eventually beat them in the browser war in the 90s. Component based design also helped Microsoft gain new partnerships. For example, AOL was able to use Internet Explorer as a browser by re-branding it easily [5].

Even though software reuse is a powerful concept, it is hard to achieve high level of reuse through components. The problem lies in the fact that it is hard to make really simple and generic interfaces. If the interface is simple, it is not going to be useful in a wide variety of usage scenarios. If the component tries to support many scenarios, the complexity of interface increases and the thus it becomes less attractive to the potential users. Software components written to address domain-specific problems have a higher chance of being used and to really save time. Obviously, system level components will be reused much more widely compared to application level components. For example: An XML parser component would be a good system level component that can be reused by many applications in many domains. An interest rate calculator component may be a useful in the context of software applications within a bank. The component could encapsulate the business rules of how the rates are calculated and the user applications can use the component as a 'black box' that provides interest rates.

6.1.3 Flexible Application-Development Frameworks

Applications that are built with flexible and extensible programming models can significantly simplify customization projects. In such systems, pre-built business objects allow custom programming hooks (or event handlers) and custom workflows be attached during the implementation phase. For example, a customer order system could

be customized to generate the manufacturing schedule automatically, when the customer confirms purchase. Some enterprise software applications like Ramco Virtual Works, component based application framework is built on layered business components that allow the addition of custom business logic in a quick and cost effective manner. Radisson Edwardian used this system to implement 400 screens and offered access to the application over the web to 2000 employees in 31.5 person months. This is a productivity level among the top 5% in the industry [4].

6.2 Agile Enterprise Integration Architecture

No single software application can fulfill all the needs of an enterprise. Over years, companies build or buy numerous software applications to address specific needs. One of the biggest challenges in IT organizations with large number of software applications is the need to integrate these applications running on disparate platforms into a cohesive engine that execute transactions efficiently and provide functionality that is part of the core strategy. Integrating applications is a challenging task because of differences in software platforms, operating systems, programming languages and data formats. Integration solutions will also have to overcome unreliable and slow networks and the frequent changes within the applications. Some of the classic techniques used for integrating applications are: File Transfer, Shared Database, Remote or Local Procedure calls (API based integrations). These techniques are still used on occasions if the specific integration in question is ideally suitable for it. Integrations based on Messaging and Service oriented architecture provide much higher flexibility and agility that these classic methods of integration.

6.2.1 Messaging

Messaging is a technology that reliably delivers asynchronous [20] messages between programs [19]. Each data packet is called a message, which is transmitted through logical pathways called channels or queues. A channel behaves like a collection of messages that can be simultaneously accessed by multiple computers across the network. A 'sender' is a program that originates the message and a 'receiver' is a program that consumes the message.

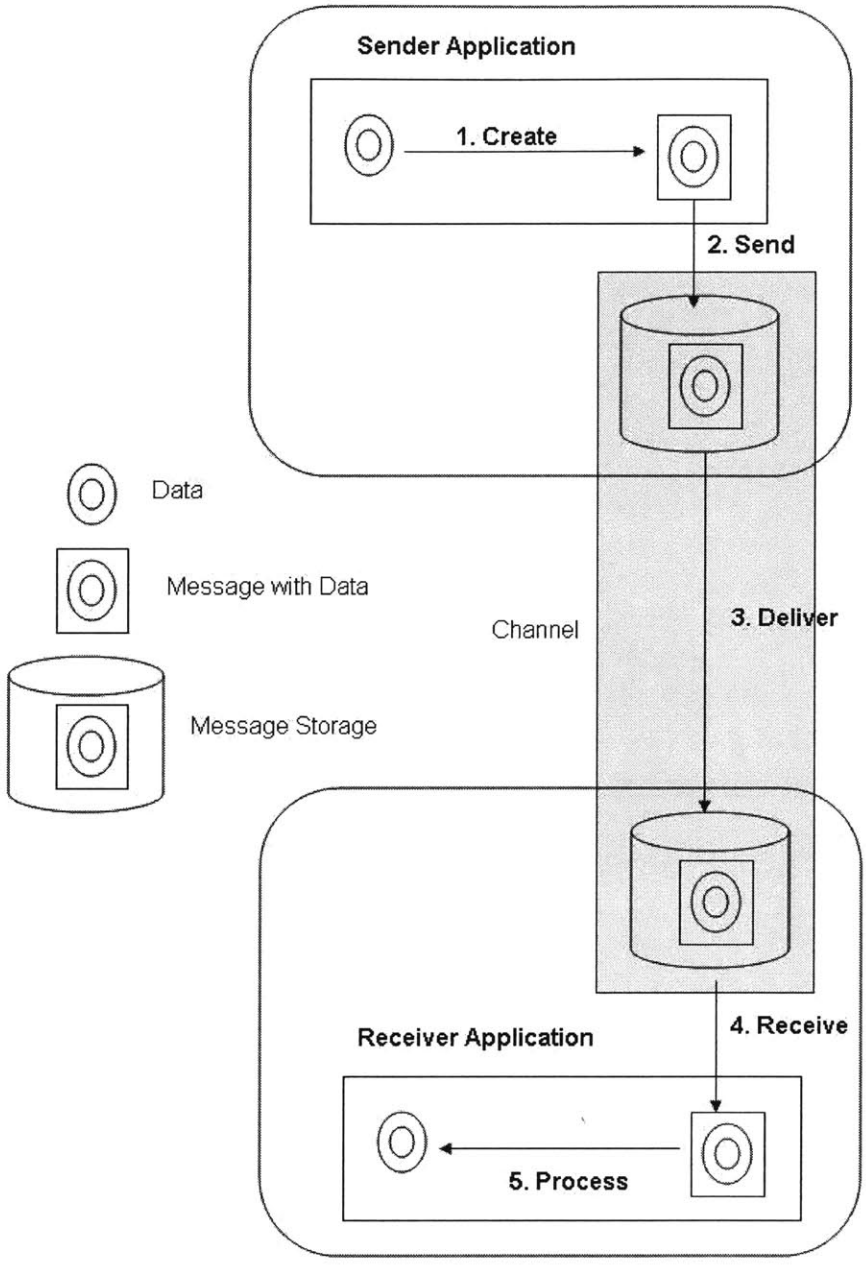

Figure 4 Messaging

The message contains two parts: a header and a body. The header part of the message contains administrative data like the sender, receiver and routing information.

Body of the message is normally a data structure (example: a customer order in XML format) that is meaningful to the sender and receiver.

Messaging is usually very fast, and it is more reliable than the integration options like the Remote Procedure calls (RPC). With messaging, applications can share data across computers without regard to platforms and languages. Messaging is done asynchronously, meaning that the sender and receiver need not be running simultaneously to exchange messages. This is similar to a voicemail in the phone system. Caller can leave a voicemail and the receiver can pick up the voicemail at any time that is convenient for him/her. Messaging is more reliable than RPC because messaging uses relational database to store the message before forwarding it. Messaging can also be made to guarantee delivery by retrying, if the receiver computer is off the network for any reason.

There are some drawbacks associated with using messaging for integrations. The asynchronous nature of the programming model is different from the typical event driven programming models and programmers will have to deal with the additional complexities of multi-threaded programming [21]. Even though the delivery of messages is guaranteed by the system, the sequence of delivery is not guaranteed. If the messages rely on sequential execution, this could be a problem. There is also an additional performance overhead while using messaging systems. Messaging systems do not yet have a common standard across platforms, creating the potential for a

vendor lock-in. Despite these disadvantages, messaging is still a valid option for reliable, flexible integration between disparate systems.

6.2.2 Service Oriented Architecture

Large Enterprise Resource Planning (ERP, [15]) systems bring efficiency to transactional systems, but they are inflexible and slow to adapt to new challenges in the market. New methods of Enterprise Application Integrations like Service Oriented Architecture are shown to significantly increase the flexibility and agility of integration projects. Service Oriented Architecture (SOA) can be considered as a logical extension to the earlier, attempts to reuse software code like object oriented design and component based design. SOA is software industry's answer to the complexities of integrating large software applications across platforms and organizational boundaries. With service orientation's openness and modularity, companies can approach integration projects with applications on disparate platforms one at a time without trying to consolidate their applications into one platform. Service Orientation is crucial to integration because it allows reuse of entire applications and continues to get value from previous investments [16].

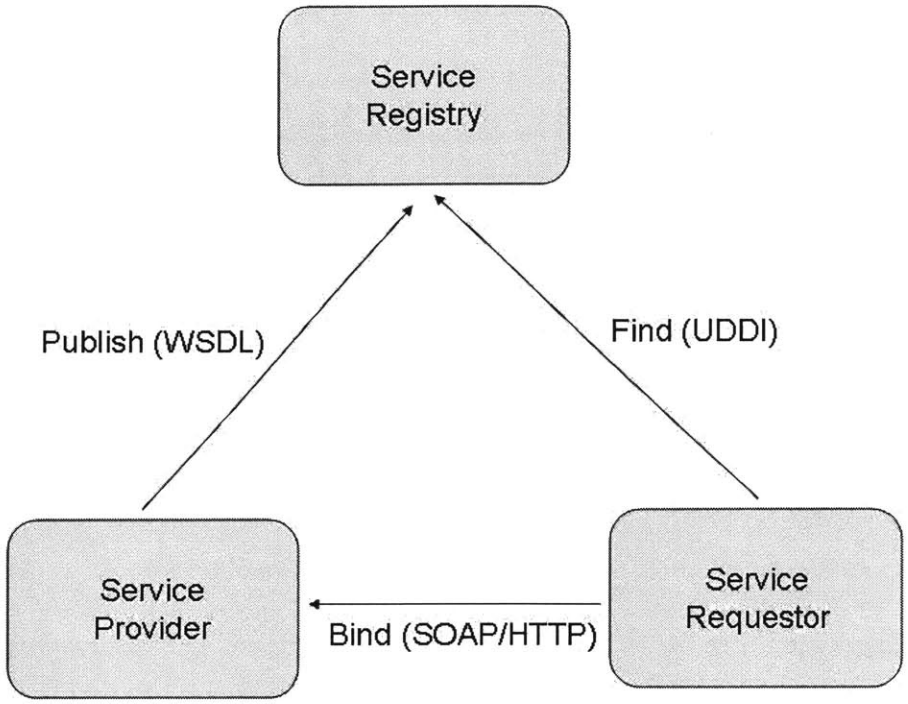

Figure 5 Web service lookup

The key concepts in SOA are the service, message, dynamic discovery and web services. The service is a functionality that is exposed for reuse by other software applications. The interface for the service is platform-neutral. The service can be invoked without a long lasting session and it is self-contained. Service application maintains its own state information. The platform independence of the service creates a very dynamic and open environment for integration between applications or with business partners, thereby increasing the agility of the organization.

The contract for the service interface is in the platform and language neutral XML (Extensible Markup Language [13]) message format. The flexibility and extensibility of XML messages create a very dynamic environment for integrating between systems. Old style API integrations would have had to overcome numerous obstacles like platform mismatch, language mismatch etc. With XML, none of those become a problem.

The directory service in an SOA provides the intermediary role between service providers and consumers. Providers of the services will register in the directory and consumers query the directory for service providers. Directory services can categorize the listed services and provide search capabilities. Providing the directory service decouples the direct link between the provider and consumer and lets consumers choose from a wide array of providers (instead of being stuck with one vendor). Interface contract based programming also allows the consumer to switch vendors at a lower cost.

The concept of SOA has been around for a long time, but web services [14] made it prominent. We services use software standards like XML, WSDL, UDDI and communication protocols like TCP/IP, HTTP and SOAP. New enterprise software applications in the market normally include a service interface to allow integration with other applications. Tools vendors are providing ways to add service facades to legacy applications, making even the legacy applications ready for integration.

The diagram below shows the typical layers in a service oriented infrastructure. The front end of each application is the service layer that accepts XML based messages and responds with the required data. Each of these service interfaces provides a well-defined business function, for example: credit card authorization, customer order entry, shipping label etc. Access to these services may be limited to the corporate intranet, or may be available to business partners through extranet [18] or available to other companies through public internet.

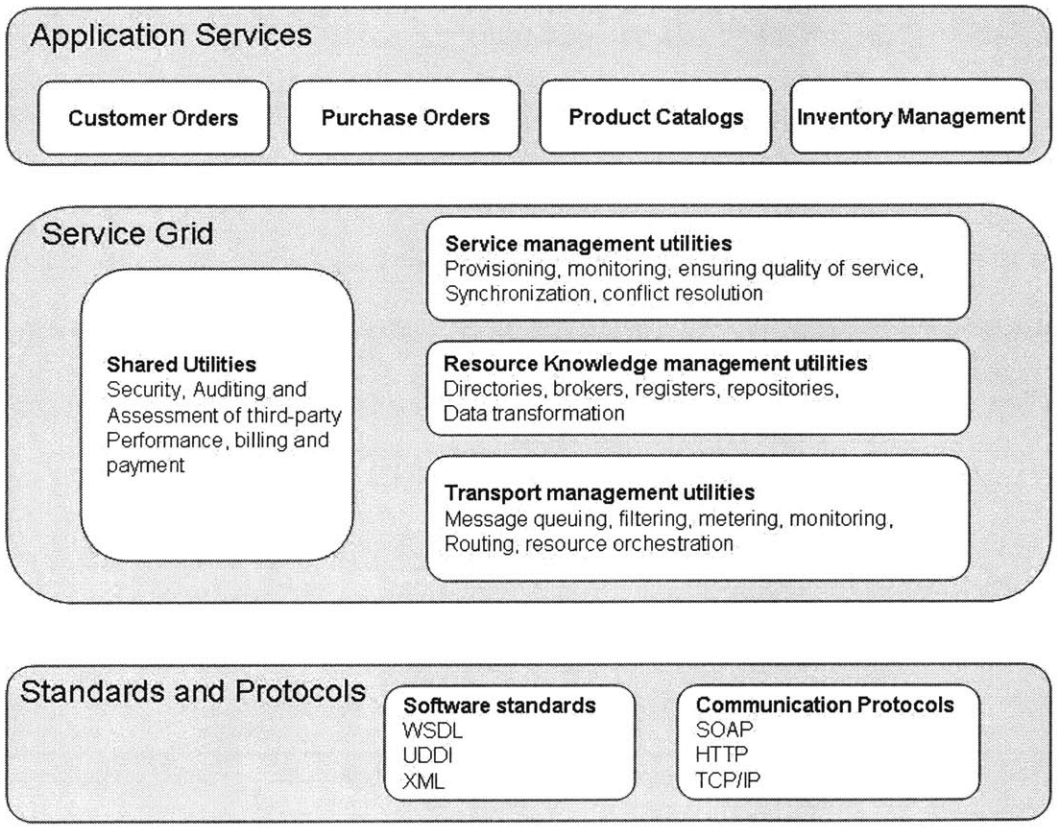

Figure 6 Integration using Service Oriented Architecture. Source [8]

This service interface layer interacts with the core business layer through custom protocols or through other middleware applications (example: WebSphere from IBM, BizTalk from Microsoft, NetWeaver from SAP). The middleware layer is normally called a service grid, which provides shared utilities (security, auditing, and billing), service management (provisioning, conflict resolution), resource knowledge management (brokers, registries, data transformations) and transport management (message queue, filters, routers).

To illustrate the use of the service oriented architecture, we can use an example of a fictitious large car dealership which provides highly personalized services to its customers. For example, if the customer would like to purchase a car with special accessories, dealer should be able to instantly search and find out if a car that meets customer's needs is available in stock within the same shop or at any nearby dealerships. Dealer should be able to place a request to hold the car for the customer or transfer it from the other dealership. If the customer needs a loan to buy the car, the dealer should be able to do an instant search to find potential banks that are willing to offer the loan and the interest rates and terms of service for each loan.

Each of these business functions is a valuable service for the customer and thereby important to the dealership. These functions require the dealership to create integration points with other dealerships, the manufacturer, the bank etc. Without a service oriented approach, the dealer would have created either a single monolithic application or a

series or point-applications that constantly needed changes (because of changes made by business partners). Each integration point would have been a brittle tight coupling that would have added to the maintenance nightmare for IT staff. Service oriented architecture provides a well-defined contract (XML schema) as the interface that never changes. The interfaces in SOA are platform-neutral and programming language-neutral. With SOA, the dealership can create a 'dashboard' style application by consolidating the services provided by the different business partners into a single, seamless application. Changes internal to the business partners do not affect the IT systems at this dealership because the integration interface stays the same. Since the SOA interfaces are loosely coupled, if at any time the business partners change, the application can point to the interface provided by the new partner and continue the service without interruptions.

The value of SOA is most tangible at the edge of the enterprise IT. This is because the edge is where platform and language incompatibilities become most visible. Dell Computer is an example for a company that makes significant use of SOA [18]. Dell spends 70% of its revenue on direct materials. So, relationships with component suppliers are critical to the company's strategy. Even small savings in the supply chain can cause significant impact to the bottom line. Traditionally, Dell had to hold substantial inventory in the supply chain to ensure timely delivery of products to the customers. Dell's business goal is to fulfill orders in 5 days, but the suppliers used to take as much as 45 days to deliver the components. With the SOA integration with suppliers, Dell now generates and distributes a new manufacturing schedule every two

hours and distributes it to the extranet. Dell was able to cut the inventory buffers to just 3-5 hours. They were able to remove stock rooms from the plants and add more production lines, increasing the factory utilization by one-third [18].

7 Software Engineering

7.1.1 Code generation

In large software projects, developers may spend significant percentage of their time writing repetitive code (for example, writing code to map business objects into database tables, creating APIs for use by the user interface layer, etc). Generating code based on templates to handle such repetitive tasks can save significant amount of programming time. Code generation also provides the additional benefit ease of future maintenance and enhancements. For example, by changing the template code can, developer can add additional features for all business objects in one shot, instead of doing it for one business object at a time.

7.1.2 Automated Testing

Automated testing gives the developers more freedom to experiment during the product development cycle. For example, both Microsoft and Netscape used automated testing to expedite product delivery cycles during the browser war [5]. Test automation does not eliminate the need for manual testing, but it can significantly reduce the load on the Quality assurance teams.

7.1.3 Synchronize and stabilize

Classic software development processes like the Waterfall model may not be suitable for a fast-changing environment. Companies like Microsoft and Netscape use the 'Synchronize-and-Stabilize' process to stay flexible during the software development cycle [5]. The idea is to not 'freeze' the design early during the development cycle and evolve it as the developers get more ideas and feedback. Daily builds would synchronize the changes with the rest of the product team continuously. Interim milestones during the project allowed the developers to stabilize features and show progress. Among the prioritized features, most important ones were completed first and the milestones gave the opportunity to review the completed work and make any mid-course corrections.

8 Conclusions

In conclusion, there is no silver bullet that can create an agile Information Technology infrastructure and keep it continuously synchronized with business strategy. Both business and technology leaders have to make a series of conscious decisions to keep the two sides synchronized. This thesis explained a series of management and technology frameworks that could increase the agility of IT and keep it in sync with business strategy. These frameworks would help only if the business and technology leaders understand each others' language and are able to communicate effectively. Business leaders should be able to clearly articulate both the long term strategy and also understand the technology implications of the changes in strategy. Technology

leaders must be able to understand the business strategy and align the IT infrastructure to match the strategy. Technology leaders must be able to guide the IT staff in making the right architectural decisions that would allow flexibility and agility for the long term.

9 Bibliography

1. Peter Weill, Mani Subramani and Marianne Broadbent. Building IT Infrastructure for Strategic Agility. Sloan Management Review 2002
2. Peter Weil and Marianne Broadbent. Leveraging the new Infrastructure. Harvard Business School Press, 1998
3. Mark Jeffrey and Ingmar Leliveld. Best Practices in IT Portfolio Management. Sloan Management Review 2004
4. C.K. Prahalad and M.S. Krishnan, The Dynamic Synchronization of Strategy and Technology. Sloan Management Review, 2004
5. Michael A. Cusumano, David B. Yoffie, Competing on the internet time: Lessons From Netscape & Its Battle with Microsoft. Simon & Schuster, 1998
6. Jeanne W. Ross. Johnson & Johnson: Building an Infrastructure to Support Global Operations. CISR Working Paper, 1995
7. Benn Konsynski and Michael Vitale, Baxter Healthcare Corporation: ASAP Express, Harvard Business School Case Study, 1998
8. John Hagel III and John Seely Brown, Your Next IT Strategy. Harvard Business Review, 2001

9. Jeanne W. Ross, Peter Weill, David Robertson, Enterprise Architecture As Strategy: Creating a Foundation for Business Execution. Harvard Business School Press, 2006
10. Martin Fowler, Patterns of Enterprise Application Architecture. Addison Wesley, 2002
11. "Rich Client" software application. The terms "Fat", "Rich", "Thin" clients are used to indicate if the client stores any data or does any processing. http://en.wikipedia.org/wiki/Client_(computing)
12. Software components http://en.wikipedia.org/wiki/Software_componentry
13. Extensible Markup Language http://www.w3.org/XML/
14. Web Service http://en.wikipedia.org/wiki/Web_service
15. Enterprise Resource Planning (ERP) http://en.wikipedia.org/wiki/Enterprise_resource_planning
16. Michael N. Huhns and Muninder P. Singh, Service-Oriented Computing: Key Concepts and Principles. IEEE Internet Computing, 2005
17. Michael Green, Michael J. Shaw, Supply-Chain Integration through Information Sharing: Channel Partnership between Wal-Mart and Procter & Gamble
18. Extranet: http://en.wikipedia.org/wiki/Extranet
19. Gregor Hohpe, Bobby Woolf, Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison Wesley, 2003
20. Asynchronous Communication: http://en.wikipedia.org/wiki/Asynchronous_communication
21. Multithreading http://en.wikipedia.org/wiki/Multithreading