

Successful Call Center Implementations

By: Rob McDougall
Upstream Works Software

May 2005

Upstream Works has an established reputation of successfully implementing advanced call center solutions. We have a practical store of best practices and insights that we have gained over dozens of successful implementations that allow us to stay focused on customer needs while delivering success. In this paper, we seek to share some of that knowledge and insight with you.



About the Author.....

Rob McDougall, President and co-founder of Upstream Works has been a catalyst for change within the Call Center industry for many years. With Upstream Works, he developed a successful business that provides call center solutions to many high profile customers. To ensure that Upstream Works continues to be a leader of innovation, Rob plays an active role in promoting the company through corporate evangelism, articles, and various speaking engagements. Rob is the author of many of Upstream Works' white papers.

Prior to the creation of Upstream Works, Rob held the role of R&D Director for TSB International/Telco Research, where he was responsible for product development and the tactical direction of InterLYNX CT, which was renamed and ultimately evolved into Upstream Works' core product "eMedia CMAS".

He graduated as an Electrical Engineer from the University of Western Ontario.

He can be contacted at rmcdougall@upstreamworks.com

Introduction

Call centers offer a unique set of challenges. They are technically diverse environments, reliant on both voice and computer technology. At the same time, they are critically people-centered - a place where people communicate with other people about issues where they have active concerns. There is no one-size-fits-all solution appropriate for every situation, so we cannot provide you a solid set of rules. Upstream Works has an excellent track record and a practical store of best practices that should help make any implementation a success. We offer you our own insights, organized according to the stages a typical project will go through.

Before starting a call center implementation, you need to understand the challenges you might face and set your expectations accordingly. Call center industry statistics are clear – there is a significant chance your project will be unsuccessful, and an even greater chance of it costing up to twice what was expected. We define success as on time, on budget delivery, the technology acting as expected, and the call center employees being enthusiastic about the results. This paper discusses the steps that you can take to ensure a successful call center integration.

Business Goals

Before beginning anything, recognize the original needs of the business, and ensure that your solution addresses them. Your project is happening because decision makers in your organization have evaluated the benefits and approved a business case in terms of ROI or savings. At the same time, there will be other, less tangible or measurable, factors. Make sure that you understand all of the drivers and factor them in. When you have questions down the road, refer them back to the reasons you are doing the project – keep the original goal in sight.

When you are formulating the goals, include the system's end users in the process. In any large project, you are going to be touching systems that have evolved to their present state for a definite set of reasons. Once you start, you might run into inherited system problems or configurations that are based on different assumptions. You might find yourself fixing problems that were not originally in scope or having to make changes that were not anticipated. Any time there are such changes of scope, they must be recognized and managed *with reference to your original goals*. If you have to make detours, don't get sidetracked.

Working With Vendors

Because of the number of technologies used in a call center, – PBX, IVR, workforce management, CTI, CRM – you will frequently find yourself working with multiple vendors.

In some cases, you will be bringing in new vendors. In other cases, existing vendors will be called upon to make changes or implement new features.

Selecting New Vendors

Broadly speaking, by selecting a vendor, you are choosing two things: their technology, and who they are. Make sure that you have carefully investigated the technology, and evaluated solutions against competitors. You should determine that the preferred vendor's technology has all the features that you want, is compatible with your technology environment, and fits in with the future vision of your organization. But technology fit is only half of the equation: you should also evaluate whether the vendor is suitable as a services provider, and partner/vendor for your business. Choose a vendor that talks meaningfully about the solution, addresses implications of provided features, and seeks an understanding of your business environment.

Working With Existing Vendors

With existing vendors, you need to secure their understanding of what you are doing, and get their buy-in. Whether this is an enhancement or diminishment of their role in your call center, you need their commitment. Make sure they are assigning the right resources to understand and meet your requirements in a timely fashion.

Ensure that everyone understands the complete scope of the work involved, and that your vendors are on board for a fair price. Beware of vendors who lowball the up front price to win the business and make the money up later, through unspecified changes, additional features from 'scope creep', and charges at every turn. When you negotiate, clearly outline what you want from the vendor and how you expect it to be delivered; define who is responsible for what. If any one price looks suspiciously lower than the others, review it skeptically and get a doubly clear statement of work to ensure that your vendor is committing themselves to delivering success. The solution is complex, so be careful. If you can create a win-win situation with your vendor – one where fair goods are delivered at a fair price – there should be no reason for ongoing pricing issues.

Project Management

Having the right project manager can dramatically increase the likelihood of success. A project manager's primary role is to guide the project to completion by coordinating activities. It is the PM who must ensure that all parties meet their responsibilities in a timely fashion and ensure that the project stays focused and within budget.

Your PM need not be technically versed in all the technologies involved in a project, but must understand the overall dependencies. They need to understand where the responsibilities lie and where the boundaries are between vendors. When there are conflicts or differences, they might not grasp all the details, but they do need to make sure that the right details are being addressed. Overall, the project manager's job is to create a team environment - their role is as an expeditor.

Seek a project manager who has previously guided call center solutions to implementation. They will be far more capable of completing the job than someone with a general knowledge of project management. Your options for PM include someone internal, a vendor-provided PM, or bringing in someone external.

Internal PM

An internal project manager knows your business, and how to drive things through in your organization. If someone with a good grasp of the technical issues is available, they are probably your best choice to run the project. They need to understand the high level picture, as well as (with proper technical assistance) drill down on important details.

Ideally, this person should be a prime decision maker in the choice of vendor. If you are unsure of their technical grasp, you may want to consider outsourcing the assignment.

Vendor PM

Having a vendor supply the project management works well if that vendor is performing the bulk of the work, or managing most aspects of the solution. In that case, you may even feel that the vendor is obliged to run the project.

Conversely, vendor project managers are not necessarily objective in their viewpoints. Also, vendors cannot effectively mobilize resources within your company or with any other involved vendors or suppliers. You are still going to have to provide someone from your company with authority who can take ownership of the project.

At the very least, each vendor should be able to manage their own share of the work, as well as any areas of mutual effort where they have the expertise. Choose a vendor as overall manager if the size of their role and their degree of expertise outweigh their disadvantages as outsiders.

Consultant PM

For a large project, it's often a good idea to bring in a contract project manager with experience on similar projects. Contractors are technically and professionally neutral, and can arbitrate issues impartially. Their main loyalty will be to the project itself. The main point to underline here is the experience on similar projects.

In this case, you need to ensure that you give them enough authority to carry out their tasks. There's no point bringing in an outsider, but not empowering them to get the job done.

The Solution's Architect

One key role that is often overlooked is that of the solution architect. The solution architect is the person who has designed the system specifically as it is to be deployed in your environment. There are very real risks involved when this role is neglected or is assumed to be inherited by the project manager. While it may be possible to combine the two roles on a small project, it is highly unlikely that you can do so on a large one: project manager and system architect are two different skill sets.

The solution architect understands how all the pieces of a solution work together. This person may be an internal resource from your company, an outside system integrator, or one of the vendors. Whatever the case, they are responsible for the overall design, and communicating it to the rest of the team. More than that, the system architect owns the solution; while it falls to the project manager to take charge of the delivery and the people involved, the system architect is responsible for the *system working*. When you run into technical roadblocks, the architect must be available to resolve them. If you are putting together an integrated solution in your call center and you cannot identify the architect, then you should do a reality check before starting.

Getting Started

Have a proper kickoff and introduction to get your project off to a good start. Your kickoff should have three main goals:

Communicate the purpose of the project – Explain to the group what the project is all about. From a 20,000 foot view, what is the project and why are you doing it? The explanation should be business-focused and is best introduced by one of the senior internal sponsors. This meeting is to establish introductions and relationships, and to establish why the project is happening. Determine what problems are being addressed and what the ultimate goals are.

Define the criteria of success – This must include the purpose of the project and what it means for the project to succeed. Also, cover the all-important question of the go-live date.

Get the ball rolling – Get the ball rolling with a technical discussion. With a large project, this is likely a separate session, with fewer account reps and more technical people. This should paint the big picture and get the juices flowing for the design and planning sessions to follow.

Have someone draw a diagram of all the pieces, and have the group participate in a walk-through of how calls and contacts are supposed to travel through the system. Each vendor will have an initial understanding of the project from the vantage point of their own technology or business need, and they will already have a set of questions and concerns. Let people deviate into details when it helps them engage.

Be ambitious – you want to get the project running, not crawling. Invite all stake holders from your organization, including business sponsors, implementers, and end users. The vendors have sold you the solution and will be doing the implementation, so they should send both the account rep who made the sale and the manager who is going to implement it.

Business reality dictates that it's not always possible to get everyone into a big conference room on the same day, but first impressions are always important and a project will be more successful if you manage at least one face to face meeting of the principles. If you are doing all of this by Webex™, you will lose the people side of the project. Subsequent meetings can easily be held using any type of remote technology desired.

Note: It's important to know what details are noise and what details may be important. When evaluating, make sure you're not stopping an important go/no go type of decision because it seems minor.

Design Requirements

In many regards, the requirements and design stage is the most important one of the project. This is your best chance to get it right the first time. It is in this stage that you agree on what you are doing, and how you are doing it. Get all the requirements out in the open; don't miss anything important, or it may come back to haunt you.

Diverse requirements come from different parts of the call center and represent the needs of different groups. Make sure that you are consulting all the people who may have requirements, or know of constraints that will affect your ability to deliver on requirements.

You may find that requirements contradict one another. For example, a requirement from call center operations conflicts with a constraint on the side of IT. The only way to resolve these kinds of conflicts without having to enlist top down support is by having everyone with a vested interest in the room. Get the requirements on the table; hash them out. Use psychology: if someone is throwing up road blocks, let them speak up and be part of the solution. Keep people focused on the common goal, and remind them of the value of what you are doing.

Documents and Specifications

During this phase, you are going to generate two kinds of documentation: requirements documents and functional specifications. These documents are a contract between the end user and the people doing the work: between them, they state what the project is setting out to achieve, and how it will be achieved. Your functional first specification should be comprehensive and accurate, and capture the outcome of your design. It should be a composite design document – many different specifications gathered together into a binder.

Be careful that you don't let the specification become a fight between yourself and your vendor - it is a waste of time and money for both parties. Both you and your vendors should have a clear vision of what you wish to achieve, so don't get caught up in an overly complex specification and approval process. Keep it simple, and then work quickly towards a prototype application to review, critique, and improve.

A prototype allows your team to see what the system is going to do and how it's going to work. This allows your team to provide valid input on something concrete. Most people are not very good at creating things out of thin air, but do tend to be good at pointing out how to improve things they can see and interact with. Give the vendor feedback: move this button, change this logic. It's generally not much work to make these changes if the vendor has any grasp of your business requirements and did their homework up front.

Implementation.....

Once you have a good idea of where you are going, you need to start setting up your development systems. Concrete progress can be made in this area while the design is being worked out, particularly when there are questions that need to be resolved before design completion – how is my IVR going to communicate with this backend system? Can I embed these controls in the CRM application or do I have to take a different route? The sooner you have a working environment, the sooner you can work on a prototype.

With all else equal, a good implementation comes down to having the right resources in the right environment. That means that developers and integrators must have access to the systems that they need to work with, that they have the permissions and clearances required to install and configure systems, and that they have the data to simulate real conditions.

Implementation needs to be scheduled in detail, and that schedule should be adhered to as closely as possible. You do not want someone to show up on site for two days of work and be forced to wait a day and a half to be “set up”. When the vendor shows up, the customer must have met their commitments, and be ready. Make sure that your project's requirements are well publicized throughout your organization. You do not want to run into expensive stoppages through scheduling conflicts with other initiatives.

Testing and Acceptance.....

Ownership is one of the key issues of testing. Your vendor is responsible for system testing – ensuring that the entire system is operational from end to end. Acceptance testing, on the other hand, is an approval process where a project owner can determine that the solution meets their needs. Ownership of the acceptance test is therefore solidly with the project owners and sponsors – whoever is going to ‘accept’ the system.

Note that system and acceptance tests may be one and the same. Even in the realm of acceptance testing, the vendors know how to work the system, and must be able to assist with implementing the tests. Ultimately, you need to remember that it's your system. If you got your project off to a good start with a walk through of how the whole system behaves, then you have already laid the groundwork for acceptance testing.

Each vendor needs to submit test cases for their part of the system, and all vendors must contribute on cases that test the working of the system as a whole. The PM must assign one person to be responsible for organizing the cases into a coherent set. Make sure that you do not just test positive cases; that is, make sure that your system testing is not simply confirming that the system works when used the way you expect it to be used. In software development, 80% of the code written handles error situations. Make sure that you are testing the failure points and accounting for how the system works when misused. Still, you will not be able to anticipate all the ways that your system can be abused.

Cutover

Plan everything for your cutover well in advance. If it's a complex rollout, divide it into stages if possible. Know your back-out procedure cold. Your cutover is happening in a production environment, and if something goes wrong, you don't want to leave your agents hanging.

From the moment your project begins, have an eye to its completion. At the beginning, the cutover is still far off in the distance, but the time will come when you sit down to plan the contingencies and list the tasks involved in moving from the old world to the new. You don't want it to be a moment of truth, but an orderly sequence of steps. Any software on your IVR, your CRM, or your agent stations that can be deployed up front should be. The fewer things done at the same time, the better.

During the cutover proper, anything that can be staged should be staged. For example, if you can cut applications over one number at a time on your IVR, or cut agents over to a new application in groups, do so.

Ensure that everyone in the business is well aware of what is going on – what is being changed and how it can affect their performance of their duties, best and worst case. Supervisors should know any critical issues that they need to watch out for, and how and where to report them.

Training can be a tricky issue. Agent and supervisor training before cutover is a must. Any changes in the behavior of the telephones or applications must be thoroughly explained. If training is handled in a hurried fashion, agents are left feeling that they are last to know what's happening, but the first to feel the side effects. Use your training sessions to get agent buy in. Explain the bigger picture, and how these improvements will help them to meet their goals. Remember that good agents use their tools in a habitual, expert fashion. If they are going to have to change their habits, let them know in advance and let them know why.

An important aspect of the cutover is to ensure that all the systems are working, which means that one of the last things to do is to put in a complete system configuration. This is generally the last step before a cutover, but often the project team doesn't realize that the configuration information they need isn't available in any single source, and that what is available can be out of date. Many cutovers have been derailed or delayed by lack of accurate configuration information. Understand what information is required, and gather it up early on. Checking twice to make certain it's accurate will ensure that your rollout goes smoothly.

If you take anything out of the system, make sure that you know how to put it back. Beware situations where you reuse a piece of hardware. If you are installing your new CTI server onto the old one's computer, you ought to know how to reverse the procedure.

Post-Cutover

Don't confuse the cutover with project completion. Some cutovers are clean and end quickly, while others take a while to calm down. Issues can sometimes result from hardware configuration errors that would have been difficult to foresee, and hard to isolate. In this case, you are going to need your support teams putting in extra time to iron out the glitches.

During the project, you work closely with the vendor's implementation teams, the experts in assembling the solution. In many ways, the test of a vendor's caliber comes after the implementation. If you suddenly can't access your expert project team after cutover, then you may be in trouble. While it is unfair to expect the engineer who implemented your solution to answer routine configuration problems, it is reasonable to expect that this person will be available to help resolve issues affecting your call center performance that were not properly addressed during the project.

Conclusion

Call center integration projects are complex by nature. The call center involves a wide variety of technologies that interact not only with your core business systems, but with your company's biggest asset – its customers. It comes down to having the people who can do the job, and following the processes that favor a successful outcome. Many implementations are not considered successes, but by following some of the guidelines listed here, things can be made to work.

Copyright Notice

©2007 by Upstream Works Software. All rights reserved. Unauthorized reproduction or redistribution of this document in any form, including photocopying, faxing, and image scanning is against the law and prohibited without the expressed written consent of Upstream Works Software.

Feedback and More Information

Thanks for reading this report. I hope that you have found some valuable information that will help you achieve your call center and corporate goals. I welcome your comments about this white paper, and invite you to suggest other topics that you would like us to address. My contact information is below.

Rob McDougall
President
Upstream Works Software

Phone: (905) 660-0969

Email: rmcdougall@upstreamworks.com



Upstream Works

www.upstreamworks.com