

# Project Profile: Simultaneously Redesigning Call Center Processes and Training

by Peter R. Hybert

## **The Business Situation**

A regional telephone services company had an opportunity. They had grown their business to the point where it made sense to consolidate the customer services functions from three separate locations into a single location to create a call center.

The existing work processes and roles had evolved over time and raised potential barriers to the change.

- Little documentation or standardization of the procedures
- Considerable “tailoring” of jobs/roles to fit the existing individual performers’ skills and interests
- Little to no formal training

One factor that complicated the problem was that the company’s products/services were, well, complicated. Many of the services were “bundled” versions of other services. Some of the services offered to various market segments were very similar, but not quite the same. As just one example, the company sold “long-distance” service to customers. But, the “long-distance” service offered to students in the nearby university dorms was different (and treated as an entirely separate product offer) from that offered to local residents, which was different from that offered to local businesses.

The above could have been manageable if we could have broken the products down to generic elements and then constructed the final offers. But, the computer systems used by customer support specialists added another level of complexity. Over time, new computer systems had been introduced to support new customer segments and new products. So, given the example above, to order “long-distance” required the rep to use one system for a university student and another for a local resident. But, if any of the different long distance customers called for troubleshooting or repairs, a single (but different) system was used to track trouble tickets.

The bottom line: seven different primary products and five different primary (but 37 total) computer systems, each with its own processes and procedures.

Finally, there was the matter of the learning curve. If incoming calls to a new employee could be limited to some defined narrow range (such as only new university orders), then we could at least spread out the learning over a manageable period of time. On the other hand, if everyone would be expected to take any type of call, then they would (theoretically) need to learn *everything* before “going solo.” (The decision was made to train for most situations and have a coach/mentor available for complex situations, if needed.)

## The Solution

The company decided that, since they were going to invest in a call center, they would do it right. They commissioned a project team consisting of experienced customer service managers, training specialists, a former “big three” business consultant as the project manager, and a vice president sponsor. They conducted research into the state of the art with respect to call centers. And they worked full-time to redefine the processes, roles, and tasks for the new center, as well as selecting equipment and designing the workspace.

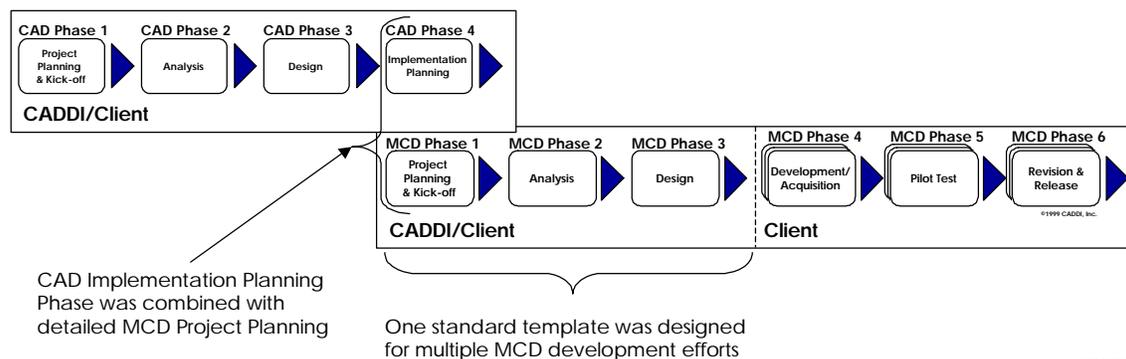
Our role was to help figure out the training design. The team was struggling with the amount of data to be documented and, though they knew the details, they were having difficulty organizing it into a cohesive structure.

The challenge was timing—when the call center was ready, there needed to be a concerted training effort to bring new staff up to speed quickly. The role of my team, as consultants, was to analyze the work and supporting knowledge and skills and then to design a modular training architecture that would allow subject matter expert (SME) teams to create and deliver training quickly. The modules would also have to be configured to make it easier downstream to build more self-paced training/documentation for future use after the initial training demand had been met.

## The Project

### Project Phases

The project phases were based on the methodology used for the project—CADDI’s standard four-phase CAD process linked directly to the first three phases of the Modular Curriculum Development<sup>SM</sup> (MCD) process as shown below.



As outside consultants, our role in the project would be to produce the following:

- Overall structure for the content for training and documentation
- Definition of training modules and events (deliverables)
- Definition of training paths for each of the roles
- Prioritized list of modules for development
- Detailed design of training for one path/role/product (to serve as a template for others)
- Evaluation and testing strategy
- Detailed development plan and project developer information packages
- Development project management tools/processes

### **The Team**

The project work team for the curriculum was the team dedicated to redesigning the call center processes. However, we also worked with an executive-level steering team and top-performing customer service specialists from the existing locations. The teams and roles are listed below.

<b>Team</b>	<b>Role Description</b>	<b>Notes</b>
Project Steering Team	Set the priorities and direction for the project.	The PST was co-led by two vice presidents—the Call Center VP (Operations) and the Skills VP (T&D).
Analysis Team	Help define detailed tasks for “future state,” and identify potential problems/ change issues.	This team included customer service specialists from each location and function.
Design Team	Work with the designers to group outputs, tasks, etc., into training modules and events and sequence them according to how they are needed on the job, in addition to a logical learning sequence.	This team was the process redesign team.
Project Work Team	Conduct and document all the project activities. Provide expertise from the supplier perspective—training and testing approaches that will be effective and feasible to administer.	This team was a partnership between the consultants and the process redesign team.

## **Project Challenges**

### **The Content**

The biggest challenges were technical—there were so many products, roles, and processes that everything had to be broken into the smallest possible pieces. For example, you couldn't teach how to place an order using XYZ system because not everyone would need to know that. Instead, you had to teach a generic order placing process and then offer separate modules on using the individual computer systems, separate modules for each product, and then exercises based on actual customer scenarios to pull it all together. The exercises were designed to not have content, only a performance situation. That way, if the computer system or the process changed later, revisions would be localized and easier/faster to find and make. This also allowed for maximum flexibility and sharing of modules with multiple audiences.

### **The Audience**

Another design challenge was the audience demographics. The current audience was made up of experienced call center personnel who really had never been formally trained on the existing processes, much less the new processes. The immediate future audience was everyone working in the new call center. That included some existing performers (who would relocate) and a majority of new personnel. The initial new call center staff would need to be brought up to speed quickly and as simultaneously as possible.

But, once the initial staff was “spun up,” the long-term audience was likely to be “onesies and twosies” as attrition and growth were projected at minimal levels. The bottom line from a design standpoint was that the design structure (i.e., the configuration of lessons/modules and the instructional process) had to work for both group-paced and self-paced delivery. Ultimately, any materials developed would need to work and/or be easily transportable to some form of self-paced or coached delivery.

### **The Toolset**

The computer systems were mentioned above as an obstacle to learning and performance. They were old school systems (before programmers thought about or had the programming tools to concern themselves with “user friendliness”). On the opportunity side, an improved tool was under consideration. But the decision had not been made as to whether the new tool would be an overall redesign (in which everything might change) or a “front end.” (The front end would use the same underlying systems, but they would look different, and simpler, to the end user). To make matters worse, there was no established time frame for the tool decision, but the call center had to go forward to meet their “go live” target.

## Resources

Budget was somewhat limited, but more critical was the limit of actual human resources for the development of the training. Throughout the analysis and design process, we were able to work with master performers and the call center redesign team. Once we moved to development, the funds were not available to pay for outside developers to actually build the training materials. The responsibility for developing all the presentation, documentation, and simulation materials would fall on the call center design team members. We had to provide a design that was executable with a small development team (with little training development experience).

Our solution to this problem was to invest in some upfront planning to organize the development by designing standard templates, grouping modules for development to fit the call center start-up time line (i.e., develop the training needed first, first), and creating some project tracking and management tools for the team. Fortunately, the team was committed to making the training work and went several extra miles to hit their target dates.

## Related (Nontraining) Issues

In addition to getting people trained, there were still obstacles to successful performance that needed management attention. Some of the key issues were

- The large number of existing computer systems had evolved over time and, as a result, were not fully integrated. This caused performance problems in a number of areas, e.g.,
  - Looking up customer data/responding to inquiries
  - Coordinating orders
  - Handling trouble situations
- Lack of, or variations in, standard procedures, e.g.,
  - Sales response intervals
  - Customer information collection in sales, inquiry, trouble situations
  - Customer follow-up
- Inherent complexity of products and services
- Poor/incomplete communication with customer service regarding new or changed products and services
- Cycle times of some processes are not within customer and/or internal requirements and/or expectations due to understaffing, system access problems, manual processes, or simply too many steps, e.g.,
  - Sales referral/response
  - Service changes/orders
  - Inquiries
  - Dispatching/trouble resolution

We took the opportunity to do some “data-mining” of the performance analysis data for deficiencies and causes. Using a group process, we prioritized the deficiencies and drafted recommendations for the highest priority items based on return and level of effort required. Many of the problems were already anticipated and would be resolved by the new processes planned for the call center. Those that were not were presented to management by the call center team for resolution.

## The Results

The proof of the project was in the implementation. After we left the project, our clients were able to take the output and develop 15 days of training within six weeks. The training included presentations, documentation, simulations, and performance tests. Our design materials provided the structure for both the training and the process documentation (work instructions). And, we didn’t design everything—we designed one of the training paths at the detail level and the team was able to clone our template for all the other paths on their own!

I wish we had ongoing data to describe here, but there is a twist to the plot. Our clients’ company was purchased by another regional telecommunications organization and the call center was put on hold. The purchaser was in the process of developing a large corporate campus and was considering bringing the call center function to their location and integrating it with their existing function.

## Key Learnings

- 🔑 ***Sometimes you can’t win.*** With respect to the buyout, who knew? Well, top management did, but those of us on the project team sure didn’t. The good news is that the design was robust and could be converted to the new company’s products and processes fairly easily.
- 🔑 ***Time spent planning upfront always pays off.*** In fact, our client made a statement to me on the phone that we used for a long time in one of our company brochures. “We have decided there is no possible way we could have been successful without the upfront planning we did with [the company].” The key here is that planning *with the entire team* enabled all the key players to fully operationalize the plan. Each person on the team understood the tasks, the deliverables, why certain decisions were made, what to leave out or go the extra mile to include when time was limited, and which team members were contributing which expertise. Most of the credit goes to the team leader who had selected the right people and had built a well-functioning team before we showed up. But, the process of figuring out all the nitty-gritty detail of the project as a group was critical to the downstream success they experienced in the “two-minute drill.”
- 🔑 ***Nothing is so complex that it can’t eventually be chunked logically.*** Even with all the complexity of this organization and project, the solution required nothing much more than just rolling up our sleeves and making some lists and matrices until we understood the items (for example, the various computer systems) and their relationship to the performance. Even if at first the task seems overwhelming, it is finite. You might have ten products or even one hundred, but you can eventually get them all written down. Once they are written down, people can then react to them. Sooner or later, you have a list that works and can be used as an organizer for the content architecture.

🔑 **Design call center jobs for “learnability.”** Actually, I believe all jobs should be designed for learnability. “Design fors” is a concept from the engineering world in which requirements are derived based on downstream considerations. An example is a computer that is “designed for installability” that is made to be easy to set up. If it were “designed for serviceability,” it might have been designed to be easy to open up with lights that show which module to unplug and replace. Designers of anything, training or job design included, should design for the most important (if not all) downstream user requirements.

When designing a job, one of the concerns should be learnability. Learnability means limiting the amount of knowledge/skill needed to perform the job during the initial period. It could also include additional support as the new employee is “coming up to speed,” such as having a coach available, time designated for learning (which could be used for observing others, self-paced readings, browsing work reference materials, as well as for attending training classes), or interviews with other performers to help with the orientation.

It is a challenge to design inbound call center work for learnability because it is difficult to control the content of the incoming calls—the customer is a big variable. Using automated menus (“push 1 if you are calling about X”) is one solution, but it is often perceived as a negative to the customer. The bottom line is, if the customer service rep has to be prepared for any and every call, it doesn’t leave you with many options...

- Have help available (electronic or live coach/expert).
- Create “goof-proof” procedures that walk through every situation.
- Provide extensive upfront training.

In this case, we used a little of each of the above. We decided that procedures should be well documented. (As an aside, we selected a paper-based reference approach because the customer service reps (CSRs) needed to be able to use their computers for interacting with the data processing systems. When the systems were upgraded later, they could decide whether or not to convert the source electronic files and publish them to the workstation (e.g., via the intranet, and/or design an electronic performance support system—EPSS.)

We also designed a simulation-based training program for new hires to complete before starting to take calls. The training was designed to walk through scenarios based on actual calls in which participants would handle calls (using the actual equipment) designed to cover the most common inquiries.

After training, the learners would also be coached/observed on the job until ready to “go solo.” Our client provided designated coaches to spend time listening in on learner calls for coaching purposes.

🔑 **The best paths are based on scenarios—simple to complex.** This is where structure and learnability can seem to conflict. A nice clean training structure might be to teach all the products/services first, teach processes, teach policies, teach the computer systems in a lab, etc., to embed all the enabling knowledge/skills before getting into application exercises. Unfortunately, this works as well as teaching a child to play piano by making them play scales and finger exercises for two years before letting them learn an actual piece of music. Intuitively it seems obvious that this approach will not work, but it no longer surprises me that design teams often try to do this.

A better path is to start with some of the basics but then apply them in simple situations. Here that meant using customer scenarios wherever possible to teach and practice new knowledge/skills. If there was a policy to learn (such as whether you can give a customer a credit to compensate them for a complaint), the policy would be explained but then examples would be provided to illustrate boundary conditions. Then, as soon as practical, there would be practice in which the learner has the opportunity to make the decision in a variety of circumstances.

This type of learning approach is not really *efficient*—data dump is faster. But, because the training is more similar to the end performance, the transfer to the job will be more effective. The rationale for using this approach (in spite of the additional training time needed) is the same thing my grandmother used to say, “If something is worth doing, it is worth doing right.”

🔑 **If in doubt, make smaller modules.** Deciding how big the “chunks” of training should be is always a challenge. In this project, if the chunks were too big, then downstream changes (which were highly probable) would require more effort to incorporate. On the other hand, if the chunks were too small, there was the risk that the training would be too complex to manage over time.

Given the number of variables (specifically products, processes, and tools), we defaulted to small, fairly discrete modules so that the design would be robust as the work environment evolved over time. The modules were templated and put in a modular inventory organization scheme so that they would be easier to maintain over time.

🔑 **If you can, take a tour of the environment.** It sounds unnecessary when the actual work is primarily cognitive—CSRs performance is primarily based on the quality/content of the information they provide to customers. Watching them work will only show people typing on keyboards and talking on headsets. So, observation of the performance would ideally include verbalization of the thought processes taking place during performance. Yet, even a quick tour of the work environment was well worth the time. Some of the relevant conclusions were

- CSRs worked in four-person cubicles with access to a central set of bookcases of reference information.
- Most of the CSRs were younger employees.
- The atmosphere was casual (casual dress and little “hustle/bustle”).

- They often checked with their cubemates if they had questions about a customer inquiry.
- There were no visible measures to show how the group was performing (it was not obvious who was available to take calls, the current volume, average wait time, etc.).

🔑 **Because ultimately, you design for the end performance environment.** Training is only one part of the human performance support system. Training is often designed to focus on loading up the performers' memory, but the real performance environment allows access of reference materials. In a CSR situation, there are virtually limitless variations of customer inquiries—there is no way to teach all of them. But, you can teach the key processes for resolving customer inquiries by referencing support materials.

Of course, if the training is simulation-based, you can also provide exposure to a large number of cases and incorporate the appropriate support resources as well.

### **Future Plans**

Since we left, our clients have continued to gain benefits from the program. An unintended, but not really surprising, result is that this project has been used as a design platform for two additional three-week programs. And, in the future, they intend to migrate much of the content to an internet delivery platform, which *was* part of the original plan.

From my perspective, I believe the downstream durability of the various designs justified the investment in upfront planning.

- Accelerated development
- Ability to reuse and repurpose training for multiple audiences and delivery media
- Reuse of the design platform to accommodate new audiences and content

Designing for the near- and long-term perspective.

This page was intentionally left blank.