Business Driven Applications Come of Age

Key Points:

- What if you could handle 90% of your business requirements in an application built largely without programming?
- What if you could implement application changes in real-time with no risk of data loss?
- On-demand solutions enable business-driven applications.

Why can't Applications meet Business Needs?

When we see how effective computers have been in scientific research, embedded systems, the computer industry, and communications, it's hard to fathom why so many business applications fail to meet their customers' requirements.

Despite a succession of technology promises, all the 'silver bullets' tried to date have made little impression on the problem. Systems maintenance backlogs are growing; customers are frustrated with IT suppliers and internal IT staff; and businesses are ever more reliant on technology – not just to provide information or lower operating costs, but to generate revenue.

If you could drive your applications according to your business needs – what would you want? You would want to handle 90% of your business requirements without programming. You would want to be able to keep applications current with changing business needs. You would want to be able to implement changes in real-time with no risk of data loss. You would want to be able to add low-cost snap-on applications to existing systems at will.

In other words, you would want an On-Demand Environment.

The On-Demand Environment

The basic proposition underlying all on-demand solutions is that a vendor provides:

(1) an off-the-shelf application,
(2) running on a secure, reliable hosted infrastructure,
(3) with flexible real-time change management,
(4) purchased on a usage basis, and
(5) with the business data owned by the customer.

The vendor takes full responsibility for the performance, availability, and functionality of the solution. As a result the vendor needs to continue delivering a superior service level or risk losing business. This competitive pressure forces the vendor to address the maintenance needs of an application, as much as, or more than the initial development. This is a key advantage for the customer, a stark contrast from the normal software sales and maintenance cycle which frequently results in customers being treated as "bought business" after the initial purchase.

However, as a corollary to this business model, the vendor needs to develop a new level of flexibility and reliability in the maintenance and customization of applications in order to be successful in the marketplace. On-demand solutions need robust configuration management and open standards-based integration at the foundation of their platforms.

Combine flexible system configuration and deployment, with packaged solutions that provide 80% functionality out of the box, and one finally has an environment that supports business driven applications. Only the limitations of legacy infrastructure stand in the way of success. On-demand solutions bypass these hurdles – all that's needed is a web browser and an internet connection.

This is the reality of implementing an on-demand application using Salesforce.com. The following case study is an example.

As anyone working in the IT industry will confirm, few if any packaged business systems are rolled out without considerable additional modification.

This is where the biggest elements of risk and expense are introduced.

A Customer Service Example

Implementing a complete customer service application for a large mass transit customer using Salesforce.com required the configuration, customization and implementation of the system across multiple sites, over multiple phased releases, and across multiple organizations. The implementation also included replacing multiple existing legacy systems and integrating data from other enterprise applications.

The key goal of the customer service project was to provide a single integrated system to rapidly and informatively respond to individual customer's service requests, as well as to provide the tools to manage the internal process of problem resolution.

Information reported from the system would be used to help identify common problem areas for follow-up. A separate application that managed Service Quality investigations of operations personnel also had to be developed and delivered in parallel. All the component systems had to feed into a common reporting system.

Customer feedback was received via multiple channels – web, email, phone, walk-ins, letters, etc. All input had to be handled as common case records for reporting purposes. Varying workflows needed to be developed and deployed for web cases vs. phone cases, HQ vs. field, Service Quality investigators vs. Customer Service agents. Management also wanted to develop escalation processes that would alert supervisors after cases had aged beyond an acceptable time interval. Business cases, generated automatically from customer input, might be screened by a supervisor, routed to a customer service agent for handling, then to one or more operations areas for follow-up, according to the particular requirements of each case.

Round 1: Initial Results

The team adopted a phased development approach in order to mitigate the risks of over-customizing the application before business requirements were clear. **After completing a preliminary analysis, the team was able to rapidly develop pilot applications for two business organizations on a common platform over a 4 - 6 week period.** Communication between the staff in HQ (newly trained in the use of the application) and the other operations staff (who were not using the system yet) took place via system generated emails. The emails included a custom web link that allowed the operations users to respond back directly into the Sales*f*orce.com application without logging in.

These pilot applications were completely implemented, permitting full cutover of affected personnel. The new system, albeit in pilot mode, effectively replaced the prior system in production, on schedule, for a limited number of users, according to a project timetable established by Customer Service management. Sounds pretty good, though not unprecedented.

No matter how much up front preparation, the fact remains that most organizations won't fully understand their business workflow and reporting requirements until they have seen the new system in use.

By then it is usually too late to make significant changes.

Round 2: Changes to Business Requirements

But something more unusual occurred next, something that diverged significantly from the normal system development cycle.
Engaging in *second round requirements gathering,* initiated by the business group after the initial system was in production, we discovered that we needed to make substantial changes to the pilot system in order to better handle accountability of the operations groups receiving the case handoff. The fact that we were able to make these substantial changes, with the system already in production, without disrupting the organization's business and with no system fallout or data loss was a very pleasant surprise to our users (and a great relief to us).

On-demand solutions need robust configuration management and open standards-based integration at the foundation of their platforms.

With the system in production, we were able to make extensive real-time modifications to business workflows, introduce additional forms sending sub-cases out for further response to additional operations areas and substantially improve data quality – all introduced by changing the system

The majority of business functionality was delivered by configuration changes and end user tools with no additional programming.

Multiple releases were implemented over the duration of the project – including two Salesforce.com upgrades – with no data loss.

in use. Reporting improvements included the addition of 18 dashboard web pages tailored to different sub-groups, offering drilldown into over 150 operations, performance, and business trend reports introduced during this period. The larger modifications were developed in a test environment (which Salesforce.com calls its "Sandbox"), while simpler ones were initially delivered in an unreleased production mode.

The secondary rollout, based on the experience gained from the initial pilot led to the application being substantially revamped to better match changing user requirements.

This secondary rollout was delivered to a variety of business groups within Customer Service, Service Quality and Operations over a 5 month period, with 150 primary users at 22 different locations. Custom onsite training was delivered to personnel from the various business units. Customer Service management worked directly with the implementation team, designing and planning adjustments to workflows, introducing performance metrics and targeted reporting for the various groups involved.

Throughout this process, some additional customization was applied using the system's web services programming interface, **but core functionality was intentionally left intact on the base platform to simplify ongoing maintainability.**

Round 3: Project Handover

The last couple of months of the implementation focused on the completion of training for users not trained earlier, finalizing reports, and preparing the business to maintain the application. This is the critical final piece to empowering the business. **In order for the business to maintain control of the process after the initial implementation, the organization must plan to commit at least one or two people to system administration.**

Most significantly, the business should know how to develop their own reports and dashboards. **Dashboards are the key interface for tracking customer or constituent activity, as well as for measuring internal performance.** Dashboards deliver a macro overview that includes live report drilldown to the lowest level of detail.

Once the business was able to control the way the application was used with their workflow and able to extract meaningful information for decision making, the implementation team could be disbanded.

The Business will Control the Application

Perhaps not surprisingly, given the intense involvement of business users throughout the project, once the phased deployments had begun, **we started to notice that the core group had gradually become fervent proponents of the new system.** They found that they were easily able to locate and view in-progress cases and access information via canned and ad-hoc reports and views. They were able to detect problem patterns exposed by customer feedback through use of the system's dashboards, and they were able to establish specific operational controls using the built in capabilities of the system.

Having been empowered in the design and development process, and become familiar with the information in the system, **the users felt their interests were directly aligned with the new application** - to a degree that we have rarely seen on other projects.

"The fact that we were able to make substantial changes, with the system already in production, without disrupting the organization's business and with no system fallout or data loss was a very pleasant surprise to our users ."

Consultant Role

New Demand provided consulting support to lead the implementation team. Our role was to manage the implementation project, introduce best practices, ensure that the application design was efficient, design and manage legacy integration, and support technology transfer from the implementation team to the business.

3 Step Process

The three steps to implementing a successful, business driven on-demand solution that this case study demonstrates are:

- **Pilot:** Deliver an initial application to a small group of users. Choose the users carefully. Make sure that they are fully trained. Use them as a core group to drive changes to the system and act as internal advocates.
- **Phased Deployment(s)**: Apply modifications based on results from the prior implementations. Add additional required reports. Import historical data. Train users as required and deploy. Repeat as necessary to address the complete system scope.
- **Application Handover**: Ensure the business is self-sufficient – fully able to maintain the application. Provide additional training so that the business is fully capable of System Administration. Implement control processes to identify and prioritize system changes.

Lessons learned from using Sales*f*orce.com to deliver business driven applications

The experience of the case study above demonstrates that **truly business-driven applications are possible.** On-demand solutions provide a cost effective alternative to traditional packaged software - without the IT costs and infrastructure dependencies of internal deployments. More importantly they are uniquely positioned to enable business driven applications.

The following section offers some guidance to business and IT managers considering this sort of implementation by highlighting the key lessons we've learned from implementing on-demand projects using Sales*f*orce.com:

- On-demand applications using Sales*f*orce.com can take advantage of a built-in Rapid Application Development approach. Using the initial system as a base and customizing as needed based on a phased release process ensures maximum business fit with minimum implementation risk. In addition, the initial pilot is not thrown away, but is fine-tuned and re-used for later implementations, leading to a better quality application.
- Historical application data can be converted and loaded into Sales*f*orce.com, or alternatively the new system data can be exported to a data warehouse for aggregation.
- If your application is moderately or highly complex, you may need to add some Web Services API programs - target these judiciously and try to keep them simple to maximize system maintainability.
- If you are deploying in a large organization, try to release the application to those users most likely to succeed first. They will help to set a peer standard for the more reluctant (or less capable) users that follow.
- It is possible to train users in the application on a multi-release schedule. Once they understand the user interface, introducing incremental application functionality becomes a much easier task.
- Consulting and/or IT support should be utilized to define the system architecture components, provide system integration, and ensure an efficiently configured and tested implementation, but the application should ultimately be owned by the end users of the service.
- Business users should, in particular, own the System Administration function by the conclusion of the implementation. Sales*f*orce.com system administration is focused on business needs, including the ability to create and manage reports, processes and workflows. Make sure to adequately plan the handover of application control and responsibility from the development team to the business.
- Reporting and Dashboards are critical to providing business information and feedback that can enable ongoing process improvement. Teach awareness of these capabilities early in the project.

- Dashboards provide a degree of transparency among members of a working group, between working groups, and between working groups and management. This establishes an environment characterized by information sharing and leads to improved performance (without management intervention) based on peer pressure.
- Increased on-demand information sharing means you will want to spend some time reviewing and enforcing data quality. This can be accomplished through front end edits in the base product, as well as back end cross checking in a custom process.
- If you are using Sales*f*orce.com, check out the “snap on” applications listed on the AppExchange**.** Using some of what you see listed there will be cheaper, faster, and better than building your own systems.
- As the acknowledged leader in on-demand applications, Sales*f*orce.com has a rapidly expanding base of customers and partners. The Sales*f*orce.com user community and the associated web sites are good resources to become familiar with. System Administrators should leverage the power of the Sales*f*orce.com user and developer communities to full advantage.

Author: Ron Pristin – Principal Consultant New Demand LLC
Date: 1 Mar 2006