

Enterprise Resource Planning (ERP)

Enterprise resource planning (ERP) is a company-wide computer software system used to manage and coordinate all the resources, information, and functions of a business from shared data stores. An ERP system has a service-oriented architecture with modular hardware and software units or "services" that communicate on a local area network. The modular design allows a business to add or reconfigure modules (perhaps from different vendors) while preserving data integrity in one shared database that may be centralized or distributed.

Origin of the term

MRP vs. ERP — Manufacturing management systems have evolved in stages over the past 30 years from a simple means of calculating materials requirements to the automation of an entire enterprise. Around 1980, over-frequent changes in sales forecasts, entailing continual readjustments in production, as well as inflexible fixed system parameters, led MRP (Material Requirement Planning) to evolve into a new concept : Manufacturing Resource Planning (or MRP3) and finally the generic concept Enterprise Resource Planning (ERP)

The initials ERP originated as an extension of MRP (material requirements planning; later manufacturing resource planning) and CIM (Computer Integrated Manufacturing). It was introduced by research and analysis firm Gartner in 1990. ERP systems now attempt to cover all core functions of an enterprise, regardless of the organization's business or charter. These systems can now be found in non-manufacturing businesses, non-profit organizations and governments.

To be considered an ERP system, a software package must provide the function of at least two systems. For example, a software package that provides both payroll and accounting functions could technically be considered an ERP software package

Examples of modules in an ERP which formerly would have been stand-alone applications include: Product lifecycle management, Supply chain management (e.g. Purchasing, Manufacturing and Distribution), Warehouse Management, Customer Relationship Management (CRM), Sales Order Processing, Online Sales, Financials, Human Resources, and Decision Support System.

Overview of ERP Solutions

Some organizations — typically those with sufficient in-house IT skills to integrate multiple software products — choose to implement only portions of an ERP system and develop an external interface to other ERP or stand-alone systems for their other application needs. For example, one may choose to use human resource management system from one vendor, and perform the integration between the systems themselves.

This is common to retailers, where even a mid-sized retailer will have a discrete Point-of-Sale (POS) product and financials application, then a series of specialized applications to

handle business requirements such as warehouse management, staff rostering, merchandising and logistics. Ideally, ERP delivers a single database that contains all data for the software modules, which would include:

Manufacturing

Engineering, bills of material, scheduling, capacity, workflow management, quality control, cost management, manufacturing process, manufacturing projects, manufacturing flow
Supply chain management

Order to cash, inventory, order entry, purchasing, product configurator, supply chain planning, supplier scheduling, inspection of goods, claim processing, commission calculation

Financials

General ledger, cash management, accounts payable, accounts receivable, fixed assets

Project management Costing, billing, time and expense, performance units, activity management.

Human resources

Human resources, payroll, training, time and attendance, rostering, benefits

Customer relationship management

Sales and marketing, commissions, service, customer contact and call center support

Data warehouse and various self-service interfaces for customers, suppliers, and employees

Access control - user privilege as per authority levels for process execution

Customization - to meet the extension, addition, change in process flow

Enterprise resource planning is a term originally derived from manufacturing resource planning (MRP II) that followed material requirements planning (MRP). MRP evolved into ERP when "routings" became a major part of the software architecture and a company's capacity planning activity also became a part of the standard software activity. ERP systems typically handle the manufacturing, logistics, distribution, inventory, shipping, invoicing, and accounting for a company. ERP software can aid in the control of many business activities, including sales, marketing, delivery, billing, production, inventory management, quality management and human resource management. ERP systems saw a large boost in sales in the 1990s as companies faced the Y2K problem in their legacy systems. Many companies took this opportunity to replace their legacy information systems with ERP systems. This rapid growth in sales was followed by a slump in 1999, at which time most companies had already implemented their Y2K solution.

ERPs are often incorrectly called back office systems indicating that customers and the general public are not directly involved. This is contrasted with front office systems like customer relationship management (CRM) systems that deal directly with the customers, or the eBusiness systems such as eCommerce, eGovernment, eTelecom, and eFinance, or supplier relationship management (SRM) systems.

ERPs are cross-functional and enterprise wide. All functional departments that are involved in operations or production are integrated in one system. In addition to manufacturing, warehousing, logistics, and information technology, this would include accounting, human resources, marketing and strategic management.

ERP II means open ERP architecture of components. The older, monolithic ERP systems became component oriented.

EAS — Enterprise Application Suite is a new name for formerly developed ERP systems which include (almost) all segments of business, using ordinary Internet browsers as thin clients.

Best practices are incorporated into most ERP vendor's software packages. When implementing an ERP system, organizations can choose between customizing the software or modifying their business processes to the "best practice" function delivered in the "out-of-the-box" version of the software. Prior to ERP, software was developed to fit the processes of an individual business. Due to the complexities of most ERP systems and the negative consequences of a failed ERP implementation, most vendors have included "Best Practices" into their software. These "Best Practices" are what the Vendor deems as the most efficient way to carry out a particular business process in an Integrated

Enterprise-Wide system. A study conducted by Lugwigshafen University of Applied Science surveyed 192 companies and concluded that companies which implemented industry best practices decreased mission-critical project tasks such as configuration, documentation, testing and training. In addition, the use of best practices reduced over risk by 71% when compared to other software implementations.

The use of best practices can make complying with requirements such as IFRS, Sarbanes-Oxley or Basel II easier. They can also help where the process is a commodity such as electronic funds transfer. This is because the procedure of capturing and reporting legislative or commodity content can be readily codified within the ERP software, and then replicated with confidence across multiple businesses who have the same business requirement.

Implementation

Businesses have a wide scope of applications and processes throughout their functional units; producing ERP software systems that are typically complex and usually impose significant changes on staff work practices. Implementing ERP software is typically too complex for "in-house" skill, so it is desirable and highly advised to hire outside consultants who are professionally trained to implement these systems. This is typically the most cost effective way. There are three types of services that may be employed for - Consulting, Customization, Support. The length of time to implement an ERP system depends on the size of the business, the number of modules, the extent of customization, the scope of the change and the willingness of the customer to take ownership for the project. ERP systems are modular, so they don't all need be implemented at once. It can be divided into various stages, or phase-ins. The typical project is about 14 months and requires around 150 consultants. A small project (e.g., a company of less than 100 staff) may be planned and delivered within 3-9 months; however, a large, multi-site or multi-country implementation may take years.[citation needed] The length of the implementations is closely tied to the amount of customization desired.

To implement ERP systems, companies often seek the help of an ERP vendor or of third-party consulting companies. These firms typically provide three areas of professional services: consulting, customization and support. The client organisation may also employ independent program management, business analysis, change management and UAT specialists to ensure their business requirements remain a priority during implementation.

Data migration is one of the most important activities in determining the success of an ERP implementation. Since many decisions must be made before migration, a significant amount of planning must occur. Unfortunately, data migration is the last activity before the production phase of an ERP implementation, and therefore receives minimal attention due to time constraints. The following are steps of a data migration strategy that can help with the success of an ERP implementation:

1. Identifying the data to be migrated
2. Determining the timing of data migration
3. Generating the data templates
4. Freezing the tools for data migration
5. Deciding on migration related setups
6. Deciding on data archiving.

Process preparation

ERP vendors have designed their systems around standard business processes, based upon best business practices. Different vendor(s) have different types of processes but they are all of a standard, modular nature. Firms that want to implement ERP systems are consequently forced to adapt their organizations to standardized processes as opposed to adapting the ERP package to the existing processes. Neglecting to map current business processes prior to starting ERP implementation

is a main reason for failure of ERP projects. It is therefore crucial that organizations perform a thorough business process analysis before selecting an ERP vendor and setting off on the implementation track. This analysis should map out all present operational processes, enabling selection of an ERP vendor whose standard modules are most closely aligned with the established organization. Redesign can then be implemented to achieve further process congruence. Research indicates that the risk of business process mismatch is decreased by:

- * linking each current organizational process to the organization's strategy;
- * analyzing the effectiveness of each process in light of its current related business capability;
- * understanding the automated solutions currently implemented.

ERP implementation is considerably more difficult (and politically charged) in organizations structured into nearly independent business units, each responsible for their own profit and loss, because they will each have different processes, business rules, data semantics, authorization hierarchies and decision centers. Solutions include requirements coordination negotiated by local change management professionals or, if this is not possible, federated implementation using loosely integrated instances (e.g. linked via Master Data Management) specifically configured and/or customized to meet local needs.

A disadvantage usually attributed to ERP is that business process redesign to fit the standardized ERP modules can lead to a loss of competitive advantage. While documented cases exist where this has indeed materialized, other cases show that following thorough process preparation ERP systems can actually increase sustainable competitive advantage.

Configuration

Configuring an ERP system is largely a matter of balancing the way you want the system to work with the way the system lets you work. Begin by deciding which modules to install, then adjust the system using configuration tables to achieve the best possible fit in working with your company's processes. Modules — Most systems are modular simply for the flexibility of implementing some functions but not others. Some common modules, such as finance and accounting are adopted by nearly all companies implementing enterprise systems; others however such as human resource management are not needed by some companies and therefore not adopted. A service company for example will not likely need a module for manufacturing. Other times companies will not adopt a module because they already have their own proprietary system they believe to be superior. Generally speaking the greater number of modules selected, the greater the integration benefits, but also the increase in costs, risks and changes involved.

Configuration Tables – A configuration table enables a company to tailor a particular aspect of the system to the way it chooses to do business. For example, an organization can select the type of inventory accounting – FIFO or LIFO – it will employ or whether it wants to recognize revenue by geographical unit, product line, or distribution channel.

So what happens when the options the system allows just aren't good enough? At this point a company has two choices, both of which are not ideal. It can re-write some of the enterprise system's code, or it can continue to use an existing system and build interfaces between it and the new enterprise system. Both options will add time and cost to the implementation process. Additionally they can dilute the system's integration benefits. The more customized the system becomes the less possible seamless communication becomes between suppliers and customers.

Consulting services

Many organizations did not have sufficient internal skills to implement an ERP project. This resulted in many organizations offering consulting services for ERP implementation. Typically, a consulting team was responsible for the entire ERP implementation including planning, training, testing, implementation, and delivery of any customized modules. Examples of customization includes additional product training; creation of process triggers and workflow; specialist advice to improve how the ERP is used in the business; system optimization; and assistance writing reports, complex data extracts or implementing Business Intelligence.

For most mid-sized companies, the cost of the implementation will range from around the list price of the ERP user licenses to up to twice this amount (depending on the level of customization required). Large companies, and especially those with multiple sites or countries, will often spend considerably more on the implementation than the cost of the user licenses -- three to five times more is not uncommon for a multi-site implementation.

Unlike most single-purpose applications, ERP packages have historically included full source code and shipped with vendor-supported team IDEs for customizing and extending the delivered code. During the early years of ERP the guarantee of mature tools and support for extensive customization was an important sales argument when a potential customer was considering developing their own unique solution in-house, or assembling a cross-functional solution by integrating multiple "best of breed" applications.

"Core system" Customization vs Configuration

Increasingly, ERP vendors have tried to reduce the need for customization by providing built-in "configuration" tools to address most customers' needs for changing how the out-of-the-box core system works. Key differences between customization and configuration include:

- * Customization is always optional, whereas some degree of configuration (e.g. setting up cost/profit centre structures, organisational trees, purchase approval rules, etc.) may be needed before the software will work at all.
- * Configuration is available to all customers, whereas customization allows individual customer to implement proprietary "market-beating" processes.
- * Configuration changes tend to be recorded as entries in vendor-supplied data tables, whereas customization usually requires some element of programming and/or changes to table structures or views.
- * The effect of configuration changes on the performance of the system is relatively predictable and is largely the responsibility of the ERP vendor. The effect of customization is unpredictable and may require time-consuming stress testing by the implementation team.
- * Configuration changes are almost always guaranteed to survive upgrades to new software versions. Some customizations (e.g. code that uses pre-defined "hooks" that are called before/after displaying data screens) will survive upgrades, though they will still need to be re-tested. More extensive customizations (e.g. those involving changes to fundamental data structures) will be overwritten during upgrades and must be re-implemented manually.

By this analysis, customizing an ERP package can be unexpectedly expensive and complicated, and tends to delay delivery of the obvious benefits of an integrated system. Nevertheless, customizing an ERP suite gives the scope to implement secret recipes for excellence in specific areas while ensuring that industry best practices are achieved in less sensitive areas.

Extension

In this context "Extension" refers to ways that the delivered ERP environment can be extended with third-party programs. It is technically easy to expose most ERP transactions to outside programs, e.g.

- * Scenarios to do with archiving, reporting and republishing (these easiest to achieve, because they mainly address static data);
- * Transactional data capture scenarios, e.g. using scanners, tills or RFIDs, are relatively easy (because they touch existing data);

...however because ERP applications typically contain sophisticated rules that control how master data can be created or changed, some scenarios are very difficult to implement.

Maintenance and support services

Maintenance and support services involves monitoring and managing an operational ERP system. This function is often provided in-house using members of the IT department, or may be provided by a specialist external consulting and services company.

Advantages

In the absence of an ERP system, a large manufacturer may find itself with many software applications that cannot communicate or interface effectively with one another. Tasks that need to interface with one another may involve:

- * Integration among different functional areas to ensure proper communication, productivity and efficiency
- * Design engineering (how to best make the product)
- * Order tracking, from acceptance through fulfillment
- * The revenue cycle, from invoice through cash receipt
- * Managing inter-dependencies of complex processes bill of materials
- * Tracking the three-way match between purchase orders (what was ordered), inventory receipts (what arrived), and costing (what the vendor invoiced)
- * The accounting for all of these tasks: tracking the revenue, cost and profit at a granular level.
- * ERP Systems centralize the data in one place. This eliminates the problem of synchronizing changes and can reduce the risk of loss of sensitive data by consolidating multiple permissions and security models into a single structure.

Some security features are included within an ERP system to protect against both outsider crime, such as industrial espionage, and insider crime, such as embezzlement. A data-tampering scenario, for example, might involve a disgruntled employee intentionally modifying prices to below-the-breakeven point in order to attempt to interfere with the company's profit or other sabotage. ERP systems typically provide functionality for implementing internal controls to prevent actions of this kind. ERP vendors are also moving toward better integration with other kinds of information security tools.

Disadvantages

Problems with ERP systems are mainly due to inadequate investment in ongoing training for the involved IT personnel - including those implementing and testing changes - as well as a lack of corporate policy protecting the integrity of the data in the ERP systems and the ways in which it is used.

Disadvantages

- * Customization of the ERP software is limited.
- * Re-engineering of business processes to fit the "industry standard" prescribed by the ERP system may lead to a loss of competitive advantage.
- * ERP systems can be very expensive (This has led to a new category of "ERP light" solutions)
- * ERPs are often seen as too rigid and too difficult to adapt to the specific workflow and business process of some companies—this is cited as one of the main causes of their failure.
- * Many of the integrated links need high accuracy in other applications to work effectively. A company can achieve minimum standards, then over time "dirty data" will reduce the reliability of some applications.
- * Once a system is established, switching costs are very high for any one of the partners (reducing flexibility and strategic control at the corporate level).
- * The blurring of company boundaries can cause problems in accountability, lines of responsibility, and employee morale.
- * Resistance in sharing sensitive internal information between departments can reduce the effectiveness of the software.
- * Some large organizations may have multiple departments with separate, independent resources, missions, chains-of-command, etc, and consolidation into a single enterprise may yield limited benefits.
- * The system may be too complex measured against the actual needs of the customers.
- * ERP Systems centralize the data in one place. This can increase the risk of loss of sensitive information in the event of a security breach.

Market share 2005 according to Gartner Dataquest^[1]

Vendor

ERP modeling, abbreviated ERP, is the process of reverse engineering an Enterprise Resource Planning software package in order to align it to an organizational structure.

Usage

Although ERP modeling could possibly be performed by several methodologies, this entry deals with ERP modeling using Object Process Methodology, or OPM.

OPM appears to be a usable methodology for modeling ERP systems, as the methodology focuses on optionality within objects and processes of an ERP system.

ERP modeling is done by analyzing the optionality within an ERP system to identify the different functions of the system that the end-using company needs, regarding its organizational structure. Reverse engineering both ERP system and organizational structure to the same level of granularity makes both layers compatible for aligning the package in the organization.

Theory

A Global Business Process Model is created which represents the whole ERP software product. This model is layered in 3 deeper levels.

* The first level is the System Configuration Level, which scopes on high-level optionality on the entire system. Option definition is therefore static: once a high-level option of the ERP system is chosen to be used within the organization, the choice cannot be made undone.

* One level deeper is the Object Level, which scopes on single data objects. The optionality on this level is more dynamic.

* The deepest level is the Occurrence level, which analyses single process occurrences. Because this level elaborates on object parameters, the optionality is very dynamic, meaning that options can easily be altered.

The meta model below depicts the optionality levels of ERP modeling.

The optionality leveling is used to reverse engineer the ERP system and the organizational structure to its full extent. Once properly mapped, both aspects are fully alignable or at least compatible to be matched.

The correct way to align both ERP and organizational models is as follows:

1. Convert the ERP system database to an object model
2. Construct a global business process model
3. Identify the system configuration-level business process alternatives
4. Identify the object-level variants of the business processes
5. Expose the occurrence-level business process options

Supply chain management (SCM) is the management of a network of interconnected businesses involved in the ultimate provision of product and service packages required by end customers (Harland, 1996). Supply Chain Management spans all movement and storage of raw materials, work-in-process inventory, and finished goods from point of origin to point of consumption (supply chain).

Idea

The definition an American professional association put forward is that Supply Chain Management encompasses the planning and management of all activities involved in sourcing, procurement, conversion, and logistics management activities. Importantly, it also includes coordination and collaboration with channel partners, which can be suppliers, intermediaries, third-party service providers, and customers. In essence, Supply Chain Management integrates supply and demand management within and across companies. More recently, the loosely coupled, self-organizing network of businesses that cooperates to provide product and service offerings has been called the Extended Enterprise.

Supply Chain Management can also refer to Supply chain management software which are tools or modules used in executing supply chain transactions, managing supplier relationships and controlling associated business processes.

Supply chain event management (abbreviated as SCEM) is a consideration of all possible occurring

events and factors that can cause a disruption in a supply chain. With SCEM possible scenarios can be created and solutions can be planned.

Supply Chain Management Problems

Supply chain management must address the following problems:

- * **Distribution Network Configuration:** Number, location and network missions of suppliers, production facilities, distribution centers, warehouses, cross-docks and customers.
- * **Distribution Strategy:** Including questions of operating control (centralized, decentralized or shared); delivery scheme (e.g., direct shipment, pool point shipping, Cross docking, DSD (direct store delivery), closed loop shipping); mode of transportation (e.g., motor carrier, including truckload, LTL, parcel; railroad; intermodal, including TOFC and COFC; ocean freight; airfreight); replenishment strategy (e.g., pull, push or hybrid); and transportation control (e.g., owner-operated, private carrier, common carrier, contract carrier, or 3PL). **Trade-Offs in Logistical Activities**

The above activities must be coordinated well together in order to achieve the least total logistics cost. Trade-offs exist that increase the total cost if only one of the activities is optimized. For example, full truckload (FTL) rates are more economical on a cost per pallet basis than less than truckload (LTL) shipments. If, however, a full truckload of a product is ordered to reduce transportation costs there will be an increase in inventory holding costs which may increase total logistics costs. It is therefore imperative to take a systems approach when planning logistical activities. These trade-offs are key to developing the most efficient and effective Logistics and SCM strategy.

- * **Information:** Integration of and other processes through the supply chain to share valuable information, including demand signals, forecasts, inventory, transportation, and potential collaboration etc.
- * **Inventory Management:** Quantity and location of inventory including raw materials, work-in-progress (WIP) and finished goods.
- * **Cash-Flow:** Arranging the payment terms and the methodologies for exchanging funds across entities within the supply chain.

Supply chain execution is managing and coordinating the movement of materials, information and funds across the supply chain. The flow is bi-directional.

Activities/functions

Supply chain management is a cross-function approach to manage the movement of raw materials into an organization, certain aspects of the internal processing of materials into finished goods, and then the movement of finished goods out of the organization toward the end-consumer. As organizations strive to focus on core competencies and becoming more flexible, they have reduced their ownership of raw materials sources and distribution channels. These functions are increasingly being outsourced to other entities that can perform the activities better or more cost effectively. The effect is to increase the number of organizations involved in satisfying customer demand, while reducing management control of daily logistics operations. Less control and more supply chain partners led to the creation of supply chain management concepts. The purpose of supply chain management is to improve trust and collaboration among supply chain partners, thus improving inventory visibility and improving inventory velocity.

Several models have been proposed for understanding the activities required to manage material movements across organizational and functional boundaries. SCOR is a supply chain management model promoted by the Supply Chain Council. Another model is the SCM Model proposed by the Global Supply Chain Forum (GSCF). Supply chain activities can be grouped into strategic, tactical, and operational levels of activities.

Strategic

- * **Strategic network optimization,** including the number, location, and size of warehousing, distribution centers, and facilities
- * **Strategic partnership with suppliers, distributors, and customers,** creating communication channels for critical information and operational improvements such as cross docking, direct shipping,

and third-party logistics

- * Product life cycle management, so that new and existing products can be optimally integrated into the supply chain and capacity management
- * Information Technology infrastructure, to support supply chain operations
- * Where-to-make and what-to-make-or-buy decisions
- * Aligning overall organizational strategy with supply strategy

Tactical

- * Sourcing contracts and other purchasing decisions.
- * Production decisions, including contracting, scheduling, and planning process definition.
- * Inventory decisions, including quantity, location, and quality of inventory.
- * Transportation strategy, including frequency, routes, and contracting.
- * Benchmarking of all operations against competitors and implementation of best practices throughout the enterprise.
- * Milestone payments
- * Focus on customer demand.

Operational

- * Daily production and distribution planning, including all nodes in the supply chain.
- * Production scheduling for each manufacturing facility in the supply chain (minute by minute).
- * Demand planning and forecasting, coordinating the demand forecast of all customers and sharing the forecast with all suppliers.
- * Sourcing planning, including current inventory and forecast demand, in collaboration with all suppliers.
- * Inbound operations, including transportation from suppliers and receiving inventory.
- * Production operations, including the consumption of materials and flow of finished goods.
- * Outbound operations, including all fulfillment activities, warehousing and transportation to customers.
- * Order promising, accounting for all constraints in the supply chain, including all suppliers, manufacturing facilities, distribution centers, and other customers.

Supply chain management

Organizations increasingly find that they must rely on effective supply chains, or networks, to successfully compete in the global market and networked economy. In Peter Drucker's (1998) new management paradigms, this concept of business relationships extends beyond traditional enterprise boundaries and seeks to organize entire business processes throughout a value chain of multiple companies.

During the past decades, globalization, outsourcing and information technology have enabled many organizations, such as Dell and Hewlett Packard, to successfully operate solid collaborative supply networks in which each specialized business partner focuses on only a few key strategic activities (Scott, 1993). This inter-organizational supply network can be acknowledged as a new form of organization. However, with the complicated interactions among the players, the network structure fits neither "market" nor "hierarchy" categories (Powell, 1990). It is not clear what kind of performance impacts different supply network structures could have on firms, and little is known about the coordination conditions and trade-offs that may exist among the players. From a systems perspective, a complex network structure can be decomposed into individual component firms (Zhang and Dilts, 2004). Traditionally, companies in a supply network concentrate on the inputs and outputs of the processes, with little concern for the internal management working of other individual players.

Therefore, the choice of an internal management control structure is known to impact local firm performance (Mintzberg, 1979).

In the 21st century, changes in the business environment have contributed to the development of supply chain networks. First, as an outcome of globalization and the proliferation of multinational companies, joint ventures, strategic alliances and business partnerships, there were found to be significant success factors, following the earlier "Just-In-Time", "Lean Manufacturing" and "Agile

Manufacturing" practices. Second, technological changes, particularly the dramatic fall in information communication costs, which are a significant component of transaction costs, have led to changes in coordination among the members of the supply chain network (Coase, 1998).

Many researchers have recognized these kinds of supply network structures as a new organization form, using terms such as "Keiretsu", "Extended Enterprise", "Virtual Corporation", "Global Production Network", and "Next Generation Manufacturing System".[4] In general, such a structure can be defined as "a group of semi-independent organizations, each with their capabilities, which collaborate in ever-changing constellations to serve one or more markets in order to achieve some business goal specific to that collaboration" (Akkermans, 2001).

The security management system for supply chain is described in ISO/IEC 28000 and ISO/IEC 28001 and related standards published jointly by ISO and IEC.

Developments in Supply Chain Management

Six major movements can be observed in the evolution of supply chain management studies:

Creation, Integration, and Globalization (Lavassani et al., 2008a), Specialization Phases One and Two, and SCM 2.0.

1. Creation Era

The term supply chain management was first coined by an American industry consultant in the early 1980s. However the concept of supply chain in management, was of great importance long before in the early 20th century, especially by the creation of the assembly line. The characteristics of this era of supply chain management include the need for large scale changes, re-engineering, downsizing driven by cost reduction programs, and widespread attention to the Japanese practice of management.

2. Integration Era

This era of supply chain management studies was highlighted with the development of Electronic Data Interchange (EDI) systems in the 1960s and developed through the 1990s by the introduction of Enterprise Resource Planning (ERP) systems. This era has continued to develop into the 21st century with the expansion of internet-based collaborative systems. This era of SC evolution is characterized by both increasing value-added and cost reduction through integration.

3. Globalization Era

The third movement of supply chain management development, globalization era, can be characterized by the attention towards global systems of supplier relations and the expansion of supply chain over national boundaries and into other continents. Although the use of global sources in the supply chain of organizations can be traced back to several decades ago (e.g. the oil industry), it was not until the late 1980s that a considerable number of organizations started to integrate global sources into their core business. This era is characterized by the globalization of supply chain management in organizations with the goal of increasing competitive advantage, creating more value added, and reducing costs through global sourcing.

4. Specialization Era—Phase One—Outsourced Manufacturing and Distribution

In the 1990s industries began to focus on core competencies and adopted a specialization model. Companies abandoned vertical integration, sold off non-core operations, and outsourced those functions to other companies. This changed management requirements by extending the supply chain well beyond the four walls and distributing management across specialized supply chain partnerships. This transition also re-focused the fundamental perspectives of each respective organization. OEMs became brand owners that needed deep visibility into their supply base. They had to control the entire supply chain from above instead of from within. Contract manufacturers had to manage bills of material with different part numbering schemes from multiple OEMs and support customer requests for work-in-process visibility and vendor-managed inventory (VMI).

The specialization model creates manufacturing and distribution networks composed of multiple, individual supply chains specific to products, suppliers, and customers who work together to design, manufacture, distribute, market, sell, and service a product. The set of partners may change according to a given market, region, or channel, resulting in a proliferation of trading partner

environments, each with its own unique characteristics and demands.

5. Specialization Era—Phase Two—Supply Chain Management as a Service

Specialization within the supply chain began in the 1980s with the inception of transportation brokerages, warehouse management, and non asset based carriers and has matured beyond transportation and logistics into aspects of supply planning, collaboration, execution and performance management.

At any given moment, market forces could demand changes within suppliers, logistics providers, locations, customers and any number of these specialized participants within supply chain networks. This variability has significant effect on the supply chain infrastructure, from the foundation layers of establishing and managing the electronic communication between the trading partners to the more complex

requirements, including the configuration of the processes and work flows that are essential to the management of the network itself.

Supply chain specialization enables companies to improve their overall competencies in the same way that outsourced manufacturing and distribution has done; it allows them to focus on their core competencies and assemble networks of best in class domain specific partners to contribute to the overall value chain itself – thus increasing overall performance and efficiency. The ability to quickly obtain and deploy this domain specific supply chain expertise without developing and maintaining an entirely unique and complex competency in house is the leading reason why supply chain specialization is gaining popularity.

Outsourced technology hosting for supply chain solutions debuted in the late 1990s and has taken root in transportation and collaboration categories most dominantly. This has progressed from the Application Service Provider (ASP) model from approximately 1998 through 2003 to the On-Demand model from approximately 2003-2006 to the Software as a Service (SaaS) model we are currently focused on today.

6. Supply Chain Management 2.0 (SCM 2.0)

Building off of globalization and specialization, SCM 2.0 has been coined to describe both the changes within the supply chain itself as well as the evolution of the processes, methods and tools that manage it in this new "era".

Web 2.0 is defined as a trend in the use of the World Wide Web that is meant to increase creativity, information sharing, and collaboration among users. At its core, the common attribute that Web 2.0 brings is it helps us navigate the vast amount of information available on the web to find what we are looking for. It is the notion of a usable pathway. SCM 2.0 follows this notion into supply chain operations. It is the pathway to SCM results – the combination of the processes, methodologies, tools and delivery options to guide companies to their results quickly as the complexity and speed of the supply chain increase due to the effects of global competition, rapid price fluctuations, surging oil prices, short product life cycles, expanded specialization, near/far and off shoring, and talent scarcity. SCM 2.0 leverages proven solutions designed to rapidly deliver results with the agility to quickly manage future change for continuous flexibility, value and success. This is delivered through competency networks composed of best of breed supply chain domain expertise to understand which elements, both operationally and organizationally, are the critical few that deliver the results as well as the intimate understanding of how to manage these elements to achieve desired results, finally the solutions are delivered in a variety of options as no-touch via business process outsourcing, mid-touch via managed services and software as a service (SaaS), or high touch in the traditional software deployment model.

Supply chain business process integration

Successful SCM requires a change from managing individual functions to integrating activities into key supply chain processes. An example scenario: the purchasing department places orders as requirements become appropriate. Marketing, responding to customer demand, communicates with several distributors and retailers as it attempts to satisfy this demand. Shared information between

supply chain partners can only be fully leveraged through process integration.

Supply chain business process integration involves collaborative work between buyers and suppliers, joint product development, common systems and shared information. According to Lambert and Cooper (2000) operating an integrated supply chain requires continuous information flow. However, in many companies, management has reached the conclusion that optimizing the product flows cannot be accomplished without implementing a process approach to the business. The key supply chain processes stated by Lambert (2004) [5] are:

- * Customer relationship management
- * Customer service management
- * Demand management
- * Order fulfillment
- * Manufacturing flow management
- * Supplier relationship management
- * Product development and commercialization
- * Returns management

Much has been written about demand management. Best in Class companies have similar characteristics. They include the following: a) Internal and external collaboration b) Lead time reduction initiatives c) Tighter feedback from customer and market demand d) Customer level forecasting

One could suggest other key critical supply business processes combining these processes stated by Lambert such as:

1. Customer service management
2. Procurement
3. Product development and commercialization
4. Manufacturing flow management/support
5. Physical distribution
6. Outsourcing/partnerships
7. Performance measurement

a) Customer service management process

Customer Relationship Management concerns the relationship between the organization and its customers. Customer service provides the source of customer information. It also provides the customer with real-time information on promising dates and product availability through interfaces with the company's production and distribution operations. Successful organizations use following steps to build customer relationships:

- * determine mutually satisfying goals between organization and customers
- * establish and maintain customer rapport
- * produce positive feelings in the organization and the customers

b) Procurement process

Strategic plans are developed with suppliers to support the manufacturing flow management process and development of new products. In firms where operations extend globally, sourcing should be managed on a global basis. The desired outcome is a win-win relationship, where both parties benefit, and reduction times in the design cycle and product development are achieved. Also, the purchasing function develops rapid communication systems, such as electronic data interchange (EDI) and Internet linkages to transfer possible requirements more rapidly. Activities related to obtaining products and materials from outside suppliers requires performing resource planning, supply sourcing, negotiation, order placement, inbound transportation, storage, handling and quality assurance, many of which include the responsibility to coordinate with suppliers in scheduling, supply continuity, hedging, and research into new sources or programs.

c) Product development and commercialization

Here, customers and suppliers must be united into the product development process, thus to reduce time to market. As product life cycles shorten, the appropriate products must be developed and successfully launched in ever shorter time-schedules to remain competitive. According to Lambert and Cooper (2000), managers of the product development and commercialization process must:

1. coordinate with customer relationship management to identify customer-articulated needs;
2. select materials and suppliers in conjunction with procurement, and
3. develop production technology in manufacturing flow to manufacture and integrate into the best supply chain flow for the product/market combination.

d) Manufacturing flow management process

The manufacturing process is produced and supplies products to the distribution channels based on past forecasts. Manufacturing processes must be flexible to respond to market changes, and must accommodate mass customization. Orders are processes operating on a just-in-time (JIT) basis in minimum lot sizes. Also, changes in the manufacturing flow process lead to shorter cycle times, meaning improved responsiveness and efficiency of demand to customers. Activities related to planning, scheduling and supporting manufacturing operations, such as work-in-process storage, handling, transportation, and time phasing of components, inventory at manufacturing sites and maximum flexibility in the coordination of geographic and final assemblies postponement of physical distribution operations.

e) Physical distribution

This concerns movement of a finished product/service to customers. In physical distribution, the customer is the final destination of a marketing channel, and the availability of the product/service is a vital part of each channel participant's marketing effort. It is also through the physical distribution process that the time and space of customer service become an integral part of marketing, thus it links a marketing channel with its customers (e.g. links manufacturers, wholesalers, retailers).

f) Outsourcing/partnerships

This is not just outsourcing the procurement of materials and components, but also outsourcing of services that traditionally have been provided in-house. The logic of this trend is that the company will increasingly focus on those activities in the value chain where it has a distinctive advantage and everything else it will outsource. This movement has been particularly evident in logistics where the provision of transport, warehousing and inventory control is increasingly subcontracted to specialists or logistics partners. Also, to manage and control this network of partners and suppliers requires a blend of both central and local involvement. Hence, strategic decisions need to be taken centrally with the monitoring and control of supplier performance and day-to-day liaison with logistics partners being best managed at a local level.

g) Performance measurement

Experts found a strong relationship from the largest arcs of supplier and customer integration to market share and profitability. By taking advantage of supplier capabilities and emphasizing a longterm supply chain perspective in customer relationships can be both correlated with firm performance. As logistics competency becomes a more critical factor in creating and maintaining competitive advantage, logistics measurement becomes increasingly important because the difference between profitable and unprofitable operations becomes more narrow. A.T. Kearney Consultants (1985) noted that firms engaging in comprehensive performance measurement realized improvements in overall productivity. According to experts internal measures are generally collected and analyzed by the firm including

1. Cost
2. Customer Service
3. Productivity measures
4. Asset measurement, and

5. Quality.

External performance measurement is examined through customer perception measures and "best practice" benchmarking, and includes 1) customer perception measurement, and 2) best practice benchmarking. Components of Supply Chain Management are 1. Standardization 2. Postponement 3. Customization

Theories of Supply Chain Management

Currently there exists a gap in the literature available in the area of supply chain management studies, on providing theoretical support for explaining the existence and the boundaries of supply chain management. Few authors such as Halldorsson, et al. (2003), Ketchen and Hult (2006) and Lavassani, et al. (2008b) had tried to provide theoretical foundations for different areas related to supply chain with employing organizational theories. These theories includes:

- * Resource-based view (RBV)
- * Transaction Cost Analysis (TCA)
- * Knowledge-based view (KBV)
- * Strategic Choice Theory (SCT)
- * Agency theory (AT)
- * Institutional theory (InT)
- * Systems Theory (ST)
- * Network Perspective (NP)

Supply chain sustainability

Supply chain sustainability is a business issue affecting an organisation's supply chain or logistics network and is frequently quantified by comparison with SECH ratings. SECH ratings are defined as social, ethical, cultural and health footprints. Consumers have become more aware of the environmental impact of their purchases and companies' SECH ratings and, along with nongovernmental

organisations ([NGO]s), are setting the agenda for transitions to organically-grown foods, anti-sweatshop labour codes and locally-produced goods that support independent and small businesses. Because supply chains frequently account for over 75% of a company's carbon footprint[6] many organisations are exploring how they can reduce this and thus improve their SECH rating.

Components of Supply Chain Management Integration

The management components of SCM

The SCM components are the third element of the four-square circulation framework. The level of integration and management of a business process link is a function of the number and level, ranging from low to high, of components added to the link (Ellram and Cooper, 1990; Houlihan, 1985).

Consequently, adding more management components or increasing the level of each component can increase the level of integration of the business process link. The literature on business process reengineering, [

7] buyer-supplier relationships,[8] and SCM[9] suggests various possible components that must receive managerial attention when managing supply relationships. Lambert and Cooper (2000) identified the following components which are:

- * Planning and control
- * Work structure
- * Organization structure
- * Product flow facility structure
- * Information flow facility structure
- * Management methods
- * Power and leadership structure
- * Risk and reward structure
- * Culture and attitude

However, a more careful examination of the existing literature[10] will lead us to a more comprehensive structure of what should be the key critical supply chain components, the "branches" of the previous identified supply chain business processes, that is, what kind of relationship the components may have that are related with suppliers and customers accordingly. Bowersox and Closs states that the emphasis on cooperation represents the synergism leading to the highest level of joint achievement (Bowersox and Closs, 1996). A primary level channel participant is a business that is willing to participate in the inventory ownership responsibility or assume other aspects of financial risk, thus including primary level components (Bowersox and Closs, 1996). A secondary level participant (specialized), is a business that participates in channel relationships by performing essential services for primary participants, thus including secondary level components, which are in support of primary participants. Third level channel participants and components that will support the primary level channel participants, and which are the fundamental branches of the secondary level components, may also be included.

Consequently, Lambert and Cooper's framework of supply chain components does not lead us to the conclusion about what are the primary or secondary (specialized) level supply chain components (see Bowersox and Closs, 1996, p.g. 93). That is, what supply chain components should be viewed as primary or secondary, how these components should be structured in order to have a more comprehensive supply chain structure, and to examine the supply chain as an integrative one (See above sections 2.1 and 3.1).

Reverse Supply Chain Reverse logistics is the process of planning, implementing and controlling the efficient, effective inbound flow and storage of secondary goods and related information opposite to the traditional supply chain direction for the purpose of recovering value or proper disposal. Reverse logistics is also referred to as "Aftermarket Customer Services". In other words, anytime money is taken from a company's Warranty Reserve or Service Logistics budget, that is a Reverse Logistics operation.

Software as a Service (SaaS, typically pronounced 'sass') is a model of software deployment whereby a provider licenses an application to customers for use as a service on demand. SaaS software vendors may host the application on their own web servers or download the application to the consumer device, disabling it after use or after the on-demand contract expires. The on-demand function may be handled internally to share licenses within a firm or by a third-party application service provider (ASP) sharing licenses between firms.

Aims

On-demand licensing and use alleviates the customer's burden of equipping a device with every conceivable application. It also reduces traditional End User License Agreement (EULA) software maintenance, ongoing operation patches, and patch support complexity in an organization. On-demand licensing enables software to become a variable expense, rather than a fixed cost at the time of purchase. It also enables licensing only the amount of software needed versus traditional licenses per device. SaaS also enables the buyer to share licenses across their organization and between organizations, to reduce the cost of acquiring EULAs for every device in their firm.

Using SaaS can also conceivably reduce the up-front expense of software purchases, through less costly, on-demand pricing from hosting service providers. SaaS lets software vendors control and limit use, prohibits copies and distribution, and facilitates the control of all derivative versions of their software. SaaS centralized control often allows the vendor or supplier to establish an ongoing revenue stream with multiple businesses and users without pre-loading software in each device in an organization.

The sharing of end-user licenses and on-demand use may also reduce investment in server hardware or the shift of server use to SaaS suppliers of applications file services.

History

The concept of "software as a service" started to circulate prior to 1999. In December 2000 Bennett et al. noted the term as "beginning to gain acceptance in the marketplace".

Whilst the phrase "software as a service" passed into common usage, the CamelCase acronym "SaaS"

was allegedly not coined until 2000-2001 in a white paper called "Strategic Backgrounder: Software as a Service" by the Software & Information Industry's eBusiness Division published in Feb. 2001, but written in the fall of 2000 according to internal Association records.

Philosophy

As a term, SaaS is generally associated by software professionals and business associates with business software and is typically thought of as a low-cost way for businesses to obtain rights to use software as needed versus licensing all devices with all applications. The on demand licensing enables the benefits of commercially licensed use without the associated complexity and potential high initial cost of equipping every device with the applications that are only used when needed.

Virtually all software fits the SaaS model well. Many Unix applications already have this functionality whereas EULA applications never had this flexibility before SaaS. A licensed copy of a word processor, for example, had to reside on the machine to create a document. The equipped program has no intrinsic value loaded on a computer that is turned off for the night. Worse yet, the same employee may need another fully paid license to write or edit a report at home on their own computer, while the work license is inoperable. Remote administration software attempts to resolve this issue through sharing CPU controls instead of licensing on demand. While promising, it requires leaving the licensed host computer on and it creates security issues from the remote accessing to run an application. SaaS achieves efficiencies by enabling the on demand licensing and management of the information and output, independent of the hardware location.

SaaS applications differ from earlier applications delivered over the Internet in that SaaS solutions were developed specifically to leverage web technologies such as the browser, thereby making them web-native. The data design and architecture of SaaS applications are specifically built with a 'multi-tenant' backend, thus enabling multiple customers or users to access a shared data model. This further differentiates SaaS from client/server or 'ASP' (Application Service Provider) solutions in that SaaS providers leverage enormous economies of scale in deployment, management, support and through the Software Development Lifecycle.

Key characteristics

Characteristics of SaaS software include:[dead link]

- * network-based access to, and management of, commercially available software
- * activities managed from central locations rather than at each customer's site, enabling customers to access applications remotely via the Web
- * application delivery typically closer to a one-to-many model (single instance, multi-tenant architecture) than to a one-to-one model, including architecture, pricing, partnering, and management characteristics
- * centralized feature updating, which obviates the need for end-users to download patches and upgrades.
- * frequent integration into a larger network of communicating software - either as part of a mashup or as a plugin to a platform as a service. (Service oriented architecture is naturally more complex than traditional models of software deployment.)

Providers of SaaS generally price applications on a per-user basis, sometimes with a relatively small minimum number of users and often with additional fees for extra bandwidth and storage. SaaS revenue streams to the vendor are therefore lower initially than traditional software license fees, but are also recurring, and therefore viewed as more predictable, much like maintenance fees for licensed software.

In addition to the characteristics mentioned above, SaaS software turns the tragedy of the commons on its head and frequently[weasel words] has these additional benefits:

- * More feature requests from users since there is frequently no marginal cost for requesting new features;
- * Faster releases of new features since the entire community of users benefits from new functionality; and
- * The embodiment of recognized best practices — since the community of users drives the

software publisher to support best practice.

Implementation

Historians can generally classify SaaS architectures as belonging to one of four "maturity levels", whose key attributes are configurability, multi-tenant efficiency, and scalability. Each level is distinguished from the previous one by the addition of one of those three attributes:

* Level 1 - Ad-Hoc/Custom: At the first level of maturity, each customer has its own customized version of the hosted application and runs its own instance of the application on the host's servers. Migrating a traditional non-networked or client-server application to this level of SaaS typically requires the least development effort and reduces operating costs by consolidating server hardware and administration.

* Level 2 - Configurable: The second maturity-level provides greater program flexibility through configurable metadata, so that many customers can use separate instances of the same application code. This allows the vendor to meet the different needs of each customer through detailed configuration options, while simplifying maintenance and updating of a common code base.

* Level 3 - Configurable, Multi-Tenant-Efficient: The third maturity level adds multi-tenancy to the second level, so that a single program instance serves all customers. This approach enables more efficient use of server resources without any apparent difference to the end user, but ultimately comes up against limits in scalability.

* Level 4 - Scalable, Configurable, Multi-Tenant-Efficient: The fourth and final SaaS maturity level adds scalability through a multitier architecture supporting a load-balanced farm of identical application instances, running on a variable number of servers. The provider can increase or decrease the system's capacity to match demand by adding or removing servers, without the need for any further alteration of applications software architecture.

SaaS architectures may also use virtualization, either in addition to multi-tenancy, or in place of it. One of the principal benefits of virtualization is that it can increase the system's capacity without additional programming. On the other hand, a considerable amount of programming may be required to construct a more efficient, multi-tenant application. Combining multi-tenancy and virtualization provides still greater flexibility to tune the system for optimal performance. In addition to full operating system-level virtualization, other virtualization techniques applied to SaaS include application virtualization and virtual appliances.

The development of SaaS applications may use various types of software components and frameworks. These tools can reduce the time-to-market and the cost of converting a traditional onpremise

software product or building and deploying a new SaaS solution. Examples include components for subscription management, grid computing software, web application frameworks, and complete SaaS platform products.

SaaS and SOA

Much like any other software, Software as a Service can also take advantage of Service Oriented Architecture to enable software applications to communicate with each other. Each software service can act as a service provider, exposing its functionality to other applications via public brokers, and can also act as a service requester, incorporating data and functionality from other services. Enterprise Resource Planning (ERP) Software providers leverage SOA in building their SaaS offerings; an example is SAP Business ByDesign from SAP AG.

Adoption Drivers

The traditional rationale for outsourcing of IT systems involves applying economies of scale to the operation of applications, such that a service provider can offer better, cheaper, more reliable applications than companies can themselves. The use of SaaS-based applications has grown dramatically, as reported by many of the analyst firms that cover the sector. But as

of 2009, SaaS has only truly flourished in recent years. Several important changes to the way people work have made this rapid acceptance possible:

- * Computers have become widespread: Most information workers have access to a computer and are familiar with conventions from mouse usage to web interfaces. As a result, the learning curve for new applications is lower and less hand-holding by internal IT is needed.

- * Computing itself has become a commodity: In the past, corporate mainframes were jealously guarded as strategic advantages. More recently, the applications were viewed as strategic. Today, people know it's the business processes and the data itself—customer records, workflows, and pricing information—that matters. Computing and application licenses are cost centers, and as such, they're suitable for cost reduction and outsourcing. The adoption of SaaS could also drive Internet-scale to become a commodity.

- * Insourcing IT systems requires expensive overhead including salaries, health care, liability and physical building space.

- * Applications have tended to standardize: with some notable, industry-specific exceptions, most people spend most of their time using standardized applications. An expense-reporting page, an applicant screening tool, a spreadsheet, or an e-mail system are all sufficiently ubiquitous and well understood that most users can switch from one system to another easily. This is evident from the number of web-based calendaring, spreadsheet, and e-mail systems that have emerged in recent years.

- * Parametric applications are usable: In older applications, one could often only change a workflow by modifying the code. But in more recent applications, particularly web-based ones, significantly new applications can be created from parameters and macros. This allows organizations to create many different kinds of business logic atop a common application platform. Many SaaS providers allow a wide range of customization within a basic set of functions.

- * A specialized software provider can target global markets: A company that made software for human resource management at boutique hotels might once have had a hard time finding enough of a market to sell its applications. But a hosted application can instantly reach the entire market, making specialization within a vertical market not only possible, but preferable. This in turn means that SaaS providers can often deliver products that meet their markets' needs more closely than traditional "shrinkwrap" vendors could.

- * Web systems show reliability: Despite sporadic outages and slow-downs, most people are willing to use the public Internet, the Hypertext Transfer Protocol and the TCP/IP stack to deliver business functions to end users.

- * Security is sufficiently well trusted and transparent: With the broad adoption of SSL, organizations have a way of reaching their applications without the complexity and burden of enduser configurations or VPNs.

- * Availability of enablement technology: According to IDC, organizations developing enablement technology that allow other vendors to quickly build SaaS applications will play an important role in driving the adoption of SaaS. Because of SaaS' relative infancy, many companies have either built enablement tools or platforms or are in the process of engineering enablement tools or platforms. A Saugatuck study shows that the industry will most likely converge to three or four enablers that will act as SaaS Integration Platforms (SIPs).

- * Bandwidth of wide-area networks has grown drastically following Moore's Law (more than 100% increase each 24 months) and is about to reach slow local networks bandwidths. Added to network quality of service improvement this has driven people and companies to trustfully access remote locations and applications with low latencies and acceptable speeds.

- * SaaS has the effect of democratizing software, allowing small and medium businesses to have access to functionality formerly the domain of large enterprises. Many analytical software tools have been released as SaaS applications and are available on a monthly subscription basis.

While the SaaS model has many advantages, it also has drawbacks. For example, data transfers take place at Internet, rather than local Ethernet speeds; the provider may go bankrupt and the firewall may not permit integration with back end systems. It may not be easy to judge the importance of

such issues when an implementation is first started, however they are largely resolved by the Hybrid SaaS model. Other concerns are detailed below.

Widespread implementation of SaaS requires well defined services. That can achieve an economy of scale and the capacity to balance supply and demand. This requires areas of IT that are ubiquitous and commodity-like. SaaS is therefore not suitable for innovative or highly specialized niche systems, though SaaS may be used to provide one or more components in such systems.

As with manufacturing, a lack of substitutability and second sourcing options with any commodity creates a strategic weakness for any customer in terms of security, competition and pricing. Various forms of this weakness, such as "vendor lock-in", are often cited[by whom?] as a barrier to adoption of SaaS as the current industry lacks portability and interoperability between vendors. This means that to change from one vendor to another will take a considerable amount of effort and time, although no more time than required to convert or migrate from one traditional, installed software package to another. This situation is resolvable by the introduction of open sourced standards and the development of markets based upon such standards.

Whilst the severe lack of substitutability remains unresolved, many vendors[by whom?] counter the concerns over potential security and operational risk with the argument that the professionals operating SaaS applications may have much better security and redundancy tools available to them. One vendor of SaaS document and process automation, DataServ, has for many years offered a "data-return guarantee" that allows clients to receive their documents and data upon cancellation of service.

Furthermore the concern that SaaS applications pose some difficulty for businesses that need extensive customization is countered[by whom?] with the claim that many vendors have made progress with both customization and publication of their programming interfaces. Note that customization will reduce substitutability and given that SaaS applications are sometimes deployed for non-strategic business activities, the strategic benefit of customization is somewhat questionable. In addition to this, the availability of open-source applications, inexpensive hardware and low-cost bandwidth combine to offer compelling economic reasons for businesses to operate their own software applications, particularly as open-source solutions have increased in quality and become easier to install. SaaS providers can offer a higher level of service and support than most open source solutions but the level of that service in any delivery model depends greatly on the orientation of the software vendor. For example, development-centric vendors who are highly technical tend to deliver the lowest level of user support whether in terms of technical support or implementation. Conversely, companies that are services-oriented tend to offer much more developed plans for technical support, user training, even supporting services such as data capture which make the application more usable. Users and purchasers of any SaaS application need to establish a strong confidence in the provider of the service, particularly if the application stores the user's data. This confidence can be enhanced and enforced by a balanced Service Agreement that gives the provider opportunities to correct issues but within limits that the client can accept. The provider needs to be trusted with both the intention and the ability to safeguard this information. Thus internet security procedures such as SSL or other encryption technologies should be required by all SaaS consumers.

Sales channels

With products below the \$100 range and its focus on the mid market, direct selling can become an expensive undertaking. SaaS companies seek alternatives by selling through value-added resellers (VARs), Managed Service Providers (MSPs), Master Managed Service Providers (MMSPs) and similar alliance partners. But since SaaS is not only a different delivery mechanism but a different business model and different technology as well, selling through channels has its own challenges.

Alternative pricing models

One reason for developing SaaS applications is the opportunity to implement alternative pricing models that focus on establishing and maintaining recurring revenue streams. Most SaaS vendors charge some kind of monthly "hosting" or "subscription" fee. Opportunities also exist to charge per

transaction, event, or other unit of value to the customer. These alternative pricing models come about because customers actually "lease" the software from the vendors and the vendors have the ability to view all transactional activity within the system.