



# The New Garage

*The coming business model for the  
enterprise software industry*

Executive White Paper

*June 2004*

# The New Garage

*The coming business model for the enterprise software industry*

## **Overview**

The garage occupies a special place in the history of American innovation. The free rent and ample space of the family garage have launched many a Walter Mitty dreamer into the upper echelons of high-tech entrepreneurs where some have garnered fame and riches beyond their wildest dreams. Garages have served as the first headquarters and development labs of recent college grads, out of work engineers, and just plain tinkerers who had ample time and little money to try out their ideas. The garage has also been mythologized beyond what is real and technology innovations that have also occurred in cheap office space everywhere. Suffice it to say that the garage simply stands for a place where a can do attitude and management made frugal by necessity first met with success in technology innovation.

Although taking an idea from conception to market is less expensive today than it has ever been, it is still not free. But imagine if the requirements for launching a software company could be reduced significantly at the inception stage — perhaps many more good ideas would come to market which could spark a new era of innovation with numerous positive productivity and economic effects.

Changes in the way we build and deploy software are affecting the way the software industry innovates and delivers its products. Two recent innovations in particular hold the promise of changing the software industry beyond recognition. Specifically, application hosting and Web services have changed the way CRM and other enterprise software can be made and delivered to users. This combination of new forces in the software industry have come about in part because the traditional means of building and delivering software functionality have become too expensive and too fraught with risk for many enterprises to cope with.

Enterprise software has always been a costly and risky proposition but the costs had always been absorbed by companies eager to get the latest generation of technology that promised to help them leapfrog the competition. But with enterprise resource planning (ERP), supply chain management (SCM), and customer relationship management (CRM) systems the industry reached a tipping point and the need for a new, less expensive method of delivering software had to be found.

Application hosting has driven down the cost of software acquisition, maintenance, and training and, at the same time, it has driven up user acceptance and streamlined delivery. Now the same market dynamics are appearing in the software development world and they promise to lower the costs associated with development. In effect, these developments are returning the industry to a point where smart people with little more than a few PC's and a garage can produce innovative software solutions for a market that is hungry for new ideas.

Beagle Research suggests that a new enterprise software industry business model is emerging that will “deverticalize” the industry. The changes wrought by this about face will enable new entities to serve customers better and will have far reaching effects not only on how software is delivered to the customer but also on how it is made, who makes it, and the financial underpinnings of the industry itself. This Beagle Research Group *Executive White Paper* examines these changes and forecasts how they could have as profound an effect on software innovation as, well, the garage.

### **The New Enterprise Software Business Model**

For more than fifty years technology and business have formed a symbiotic relationship that has blurred the distinction over which one is the primary change agent. Innovative technologies have made new business processes possible and the demands of business on technology innovators have closed the loop. But throughout the fifty-plus year history of the software industry, the business of technology has been rather stable even while other industries have seen major paradigm shifts.

Over the last three decades, many other industries have witnessed a paradigm shift caused by the split between the actual producer and the promoter or distributor of products. Manufacturing is a particularly good case-in-point and there are numerous examples of companies that design everything from car parts to sneakers in North America, for example, and build them in other countries. But the examples extend well beyond sending manufacturing to lower cost markets. In many cases high tech communications gear is designed by one company and farmed out to secondary manufacturers in first world countries and Hollywood studios have largely become promoters and distributors of movies made by independent film makers or “indies”. And in the intellectual property arena the recent up tick in outsourcing has resulted in a migration of work in such diverse areas as diagnostic radiology and legal services to professionals in other countries.

Throughout this transformation in the rest of the economy, the software industry has remained vertically integrated with each vendor developing, marketing, and selling its own versions of popular products. This situation has enabled the development of a thriving systems integration industry whose job is to knit together disparate systems from multiple vendors so

that enterprise customers could achieve a semblance of end-to-end business process automation.

Unfortunately, as demand has caused vendors to produce software applications that increasingly replace human decision making with best practices driven by analytics and business rules, the software has become more intricate and brittle. Any enterprise software deployment today should start with an assessment of current business practices and a comparison of internal practices with those offered by the software vendor. But this process is labor intensive and time consuming and rather than occurring prior to implementation, about half of the time, it is performed in line with implementation and it is a significant contributor to late delivery and cost overruns.

The solution to these and other problems is to develop pervasive standards that dictate how modularized applications work together rather than simply having each vendor develop modular architectures that work well internally but that do not necessarily integrate well or easily with the rest of the world. Taken to a logical end point, once a pervasive integration strategy is achieved, software houses will have every incentive to deverticalize and form new associations in which some perform more of the traditional tasks of marketing, sales, and support while others concentrate on applying their domain expertise to end user needs by developing and servicing applications that address new business requirements.

#### ***Deverticalizing the Enterprise Software Industry***

As with other industries that have deverticalized, the software industry will transition from one in which vertically integrated software companies perform all phases of development, sales and marketing to one made of a variety of specialists. The primary kinds of specialists will be publishers and developers. Some of the major milestones are listed below.

1. The traditional enterprise software company business model will split in half yielding publishing companies and development companies with each new company type taking on new or expanded roles in delivering products to the market.
2. The software publisher will emerge as an important new driving force in the industry. The new software publisher will combine the functions of sales and marketing, application hosting, venture capital, and technology and billing infrastructure for software developers. Most importantly, however, the publisher will become the primary aggregator of requirements from the market designing the solutions that need to be built to satisfy customer demand.
3. This new arrangement will drive cost reductions and improve reliability for end users, and it will provide better integration capabilities than ever before. The software publisher will

offer a federated suite of applications from multiple sources that end users can use in a best-of-breed mode. The applications will be based on the publisher's standards, tool suite, data model, and Web services that all developers within the publisher's portfolio use to craft their products. Finally, the publisher will be responsible to the customer for all aspects of quality assurance at the sub-assembly and final assembly levels.

4. The publisher will also provide hosting services for end users or subscribers. The software publisher will be the interface between the customer and the developer aggregating demand and collecting subscription fees from customers and paying royalties to developers. Some publishers may also take on role currently filled by venture capitalists to provide funding and other support to emerging software developers who can add functionality in the form of new modules or whole new applications to the publisher's portfolio.
5. The second new class of enterprise software company, the software developer, will retain most of the research, development, and domain expertise capabilities traditionally associated with software companies. However, this new entity will have the option of outsourcing its sales, marketing, and platform support issues — and the costs and overhead associated with these endeavors — to a software publisher. The software development company will be better able to focus its energies and talents on serving customers. Post implementation services based on the developer's applications will become a larger part of both the company's activities and its cash flow.

### **Taking a Page Out of the Linux Book**

The model proposed here is deliberately hosted to provide end users, developers, and publishers with specific benefits that include lower costs, faster time to market, greater reliability, and greatly reduced or eliminated integration effort. Multiple applications built using the same data model, the same tools, and development environment would effectively be a single application. This approach could yield a level of sophistication, modularity, and reliability that has never been approached in the current system of competing companies with relatively small development groups working to build similar applications. The only close comparison is the way the Linux operating system was built.

One of the driving forces behind the development of Linux was the sheer size and complexity of modern operating systems. Operating systems are large and complex and difficult for a small group of developers to build and maintain and errors in their code can cause problems for literally millions of people. The solution to operating system complexity was to

open up the source code to qualified developers and to provide an infrastructure that enabled quality control on all new submissions or changes. The result was to enhance and improve the operating system to the point that today Linux is one of the most stable and error-free (and inexpensive) operating systems on the market.

Taking an identical approach to Linux with enterprise software will not work for several reasons that are mostly profit related, but a business model that is a close equivalent of the Linux approach and that is based on the profit motive could work. A publisher's installed base would be the primary attraction of developers who could enhance existing modules and build additional applications. A large installed base means a large captive market for new or improved applications. Leading publishers will be able to recruit larger groups of developers to work on a business problem than can be sustained under the current model because the costs of software development would be shared by entrepreneurial developers.

The division of responsibilities between author and publisher in the book publishing industry makes a good analogy. Publishers play a key role in deciding what subject matter and which authors reach the market, presumably because they track and understand demand patterns and because they understand literature. Authors who wish to access a publisher's distribution channel must conform to publisher standards in their work. Book publishers take on all responsibility for marketing, sales, production, warehousing and other support activities. In turn publishers pay royalties to writers based on book sales. Individual authors contract with publishers to sell their books and, although it is difficult to have a first book published, once a writer is established with a publisher, he or she can have access to the publisher's distribution channel and considerable market presence. At the same time, the writer is free to engage in other related market activities including other writing and public speaking based on the writer's domain expertise.

### **Missing Pieces**

Key missing components that still need to be worked out before the model becomes a reality include infrastructures for both business and technology. Several years ago when application hosting came onto the market, vendors struggled with defining the business model and developing contractual arrangements that defined their services and obligations to customers. At the inception of the hosted CRM market, we counted 13 different business models which have been reduced to one by market forces. A similar scenario could evolve for the publisher-developer model but it may not provide for such an extreme number of varieties.

### **Technology Issues**

Hard technical issues still need to be worked out that will enable large scale federated applications to function as one. Also, publishers will need to develop best practices for validating new applications and new versions of applications within their portfolios. Some method of grading applications

according to security concerns and the business practices they support is needed across the industry. For example, a new application supporting a heretofore unrecognized business process might be graded differently than a standardized accounting package.

### ***Legal Issues***

Legal issues could be significant as the industry grapples with issues of ownership, exclusivity, copyrights, revenue splits, and intellectual property. And no doubt there will be missteps and legal challenges along the way, especially as this model globalizes. Nevertheless, business issues are largely issues concerning people and agreements and there is ample talent in the industry to fathom the intricacies and nuances.

### **Concrete Examples**

#### ***Salesforce.com and Sforce***

Because it is a disruptive innovation, there are no examples of a fully articulated publisher-developer model operating today but *salesforce.com* (SFDC) comes close with its *Sforce* offering. SFDC was one of the first companies to initiate a disruptive innovation in the enterprise software industry by offering CRM as a hosted service and it is now building what it calls a hub-and-spoke model of a developer community — the second disruptive innovation in enterprise software.

SFDC's model uses its core application, storage, and hosting technologies as the hub and it invites developer companies to participate as spokes by offering free access to development tools, an open API with documentation, and all the trappings of a developer community to promote interaction and cross pollination of ideas. Currently SFDC says it has more than 1,000 developers involved in the program representing about 100 discrete companies. The development environment, *Sforce*, is hosted using the same facilities as the CRM application and it is currently in release 2.0. SFDC plans a steady stream of announcements over 2004 as it enhances *Sforce* and builds out its community.

SFDC recognized the limitations of the hosted model in the areas of customization and integration early on and *Sforce* was initially a response to those needs. By opening up its API and offering a set of standard tools, developers could begin to add features to their companies' instances of the CRM application and integrate that application with in-house applications. *Sforce* has enabled corporate developers to integrate existing applications like financial systems to provide the front office staff with a more rounded view of the customer from within the CRM application.

*Sforce* lacks some of the basic infrastructure that would make it the prototype of a fully articulated publisher-developer model, such as single sign-on capability and any processes for validating technology or business infrastructure for paying royalties; but overall the program has enough of the pieces to clearly show the broad outlines of the future. For example, *Sforce* has been positioned from the beginning to provide three key

capabilities to SFDC customers: extension, enhancement, and new development within the SFDC architecture.

The 1,000 developers that SFDC says are using *Sforce* are not corporate developers working on enhancements; they are a group of independent developers building complementary solutions to the core CRM applications. Some of these developers have already brought their products to market and their solutions range from small dynamic linked libraries (DLLs) to whole application modules.

By opening up its application and development environment, SFDC has given itself a huge advantage over competitors. As a small company that is now self funding, SFDC competes with much larger rivals that have significantly greater resources to spend on enhancing their suites. SFDC's trump card is the more than 130,000 deployed users of its application. Unlike any other enterprise software company, all of SFDC's customers are synchronized on a single version of the application giving the company an attractive target to put before the developer community. The company has leveraged this large customer base to attract 1,000 developers who are working for future profits to build out SFDC's portfolio of relevant and related applications to its core product. This leverage has enabled SFDC to compete head-to-head with its largest rivals to develop and deliver to market advanced customer facing applications.

### ***Siebel, UAN, Integration OnDemand***

Siebel Systems is another company that has been working the issue. Two years ago the company brought to market the Universal Application Network (UAN), a standards based integration utility based on emerging standards for exchanging process information among applications and integration servers.

A key difference between UAN and Sforce is that the Sforce product appears to support integration at the data level whereas UAN supports integration at the process level. While a data centric integration approach will provide for most of the needs of small to medium businesses, the integrated processes of larger enterprises might be better served by a more process centric approach such as UAN. Regardless each product offers a glimpse into the future and should be applauded.

Recently, Siebel made UAN available as part of the Siebel-IBM CRM OnDemand suite in an offering called Integration OnDemand. It appears Siebel's short term interest is in supporting its enterprise customers who may need to integrate legacy applications. On the contrary, SFDC is actively courting third party developers who can enhance or add to its core functionality.

## **Reshaping the Enterprise Software World**

Responsibilities for various aspects of software development, sales, and marketing will obviously change in the new model and there will be

winners and losers. But as with most disruptive innovations, there is plenty of time before the new model becomes dominant for forward-looking companies to insulate themselves from the worst prospects and take advantage of what should be new opportunities. There are multiple constituencies in enterprise software and each will be affected in different ways. This section examines how some of them might fare.

### **Software Publishers**

The new business model is decentralized, which will cause the typical traditional software company to devolve its sales, marketing, packaging, and some support duties to a publishing company. Revenue sharing would closely resemble that in the publishing industry, or perhaps the deregulated utility industry, in which the publisher distributes portions of the revenue based on prior agreements. In the new model, the publisher would split revenues with the developer and possibly other entities – for example a hosting service if that function is outsourced – retaining a portion as earnings from hosting, sales, and marketing activities.

The new model offers significant opportunities for improved cash flow to the publisher through economies of scale for such activities as sales and marketing. However, with a growing portfolio of software titles in its inventory, the publisher as sales agent to an increasingly captive audience will need to invest more in knowing a customer's business requirements than many of today's software companies have the time and resources to do. Today's software vendors, with their relatively limited number of modules, must sell what they have regardless of fit and hope to customize their applications to meet specific demands. In a publishing model, customization will be a less onerous process because there will be more developers focused on delivering exact fit solutions and because a publisher will have access to more varied solutions. Finally, a bandwagon effect will be generated. As additional integrated software titles become available from a publisher, more developers will wish to become associated with that host's platform and in order to sell to that channel.

### **Software Developers**

Software developers — companies that elect to focus on application development — stand to benefit substantially from this new model. Obvious benefits, especially for new companies, include eliminating the need for some upfront investment in hardware, databases, and development tools all of which would be part of the infrastructure provided by the publisher. But any niche software company that has ever tried to partner with a large enterprise software company will find value in this new model. Today partnerships evolve out of negotiations between companies with existing products and frequently most of the negotiations involve integration issues. In the new model, integration is dealt with at the beginning of the relationship before there are applications and the same is true for go to market strategies, billing, splits, and other relationship issues.

Under the new model developers can free themselves from the overhead of product packaging, marketing, sales, and hosting leaving them better able

to concentrate on research and development as well as deepening their domain expertise. And because these companies focus more on their domain expertise they will be better positioned to provide consulting services to customers that need it. Under this new framework the software developer will evolve further into a business process/services expert and it will be these additional services that fuel growth as the developer sells directly to customers subscribing to application services.

Finally, this new model will leverage Web services significantly and many software developers will see benefit in making their applications available through multiple publishers unless competition or contractual obligations prevent it. Nevertheless, supporting multiple publishers via Web services will be easier than supporting multiple operating systems as, for example, in the days of the mini-computer.

### **End Consumers**

Beagle research consistently finds that issues of customization, integration, and security are the highest ranking concerns among consumers considering hosted CRM<sup>1</sup>. Early hosted solutions featured a single application with relatively narrow ability to modify application behavior beyond the configuration options provided to system administrators, in addition, security has been a perennial issue for those considering a hosted CRM deployment. There has been significant progress on all fronts but especially in security as hosting vendors have implemented procedures and technologies to provide safeguards that meet or exceed anything found in a corporate IT setting.

Customization and integration issues are being addressed today and the same capabilities that are enabling the publisher-developer split are responsible for making hosted CRM applications better able to integrate with legacy applications as well as enabling greater customization options. The trend toward more openness and flexibility is likely to accelerate and the functionality line between hosted and licensed applications should continue to blur as the new model takes shape giving end users more choices in how they use applications to accomplish their business objectives.

### **New Applications**

Perhaps the greatest benefit to end users will come in the form of applications that have not been invented yet. There is ample evidence in the market of a new class of hosted applications, called “Web-necessary” (WN) that leverage the Internet as part of their value proposition rather than simply using it as the delivery mechanism<sup>2</sup>. WN applications are frequently used to capture the voice of the customer (VOC) in collaborative settings which may include individuals from the vendor and customer as well as third parties providing services to either or both. Capturing more

---

<sup>1</sup> Hosted CRM's Continuing Evolution, Changing Leadership?, Beagle Research Group, March 2003

<sup>2</sup> See *CRM's Necessary Divergence*, a Beagle Research white paper for a full discussion of Web-necessary applications.

customer input will enable vendors to gain better insights into trends that can affect product development and will enhance closed loop marketing efforts.

### **Challenges for Existing Enterprise Software Vendors**

The new software business model should not be seen as a threat to the long term viability of today's enterprise application software companies, but it will require them to adjust their approaches to the market. The evolution of this business model may very well help assure the long term viability of many of today's enterprise software companies, some of which are currently locked in life and death struggles with competition within their niches and those niches that are adjacent to their core businesses.

To adopt a software publishing model, current enterprise software companies will need to open up their architectures and APIs to outside developer partners and develop the infrastructure and business practices needed to support a portfolio of developer partners. But even before these actions are taken each enterprise software company will need to decide whether it wants to be a publisher or a developer. Many market leading vendors in today's marketplace will be able to pivot their businesses to the role of publisher with little difficulty. But this group will be rather exclusive and second tier enterprise software companies will need to perform judicious self examinations to determine if they have the critical mass to be publishers or if their best course of action might be to become developers teamed with other publishers. A miscalculation here will cause some companies to fail.

### **Challenges Ahead for SIs and VCs**

Some constituencies will benefit more than others under the new model but most of those that adjust will be able to carry on and flourish in the new environment. Two tangential constituencies to the software industry could be particularly affected.

#### ***Challenges for Systems Integrators***

Systems Integrators or (SIs) may be less affected by the new enterprise software business model but they will still need to make adjustments. For years systems integrators have worked with end user software consumers and software vendors by providing the expertise to knit together a company's existing systems with new enterprise applications. While that work will continue, SIs domination of the space will be challenged by software developers who will also have specific domain expertise to offer to the consumer. At the same time, though, it would be naïve to suggest that requirements for integration services will cease simply because there are larger suites of application software. Companies will still require assistance integrating legacy applications, implementing systems, and training staffs, for example. Many SIs will no doubt consider forming close associations with specific publishers or possibly compiling some of the

experience they have acquired through numerous engagements into specific products that fit into a publisher's portfolio.

### **Challenges for Venture Capitalists**

After years of funding start-up software companies, the new enterprise software business model could make software innovation self-funding leaving venture capitalists (VCs) with a diminished role. However, there are steps that VCs can take to minimize their risks and ensure they are able to play in the new environment. For example, the traditional investment role for VCs with emerging companies has been primarily financial and advisory. Some VCs may decide to become more active participants in the operations of software companies and decide to become publishers or to start publishing entities.

Alternatively, VCs that develop close relationships with publishers could tap a steady stream of new investments in companies that wish to access a publisher's distribution channel. Those distribution channels will be larger than the installed bases of most enterprise software vendors today and they will be synchronized to a far greater extent than is currently the case. The hosted application model usually calls for a single version of an application running across the user population. Such a well homogenized user base will make distribution and uptake of popular new modules much easier and faster than it is today and thus reduce some of the risks that new software companies experience today.

On the other hand, the combination of a publisher-developer model plus access to larger and more homogeneous markets will reduce demand for venture capital on an individual company level; however, this also raises the possibility that there will be more fledgling software companies each of which will require funding.

Finally, existing portfolios of software companies in various stages of maturity will need to be rationalized. Given the new model, most emerging software companies within a portfolio will be under pressure to fit into the new model as application development partners for various emerging enterprise software publishers. There will be opportunities and dangers for all concerned and each will need to be reviewed on an individual basis which could lead to some horse trading.

### **Changing the Financial Landscape**

For several years analysts have been saying that the hosted deployment model would take some of the volatility out of investing in software companies. With the more predictable revenue streams that accompany a hosted model, predictions of future revenue would be more precise and dampen the sharp stock price fluctuations the market takes for granted. It was this aspect of application hosting that led some to dub it the "utility" model since the earnings predictability mirrored that found in the utility industry. But as of this writing there has been insufficient experience in the financial markets to prove or disprove this point.

Despite a lack of direct economic experience, there are some things we can reasonably predict about the financial side of the software industry under the new model. For example, the cost of entry into the industry will be reduced, though the real barrier to entry might be in finding a publisher willing to add a new application to its portfolio. To extend the book publishing metaphor, publishers reject many more books than they print.

The combination of lower cost of entry, global access to talent via the internet, and the Web services paradigm, might make it easier for software developers to access the global software market – regardless of where they live. But in a marketplace with increased expectations about customer service, the new model could also provide a new barrier to those without a local presence. If native software developers are to maintain an edge in their home markets it will be through understanding those markets and offering services uniquely tailored to local demand.

### **Analysis and Conclusions**

The enterprise software business model has lasted for more than 50 years — quite an achievement in the high technology industry, which seems to remake itself at least once or twice each decade. The enterprise software industry has entered a new era in which customers expect predictable, low cost products that can deliver on critical needs such as low maintenance, easy integration, and reasonable expandability along with performing the intended mission. Meeting these key new market demands will require a paradigm shift the likes of which the software industry has never seen. That shift represents a disruptive innovation in software second only to application hosting.

The success of the hosted CRM industry is proof of the market power of low cost and high performing applications. Hosting took cost out of enterprise software by significantly reducing the labor associated with acquisition and deployment. But enterprise software is still expensive to develop and market and similar savings can be attained by changing the way software is made and sold. As we have seen, the changes to the software industry will simply mirror changes that have already occurred in the economy at large. Such a paradigm shift will enable developers and publishers to be more responsive to customer needs and will eliminate a good deal of redundant activity now seen in each software company that supports a sales and marketing infrastructure.

But the publisher-developer model will also eliminate waste and redundancy in another way as well. During the late 1990's boom a great deal of venture capital went into new software companies and most of them failed. At the time it was common knowledge that even specious business plans were being funded by venture capitalists whose bankrolls exceeded their vision.

In the publisher-developer model, less capital will be required to fund a new idea because the overhead will be lower and there will be a more

even distribution of risk between the funding agent and the development entity and that funding will come from publishers who have vetted the idea with a clearer view of market demands.

The publisher-developer model will be a catalyst to redeploy vendor resources to focus more on the end customer. Enterprise applications that merely track transaction data are ubiquitous and rather than bringing the vendor and customer closer together, as CRM promised, customers are more alienated and show less loyalty than ever before. If the vendor community expects to make headway at getting closer to customers, vendors will need to access greater domain expertise but the resources to fund that access is currently locked up in redundant activities in areas like sales, marketing, and support. If software developers could offload these activities to publishers who can perform these duties with greater economies of scale, those with domain expertise could focus more of their resources on customers — gathering their feedback and solving their problems.

The combination of application hosting and a new business model based on the division of labor between the software publisher and the software developer will free up resources and creativity that will enable the development of new applications and services for business niches that are now coming into existence. These products and services will generate a new iteration in a long-term improving productivity cycle that, in many cases, started long ago — ideas in peoples' garages.

*Beagle Research Group is a consulting and research organization focused on emerging companies and technologies that will have an important impact on the way business is conducted in the years ahead. Our work is based on professional standards of quantitative and qualitative research which informs all of our publications.*

*This document was researched and written by Beagle Research Group.  
The information it contains is accurate as of the date of publication.*

*Contact the author at [denis.pombriant@beagleresearch.com](mailto:denis.pombriant@beagleresearch.com)*

*Copyright Beagle Research © 2004*

*Beagle Research Group  
264 Greenbrook Drive  
Stoughton, MA 02072  
781-297-0066*

I N S I G H T + A D V I C E + R E S U L T S