

**Agility: The Key to Survival of the Fittest
in the Software Market**

by

Julie Smith David
Arizona State University
julie.smith.david@asu.edu

William E. McCarthy
Michigan State University
mccarth4@msu.edu

Brian S. Sommer
B2B Analysts
brian@b2banalysts.com

June 6, 2016

Text Word Count: 2,984

Agility: The Key to Survival of the Fittest in the Software Market

Introduction

The software industry and IT departments are facing extreme pressures to provide new applications that add value in today's competitive environment. Whereas in the 1990's companies concentrated on implementing systems that re-automated functions to provide specific benefits (the ability to process transactions on January 1, 2000), today's market demands new applications, and better integration within and between organizations. This has sparked the formation of many software companies to solve new problems, and well-established application providers are looking for new features and business models to improve their revenue streams.

Concurrently, there has been a significant increase in the number of small, interconnected organizations who are working to provide emerging services and products to today's customers. These firms have different information needs than large firms. They often provide a limited number of products or services so their information needs may be simpler. However, they often work in concert with many other firms to complete projects, so they need the ability to communicate seamlessly within this web of firms to share information about each project. In response to traditional and emerging markets, software vendors are realizing they must satisfy the needs of a wide range of companies and to develop applications tailored to each niche.

Facing these changes, those needing to evaluate applications are often faced with a daunting task of understanding key differences among software packages and attempting to identify the major players within each market segment. To facilitate such analysis, we have developed a framework that organizes software applications similarly to the evolutionary

categorization of animals. In this framework, analogous to the vertebrate/invertebrate classifications, enterprise systems can be distinguished as those that have been created without an organizing principle while others are either inwardly organized or outwardly organized. Figures 1 and 2 show the comparative evolutionary trees for animals and enterprise systems.

Each category of enterprise system can be sub-divided, and each species of system can be studied to identify (1) the characteristics that can lead to its continued existence and (2) the niches in which it can flourish. This structure can provide several benefits. First, those interested in purchasing software can use this categorization to identify the key differences among systems currently available. Second, software vendors and customers may use this framework to highlight today's software market trends with an eye toward guiding organizations toward tomorrow's software choices. Finally, we have successfully used this framework to design a graduate course to introduce students to a wide range of software applications and prepare them to enter the market in which agility is the key to survival.

II. Developing the Framework: The Evolution Analogy

The ideas of Darwinian evolution can be used to describe the tumultuous IT environment in which enterprises compete. As illustrated in Figure 1 for animals, a first pass at natural classification produces the three categories of (a) invertebrates, (b) vertebrates, and (c) a speculative category of how animals interact as communities. In Figure 2 a similar first pass categorizes enterprise systems into (a) systems with no overall organizing rationale (i.e., with no backbone), (b) systems with inward organization (i.e., can be likened to vertebrates), and (c) systems with outward organization (i.e., able to communicate and interact as communities). Each of three information system categories is explained below and can be further subdivided, as shown in Figure 2.

Insert Figures 1 and 2 here

Systems with no organizing rationale: Invertebrate animals such as insects are not as advanced biologically as their vertebrate counterparts, but they survive well in a multitude of environments. Similarly, single entry enterprise systems can be successful in guiding a multitude of small organizations. However, these systems are not able to provide robust classification principles to guide the recognition of transactions. *QUICKEN*™ is an excellent example of such a system that has found a niche and is flourishing within that narrow segment— individual users and very small businesses in which the owner is the key participant and decision maker. These systems work well when the owner or manager participates in all of the key business events. However, they face survival problems if the organization has significant transaction volumes, reporting requirements, or outside information users. For example, it is difficult to provide a substantial amount of non-financial information using of these systems.

Systems with inward organization: More advanced enterprise systems incorporate an organizing principle to bolster their organizing and processing capabilities. These systems adhere in some fashion to one of two major organizing principles: (a) the classic double-entry accounting equation of **assets = liabilities + owners equity** ($A=L+OE$) from Pacioli's 1494 treatise [4], and (b) the **enterprise value chain** concept of Porter [8]. These two divisions and their hybrids are explained below:

The first organizing principle, $A=L+OE$, has successfully guided enterprise information needs for over 500 years. With these systems, information continues to be focused on the financial implications of economic events, much as it was with single entry systems. However, users now are provided with a framework that can help to insure information completeness and to enforce rules about how and when transactions should be recognized.

With advancing technology, managers recognized several weaknesses with systems that summarize data to fit the categorization of general ledgers and charts of accounts. Therefore, other functional areas within organizations independently developed software to meet their needs. For example, materials requirements planning (MRP) systems were created to better support manufacturing processes and to assist with production schedules and materials lead-time requirements. Additionally, activity-based costing (ABC) systems attempt to identify the activities that result in cost expenditures, rather than relying allocations based on direct labor and materials usage. These MRP and ABC systems are examples of “hybrids” which have evolved beyond traditional $A=L+OE$ systems, but many products in this category retain some fundamental bookkeeping flavor.

Rather than focusing on financial transactions, more recently-created systems are better understood using Porter’s enterprise-wide value chains concepts as their foundation. Their goal is to capture a wide range of information about all key business events. These systems recognize that customer demands pull resources into the organization to be consumed in pursuit of providing customer value. While these enterprise resource planning (ERP) systems are able to generate financial statements and to work in an integrated fashion with general ledgers, these are no longer their preemptory goals.

When implemented correctly, ERP systems provide firms with many advantages. They help standardize procedures across divisions and nations, they are able to consolidate detailed transaction data from different functions, and they provide methods to access data throughout the entire range of organizational activities. Thus, by reducing internal processing costs and enhancing communication within the organization, these systems have facilitated the growth of large, multi-national firms. However, while providing these benefits, they also impose many

constraints on adopting organizations. Many of these packages are inflexible, and firms implementing them often must adapt their business rules to meet software specifications rather than the other way around. As a result, some organizations are struggling to implement best of breed solutions. Although this solution may provide better a functional fit, integration is very difficult and costly.

Systems with outward organization: Outwardly organized software systems, such as customer relationship management (CRM) and supply chain execution (SCE) systems, support not only a single enterprise's set of business processes, but work among organizations to provide data to other participants in predictable formats. For example, e-Procurement software can enhance interorganizational communication by focusing on a standard set of document definitions (such as those based on EDI or XML) that allows uniform integration of data elements.

Future systems solutions may go further by utilizing common and independently-viewed definitions. For example, with a system that captures shipping information, the originating company could use the data for sales calculations, while the destination company could use the same data as their receiving records. These integration systems provide trade facilitation advantages because they increase efficiency with consistent and non-redundant storage. However, the standards for supporting this type of processing are in their infancy. Some firms (such as those in countries that dictate adherence to account standardization) may choose to standardize their interorganizational data transfers at the bookkeeping level (that is, use $A=L+OE$ as the transfer protocol) with the result that the common data will be reduced to account entries. However, we believe it is much more likely that these common systems will evolve from ones that capture more detailed semantics about business resources and events such as envisioned in

the Semantic Web [2]. These higher order systems will allow users to store, access, and format their information in a manner suited to their own goals. Additionally, these systems may enable new organizational structures. By reducing the cost of interorganizational coordination, small, nimble firms that focus on one activity in the value chain can flourish, relying on communication capabilities with other firms that provide complementary services.

III. Using the Framework

This framework has several uses, and perhaps the most obvious one is to explain the range of software available. Having shared this with a wide range of international business people and students, we have seen that it can minimize the complexity of today's market (especially for software non-experts). However, we believe that it also has value to those who purchase software and those in the software industry. The following sections share several insights that become apparent when you extend the framework by applying other evolutionary concepts. Similar to the organization of individual animal species, we grouped several of today's software packages into categories that would streamline a software selection project. Second, we describe how software categories are adapting to better survive in nearby niches. Finally, we identify recent mutations that provide insights into possible business models that may influence tomorrow's software market.

A. Species Classification: “the systematic categorization of organisms. One aim of modern classification, or systematics, is to show the evolutionary relationships among organisms.” [3]

One of the main ways that evolutionary structures have been helpful in biology is to categorize individual species, helping to understand their similarities and differences. Similarly, organizations that need to purchase software are often overwhelmed with the range of application solutions and amount of information available about them. If corporate managers understand the

enterprise evolution structure, they can quickly narrow their search to a category that supports their business needs. Thus, an important first use of our framework is categorizing software applications.

B. Adaptation: "the adjustment of living matter to environmental conditions, including other living things." [3]

In the animal world, species able to adapt to different environments survive. This concept can be applied to the software market to see that flexible software vendors are able to successfully adapt their applications to a range of market niches are more likely to flourish in today's economy. With this in mind, one obvious route to increased revenues is to identify market niches that are closely related to their core customer group and to modify their existing system to move into this new market. In terms of the software hierarchy, this means that vendors may attempt to expand to their left or right, especially within an organizational family. For example, within the Inwardly Organized categories, we see several of the smaller ERP systems adapting to move toward the larger markets (move toward the right on the evolutionary tree), such as JD Edwards' successful move from the Hybrid category to become a Single-Source ERP vendor. Additionally, the larger systems have nearly saturated the Fortune 500 market, so they have begun to streamline their offerings and make them available to smaller firms. For example, SAP states that their product, R/3, can work in companies of "all sizes." While vendors have these aggressive goals, we believe that moving slowly and methodically to nearby categories can prove more successful than attempting to simultaneously adapt to every environment.

Vendors are also adapting their applications by reaching across organizational categories to integrate new functionalities into their products. For example, advanced planning and scheduling (APS) systems and CRM systems have moved beyond novel, stand-alone applications

to become important in today's market. Because they provide significant value to potentially any organization, the market is attempting to determine the best way to integrate these features into other systems, and several approaches have emerged. Consider the ERP market. Some vendors have entered into partnerships and then to built seamless integration between their systems. For example, J.D. Edwards initially partnered with Siebel Systems to provide CRM functionality. Another approach to adding functionality is for firms to purchase stand-alone products to more tightly integrate into their systems, such as the J.D. Edwards acquisition of Numetrix (an advanced planning system) in 1999 and YOUcentric (a CRM system) in 2001. Finally, some vendors have chosen to create their own application modules. For example, SAP has released its own supply chain modules to be sold with its R/3 product. The advantage of this approach is that integration concerns are eliminated as the products have been designed to work together. The difficulty, however, is that stand alone vendors often have a long history of successful implementations and years of refining their systems. Therefore, such ERP vendors have largely (and, perhaps unsuccessfully) found themselves in a "catch up" mode, trying to develop a system that matches the functionality of the older products [6].

C. Mutation: "a sudden change in a gene, or unit of hereditary material, that results in a new inheritable characteristic." [3]

The software market is also being changed as a result of mutations introduced by new vendors. For example, an important mutation is radically different business models for the software industry (see Figure 3). In the traditional model, companies buy both software and hardware, accepting responsibilities for ownership and operation of these systems. Mutations to this basic model have occurred recently, and their impact on the software market may be dramatic.

Insert Figure 3 here

As the economy has become more competitive, many companies have recognized that systems operations is not their core competency, and that there are significant inefficiencies in maintaining internal IT departments. In response, a new threat to software vendors has arisen: Application Service Providers (ASP's) who host an organization's systems on centralized hardware and, in many cases, also process the organization's transactions. Additionally, the ASP's may be responsible for integrating modules and customizing code if needed. This poses two threats to software vendors. First, if ASP's are able to consolidate processing resources more efficiently than individual companies, then software sales will shrink. Additionally, if the ASP's provide value-added services such as consulting and customization, the software vendors will lose these additional revenues. The impact of these threats can be significant. For example, GM outsourced all of its human resources processing to Arthur Andersen for a 10-year contract of \$250 million [7]. In response to this change in the software market, hardware vendors, software vendors, consulting firms, and public accounting firms are each vying to become prominent ASP's [5]. The key to their success will likely be their ability to offer very high levels of reliable service and to efficiently integrate separate software products.

The Internet is further confounding this phenomenon. Whereas the ASP's have largely hosted client-server applications, *portal* sites challenge our traditional notions of how software should be delivered. By creating web-based software and operating it through their portals, software vendors are operating similarly to ASP's. However, they are also able to provide additional services at the site such as coordinating relationships among related sites and organizations. Taken to the extreme, this model would result in componentized web-based

applications for all functions, and a community of users who work together, streamlining the supply chain.

Portal providers can earn revenues from individual users, from advertising, and from transactions that occur between its community members. Simultaneously, this mutation offers software customers several key advantages. First, the advantages that ASP's provide to their customers would also apply with this business model. Additionally, software implementation costs are likely to be reduced further because the software runs over the web, and any employee who is familiar with browser software is likely to need minimal additional training. Finally, by providing firms with the ability to interact seamlessly with their customers and suppliers, overall efficiencies should be improved, thus helping all organizations in the supply chain remain competitive.

These types of environments are new, and there is still considerable uncertainty about how they will evolve. For example, similar to the *Single Source ERP* versus the *Best of Breed* tension in the ERP market, we are curious to see if software-providing portals will develop complete information solutions, or whether they will focus on core competencies. If the latter occurs, the need for inter-portal communication standards will be critical to the success of these niches.

Portal providers could also develop radically different data structures for their systems -- as different as the break between systems with no organizing rationale and those that are inwardly organized. If the portals focus on streamlining interorganizational communication, they will need to capture detailed information about each customer-supplier interaction, similar to today's CRM products. The difference, however, will be that the data could be stored once and then made available to both parties in the transaction. In this environment, the records could be

used by the customers to perform vendor analysis and supply chain execution, while the suppliers could use the records to perform customer analysis and demand planning.

IV. Conclusions

We have developed an evolutionary framework to categorize today's enterprise software market. The value of this framework lies in its ability to explain today's market and to help identify potential evolutionary changes. Today's market is complex, and the changes business firms face are dramatic. Our framework highlights that any application's ability to survive in such an environment relies on its producer's ability to remain flexible. They must be able to identify important market mutations that challenge the status quo and to adapt to those changes with either incremental or wholesale changes in product components.

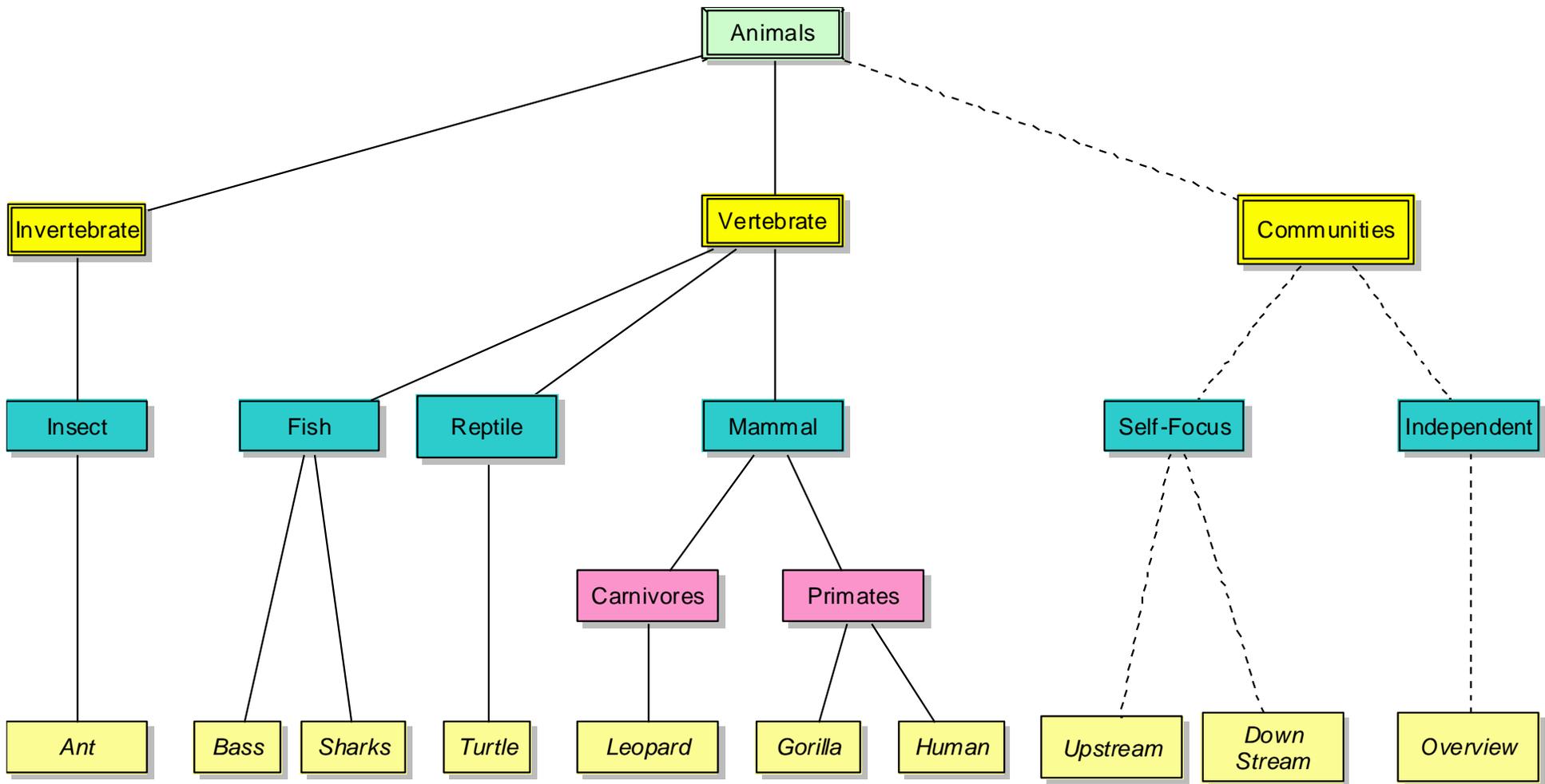


Figure 1: Evolution of Animals

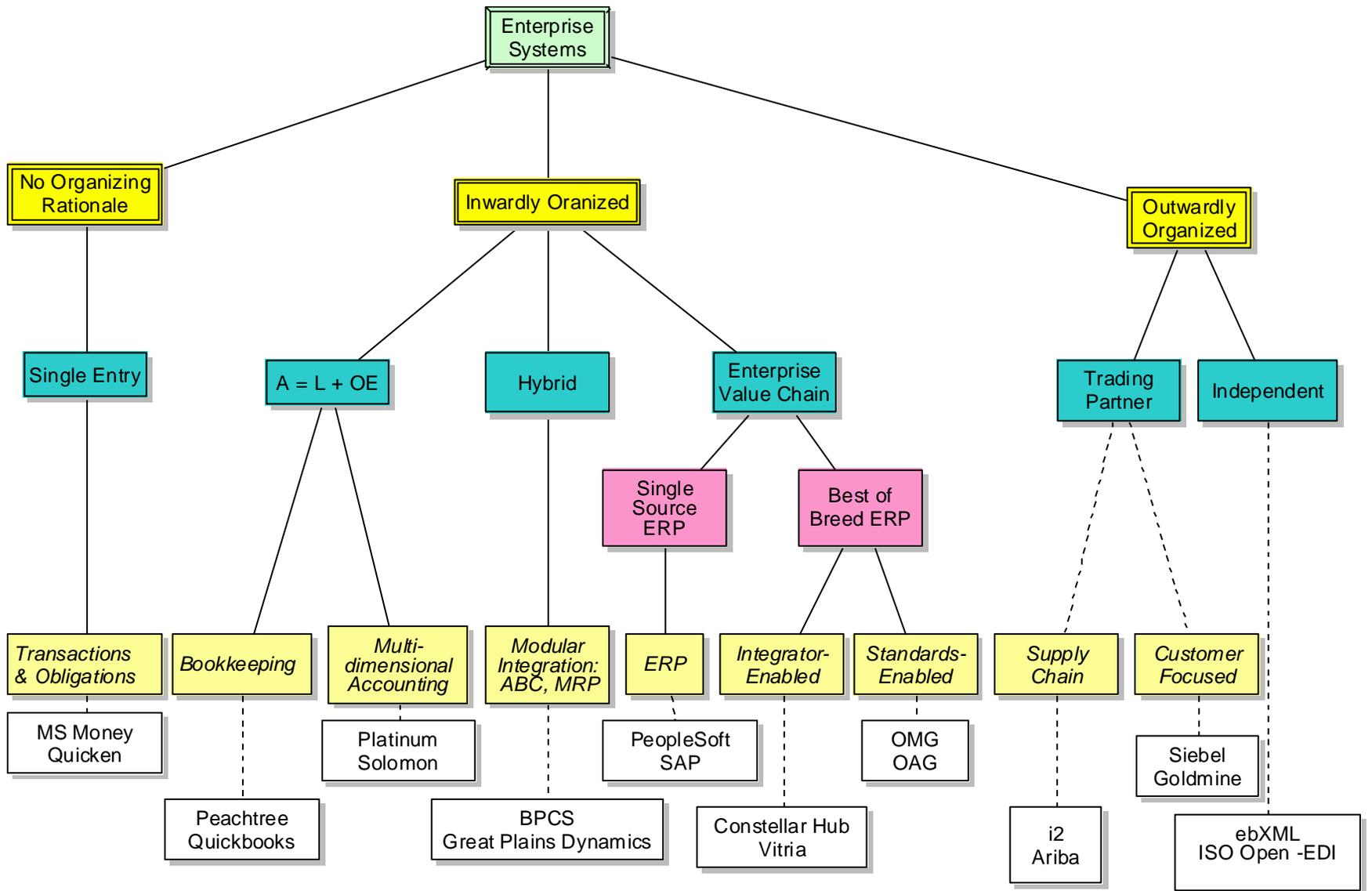
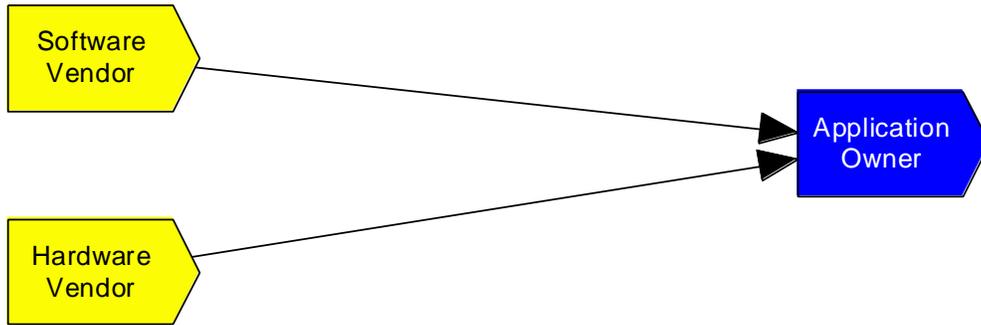
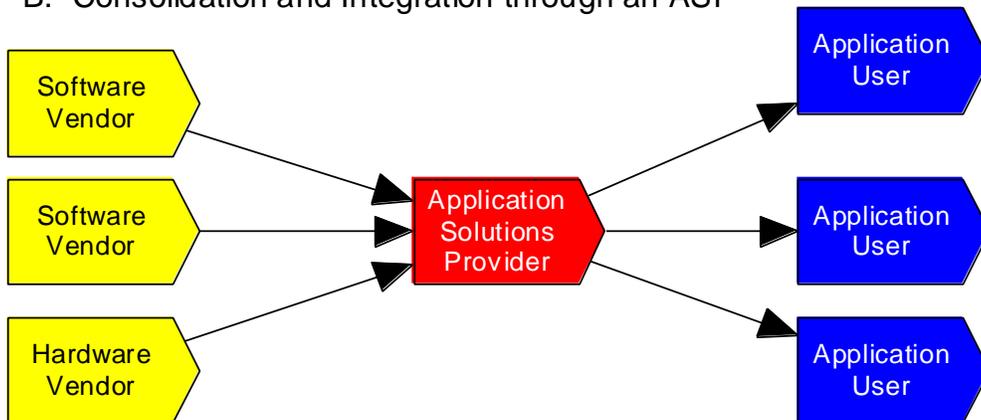


Figure 2: The Evolution of Enterprise systems

A. Initial Business Model



B. Consolidation and Integration through an ASP



C. Technical and Organizational Integration through a Portal

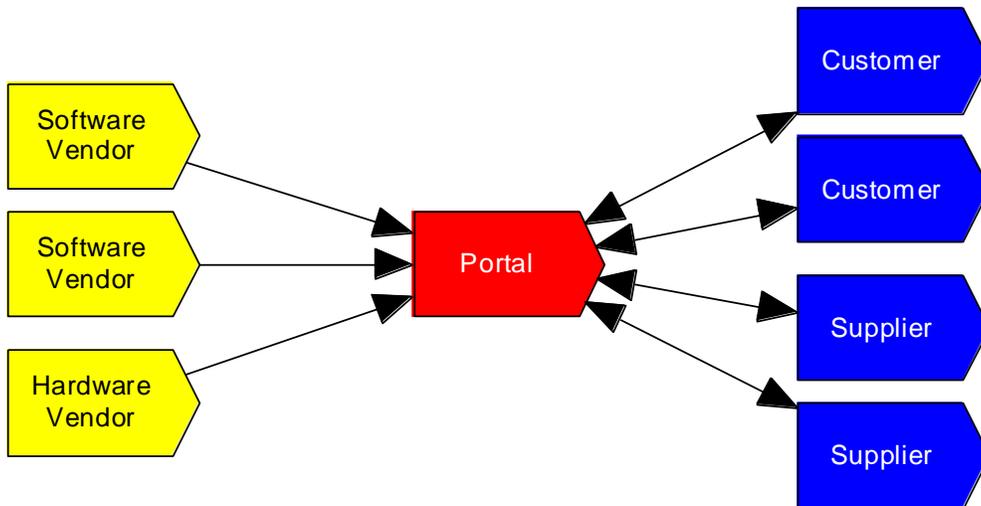


Figure 3: Business Models for Application Solutions

References

1. Altman, R. Enterprise Application Chaos.
http://messageq.com/executive_corner/altman2.html. April 6, 2000.
2. Berners-Lee, T., J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*. May 2001.
3. Encyclopedia.com, www.encyclopedia.com.
4. Geijsbeek, John B. Ancient Double-Entry Bookkeeping: Luca Pacioli's Treatise (A.D. 1494 - the earliest known writer on bookkeeping) reproduced and translated with reproductions, notes, and abstracts from Manzoni, Pietra, Mainardi, Ympyn, Stevin, and Dafforne, Scholars Book Co., 1974, c1914.
5. Gill, P.J. ERP vendors stake claim to application outsourcing market. *Datamation*. September (1999.).
6. Marion, L. The limitations of extended ERP. *Datamation*. November (1998).
7. Ostrander, J. Arthur Andersen Signs \$250 Million Agreement to Provide Financial Services to General Motors, Acquires Related GM Assets.
<http://www.arthurandersen.com/WebSite.nsf/content/MediaCenterNewsDesk17?opendocument>. (1999)
8. Porter, M.E. *Competitive Advantage*, The Free Press, New York NY, 1985.

Biographies

Julie Smith David (julie.smith.david@asu.edu) is an associate professor in the School of Accountancy and Information Management at Arizona State University in Tempe, AZ.

William E. McCarthy (mccarth4@msu.edu) is a professor in the Department of Accounting and Information Systems at Michigan State University in East Lansing, MI. He is active on several e-commerce standardization efforts including ebXML, eBTWG, ISO Open-EDI, and XBRL.

Brian Sommer is a partner with B2B Analysts, an innovation and value realization research firm based in Cambridge, Massachusetts. Previously, Mr. Sommer was CEO/Chairman of IQ4Hire and a partner at Andersen Consulting. He can be reached at: brian@b2banalysts.com.