Developing a Web Browser–Based User Interface for Customer Care

John R. Odden

In an era when mergers, acquisitions, and consolidations happen at the speed of light, maintaining personalized, efficient customer care can be a daunting challenge. However, patients' initial impression—and often their lasting judgment—is formed according to the quality of service they receive. This makes a solid customer care strategy imperative.

When one major medical practice management business underwent rapid growth and change due to acquisitions and consolidations, customer service leadership reevaluated its strategy and chose to launch a web-enabled customer service application with an integrated database. At the inception of this project, all nonclinical issues were handled by a centralized customer service call center. A separate team delivered nursing advice services, and a third handled appointment scheduling. In the midst of rapid consolidation, the customer service function adopted paper-based processes for patient inquiries. They also used Microsoft Excel spreadsheets and Microsoft Access databases for basic tracking and workload reporting.

The increased workload resulting from rapid growth rendered the paper-based processes inadequate. Customer service leadership immediately began evaluating automated approaches. From a longer-term perspective, software packages for customer relationship management (CRM) that include an enterprise workflow engine, based on a standard database platform (SQL), emerged as the ideal way to attack pressing issues.

The group chose an enterprise workflow engine solution as the long-term answer because of its ability to solve customer care issues in healthcare—specifically, its ability to move information back and forth across organizational boundaries. Healthcare issue resolution and other forms of customer response often require information and action from a variety of groups. An enterprise workflow engine's ability to track and enable the timely movement of issues across clinical, administrative, central, and local departments gives customer service representatives the visibility and tools to deliver timely, consistent responses.

Although an enterprise workflow engine was the ideal solution, a quick readiness assessment yielded a number of problems. Insufficient financial resources, the inability to commit resources across the broad range of participating departments, and the challenge of gaining consensus all prevented the customer service team from sponsoring and launching this solution. Furthermore, it became clear that none of the natural sponsors could step up to the plate in time to meet the critical needs of customer service.

These realities, coupled with the increasing need for a better customer service solution, induced the leadership to sponsor a pilot phase project. This reduced-scope activity would not have the broad impact or long-range benefits of the original vision. However, it would remove a major point of stress from the customer service team—the inability to keep up with the volume of transactions using the existing manual process. With this goal, leadership established a new working group to further define the potential pilot phase activity. However, prior to selecting technology and defining the approach for this alternative solution, the working group needed to define the organization's customer service issues.

Defining Customer Service Issues

Before selecting technologies or shaping potential solutions for an alternative "pilot phase project," the working group defined four key customer service issues to be addressed. These issues, which are somewhat common to any health delivery organization, helped the working group define priorities, goals, and requirements. They are as follows:

- Every patient query must receive a consistent, efficient response.
- The organization must employ an effective, automated system for issue tracking and ad hoc reporting.
- The customer service facility must manage the call workload in an integrated, consistent fashion, particularly during peak hours.
- Management, customer service, and clinical staff must work in partnership to devise an integrated solution.

Consistent, Efficient Responses. The customer service design required an efficient central group that could respond to all non-health-related issues, including benefits, procedures, complaints, and administrative inquiries. This design would not only provide access to economies of scale but also would free clinic personnel to concentrate on direct delivery of care. In addition, a centralized unit would enable staff to proactively identify and address local and systemwide issues. Further, it would facilitate better customer service training, placing customer service teams close to expert agents and management resources.

An Effective, Automated System. For this organization, rapid growth via mergers and acquisitions had led to consolidated manual procedures. Basic customer contact information was captured on paper forms. Complex inquiries on matters requiring escalation were routed to specialists. Although it was possible to log responses to individual customer issues, the staff was unable to conduct real-time research. Ad hoc report requests could not be accommodated, as each transaction was handled in isolation. The drive to actively manage service levels drove the quest for enhanced reporting, which would enable better research and evaluation. In particular, on-line access to a transaction database emerged as an essential tool to measure and manage inquiries.

Managing the Call Workload. During seasonal peak workload times, when response intervals are lengthened, customers have a tendency to call multiple times with the same problem. These repeat calls exaggerated the customer service representatives' workload. Further, when the facility employed manual procedures, representatives would not have the caller's history readily available and might provide a different answer to the same question. The customer service facility needed an application that would capture customer identity, provide a time stamp, and designate an issue code, making it possible to rapidly recognize, acknowledge, and reassure repeat callers.

Partnership Among Management, Customer Service, and Clinical Staff. Regardless of the customer service solution's size and budget, it would drastically affect service and internal operations. The customer service working group had dismissed the idea of an enterprise workflow engine because it required too much financial commitment and organization-wide consensus. However, they knew any good solution would require buy-in from key leadership and users.

"Superusers" and key management would have to spend considerable time participating in prototype definition and team reviews. The user group would have to agree to accept significant reductions in scope and functionality compared with the original wish list, as well as decide on an approach to build the application that transferred critical duties directly to their responsibility. Therefore, the group decided to endorse a solution using Web technologies for the pilot application project. In addition, the group endorsed rapid application development (RAD) as the approach. The four key customer service issues (just discussed) drove the development of the objectives.

Developing Customer Service Objectives

The working group needed to create a pilot that would provide urgently needed functionality with minimized effort, yet serve as a platform that could be enhanced or converted in the future. All noncritical functions had to be pushed into the "next phase." Information capture, consistent response, reporting, and compliance were established as critical functions.

From the beginning the customer service working group unanimously cited one-time information capture with no redundant data entry as the top requirement. The scenario already in place involved repeated capture of customer demographics, with possibilities of miskeying or miscoding every time. One-time accurate data entry would reduce opportunities for error and facilitate the merging of discrete information sources into an integrated application.

Consistent response to customers also topped the requirements list. Customers quite naturally experience great frustration when they receive different answers to the same question, especially when the inconsistencies arise from the same department or office. The desire for consistency led to the intent to build the application on “some kind of database.” Working group members immediately began laying the groundwork for the definition of data elements; as such, delineation was crucial to both the pilot and long-term success.

The existing manual process and downstream key entry offered no mechanism for monitoring volume and identifying backlogs; however, a new database would enable real-time reporting as well as ad hoc analysis of transaction volume and customer service status checks. The ability to measure performance against standards and develop remedial actions based on results came alive as the working group finalized the initial database and reporting concepts.

Consequently, the intranet-based pilot application that emerged had the following components:

- A “knowledge base,” or database, which gave staff a single place to retrieve critical information: forms, directories, policies, and procedures
- A “concerns” module for issue tracking and resolution
- A physician-paging capability, which allowed staff to look up on-call information and schedules and page physicians from their desktop computers

Building a Solution with Intranet Tools

Building the solution with intranet tools required many steps:

- Beginning with a minispecification
- Employing the RAD method
- Prototyping and iterating
- Sponsoring and supporting the process
- Succeeding through training

Beginning with a Minispecification. After conceptualizing the pilot phase, the working group had to determine how to run the intranet application on top of the existing technology platforms. Customer service representatives were already equipped with personal computers running terminal emulation to access legacy applications. The joint project team was committed to conducting a quality assurance review of the intended application and process changes prior to commencing work. They accomplished this review by using most of the early prototype screens, along with a descriptive narrative, to create a minispec. This minispec allowed review teams to validate and refine the intended database with minimal business logic. It enabled adherence to project budget and timelines and later served as a guide to keep the teams focused on the proper scope for the pilot phase.

Employing the RAD Method. While prototyping the pilot phase, the working group decided to deploy the RAD methodology for the Web-based user interface development. Using Microsoft's FrontPage 98 and Visual Basic Script, a team of experts (or "superusers") would provide the initial description of the prototype. The development team would create a working model. The superusers would review the model and provide feedback; and the process would repeat. This method was chosen over the traditional approach—which requires formal, detailed specification and design—because the pilot phase was limited enough to ensure success with RAD.

There was some concern that the RAD approach could not create a sufficiently robust "whole application" to fully address customer service. The committee of users who were appointed to serve as subject matter experts was guardedly optimistic that their efforts would produce a tool to facilitate customer service improvements. Some team members needed help understanding the difference between refinements in the user interface and navigation based on the approved data model on the one hand and "blue sky" changes that would require new data tables and major business process changes on the other. However, comparative analysis and reviews indicated the method would indeed enable the developers to adjust and tune the three-tier intranet application to sustain peak volume levels.

The RAD, Web-based user interface method only supports refinements in the detailed structure of the application, and a few energetic team members had a hard time accepting the practical limitations of the method. These individuals wanted to believe their appointment to the RAD team was a charter to engage in drastic business process reengineering of the customer service operation. In the end, however, there were only minor adjustments in the core business processes. For instance, some RAD team members found the practice of manually routing concerns to be tedious and time consuming. They hoped for the creation of an automated issue-routing and tracking module. Manual routing was streamlined through the use of carefully designed drop-down boxes. Subsequent changes provided a rerouting capability that supported easy changes to prior routing decisions. In hindsight, it is clear that some kind of fundamental default routing feature would accomplish some of the objectives embedded in the desired routing automation package. RAD team members were not looking for total automation of this intensive decision process—they just wanted some shortcuts and time-saving steps.

Prototyping and Iterating. After the first RAD review session, the development team applied the requested changes in a few days' time. At the second and following RAD review sessions, some team members still hoped to make major changes in the data and processes—creating an air of increased tension. Although the RAD approach provided some opportunity to refine and "personalize" the pilot phase application, accepting the scope of available customization options was a real challenge. Despite the final design for the pilot phase application, some team members led an ongoing push for "major change." Even when the RAD cycles were complete, a few members continued to express the view that the RAD method should have taken a much broader scope. It seemed that these individuals could not step aside from the original concept of procuring a high-end, full-featured, packaged solution. The realities of the business allowed for just the pilot phase project. This was a hard message for the team to integrate into their work.

Pushing Forward Through Management Sponsorship and Superuser Support. Continuous effort by superusers enabled the team to tailor the user interface and finalize functional content in a timely manner. These users not only attended the RAD sessions to participate in critiquing the prototypes but also spent time talking with their peer group and developing ideas and examples to support the RAD cycles. Their involvement and commitment made it feasible for the iterative refinement of RAD to lead to a viable application.

Management support in keeping the RAD team focused and approving the completed work was a second essential component of team support. During the RAD cycles, managers endorsed a change in working relationships. Team members gained a degree of authority and autonomy that was atypical of their normal roles. In addition, they could see their improvements and input included in the emerging application. This partnership of the managers and the RAD team set the stage for success and shaped the application that became the core tool for the end user community involved in customer service.

Ensuring Success Through Training. As soon as the results of the final RAD cycle were approved, the project team developed a training manual, courseware, and proficiency exams. Prior to going live, each end user was scheduled to attend the training course. Before obtaining a user ID and password for the new application, each user was required to pass proficiency exams. Although this application was "just a few database tables and input forms," it represented enough of a change from the previous manual procedure to warrant what was nearly a complete retraining of the workforce. Fortunately, because healthcare customer service is very complex, the organization already had a skilled, dedicated training team. Without the training team's support, the degree of change would have been risky, even disruptive. Although an intranet application gains valuable capabilities with the addition of forms, dialog boxes, and database connectivity, it also transforms from a casual-use application to a full business system. Anything less than the level of training, preparation, and support appropriate for such an application will likely lead to trouble.

Going Live

Going live led to many challenges—for management, for training, and for the future.

Balancing Potential and Expectations: A Management Challenge. Once the minispec is approved and the RAD team is formed, controlling expectations becomes the chief management concern. Typical user communities do not have a base of experience with customized RAD expectations. Members of the RAD team and their peers assume their new application will provide everything they could dream of. With a traditional application package, users are more likely to accept what they get and adapt to it. Even though the RAD method is limited to customization or refinement around the prototype, the expectation escalates to a desire for completely customized data tables and business logic. Thus, the RAD method is only viable if managers are successful in pulling these expectations back in line with what is really achievable.

Training and Rollout: Surprisingly Demanding. Even though "just an intranet application" is not at all comparable to an enterprise CRM software product, it is important not to underestimate the demands of training, rollout, and support. These activities can be just as demanding for a pilot application as they are for an enterprise CRM package, and members of the business team will need comparable assistance in all of these areas. After any applications change, the basic needs of the business remain. It is too easy to underestimate the impact of "going live" with an intranet application, especially if that application drives databases or contains any business logic to support transaction handling. The safer course is to expect that complete change management, retraining, and procedure documentation will be required. Even if the documentation is stored electronically as web pages on the intranet, it will be needed.

Whetting the Appetite for Increased Features. The reduced scope of an intranet application produced by the RAD method will lead to pressure to add more capabilities. Once users experience any amount of automated process support via the intranet application, they will quite naturally ask for more. The RAD method yields limited-scope results very rapidly. Often, users have the false expectation that equal chunks of data and business logic can be added with comparable speed and economy. Again, expectation management arises as a critical leadership skill. A properly conceived intranet application will handle defined data and procedures—no more and no less. The human relations part of change management involves aligning user expectations with reality to ensure success and satisfaction.

Lessons Learned

As the group implemented its solution, they quickly learned many lessons:

- Do not just prototype the user interface—prototype the processes.
- Plan for phase two—the users will want it.
- Check and recheck all built-in functions—and still expect change requests.

Prototyping the Processes. The best tools for database connectivity and dynamic content management make it reasonable to embed business logic behind the user interface, creating a real application that can handle transactions. Visual prototyping does an excellent job of driving the RAD method. Process changes can be implemented parallel with the creation of the new user interface. Therefore do not forget to define and prototype the processes!

User interfaces can be evaluated by visual inspection, particularly if the reviewers are experts in the functional area. However, process changes are quite another matter. If the processes to be embedded in a new intranet application are not precisely the same processes contained in either manual methods or a prior application, each process change must be validated and integrated into the balance of the business.

Planning for Phase Two. Given the incremental nature of the intranet application capability, some unfulfilled requests for extra functionality should and could be resurrected in the future. Plan for either a second phase or some postproduction incremental changes. The focus on improving the business and adding new capabilities is strongest when the RAD method is in operation. This is also the best time to rank extra functions for possible implementation in a second phase or as incremental change requests. The human resources and agenda during the implementation of the RAD method do a wonderful job force-ranking interface and function changes. This force-ranking process is also optimal for developing content and priorities for future phases or change requests.

When you experience pressure from end users for more capability, it is an opportune time to gather information and develop a business case for additional development. Even if a second phase is deferred for an extended interval, defining the content of the second phase serves as a powerful tool for managing and even containing expectations.

Checking and Rechecking All Built-in Functions. Delivery of business logic is a potential liability with the RAD method. With the customer service application, all built-in business logic was subject to considerable testing and review, yet it spawned an ongoing dialog. Was it correct? Would it interact properly with external systems? Could it be improved or streamlined? Be prepared that each function incorporating business logic will need careful discussion, checking, rechecking, documentation and approval. And after all of that, there will be continual requests for changes.

Going Forward

Going forward required that the group prepare to manage the following challenges:

- Getting ready for broad deployment
- Fielding change requests
- Protecting the investment
- Identifying the system manager
- Ensuring continued success

Getting Ready for Broad Deployment. Once an intranet solution is operating successfully in the customer service area, other departments may insist on expanded capabilities to address their needs. A limited set of intranet functions may also be deployable to other groups outside of customer service. For example, questions about benefits may need to be referred to a claimsprocessing unit. Inquiries about appointment scheduling may need to be routed to a clinic or regional office. Employees who respond to customer service matters may find that their tasks can be conducted more efficiently by direct access to a portion of the intranet application. In this way, the limited nature of a narrow-scope intranet solution makes the push for enterprise-wide deployment increasingly likely. An extended rollout will be much easier if these issues are addressed during initial design and production. During initial design, server and database performance can easily be sized to allow for growth. This is much wiser than targeting the hardware platforms to only 100 percent of the initial user population.

Fielding Change Requests. While RAD is under way, changes to the iterative prototypes are relatively quick and easy. Do not let this expectation transfer to the production application, where changes are not quick and easy. Integrity of data and reports, as well as audit controls where monetary values are handled, depend on formal change control. There is no shortcut for establishing and conducting proper change control on every production application. Intranet applications created via RAD methods arrive in their final form by a nontraditional path. In production status every application depends on the same controls and supports in order to achieve safe, reliable operation. Although many temptations, including savings of time and cost, argue for streamlining the change control methods for a "lightweight" application, no economies should be made. Once the application handles production data for actual patients and their customer service needs, all of the processes and protections afforded any application must be brought to bear. In particular, all editing or adjustments to data values must be logged and approved.

Protecting the Investment Through Backup and Redundancy. Web and database servers on an intranet often are not managed with the same rigor as production-processing environments. However, proper operation of an intranet application demands careful planning and rehearsal of the procedures that ensure reliability and availability. The nightly processes to back up the data and selectively reorganize the database should be tested and followed rigorously. If an intranet application migrates to serve production needs, the whole environment, from desktops to the local or wide area network, to the servers, must be handled according to production system practices.

Some healthcare customer service operations run twenty-four hours a day, seven days a week. This status can place significant stress on server configuration, systems, and people. Not all intranet components are designed to this standard. It falls on the information technology professionals designing and supporting intranet applications to understand production requirements, reliability, and recovery and maintenance windows. Each component of the application environment must be configured, managed, and tested to the appropriate level of performance. Recovery equipment and procedures should be tested prior to production. Otherwise, the recovery process may be debugged as the application component breaks or crashes. If performance requirements indicate the use of cold standby or hot swap redundancy, then the operations processes needed in an emergency must be tested. Design, planning, and rehearsal are essential to achieving safe, successful production.

Identifying the User's System Manager. The likely system manager for an intranet application is often in the end user area. Most intranet applications have user-maintained content and interfaces accessed via a Web browser. In addition, the component-based nature of an intranet application often results in the lack of one consolidated system management interface point. Thus, the appropriate user team members can most effectively manage the components they use most frequently. If system management is distributed, one accountable system manager should still coordinate critical decisions such as production promotion, rollback of new component versions, or recovery of databases. The system manager will also need to ensure that system management documentation and procedures are adequately updated. A system manager located in the end user team is also a good local point of contact for any users or supervisors who have questions or concerns. A remote help desk, even when equipped with sound tools and skilled agents, will at times have difficulty responding rapidly enough to support a typical intensive intranet system.

Ensuring Continued Success Through Best Practices. The customer service application relies on many components—the intranet, servers, personal computers, local area and wide area networks, custom-built databases, and third-party software. In addition to retaining high-quality staff (such as the system manager), service level agreements must be established, with ample reporting to communicate standards and progress. Service level agreements must address the needs of two types of customers: external and internal. On the one hand, customer service representatives must strive to deliver high-quality service to patients and customers. On the other, these same representatives are "customers" themselves and must have an effective, user-friendly system that is reliable and secure.

Conclusion

As this organization realizes the full benefits of an integrated, automated customer service solution, it is presenting a new, improved image to its patients. Customer service, administrative, and clinical staff will now be able to deliver consistent, efficient responses. Management will be able to track issues, generate reports in real time, and keep a close eye on call workloads and procedures. While building the intranet application using the RAD approach, the development team encountered some dissension. However, the approach was successful, and the application has yielded the intended result: ultimately, patients will receive faster, better service.

About the Author

John R. Odden (jodden@fcg.com) is a vice president of First Consulting Group, where he is a member of the Advanced Technology Service Line.