

WHITE PAPER

What Every Senior Executive Needs To Know Before Buying Contact Center Technology: Assessing The Financial Impact of Technology Maturity On Contact Center Revenues & Operating Costs

July 11, 2003

About The Authors:

Dr. Jon Anton is the director of benchmark research at Purdue University's Center for Customer-Driven Quality and is the author of 18 books and 75 papers on customer service strategy/delivery. He has assisted over 400 companies improve the efficiency of their inbound/outbound contact centers and has been honored by Call Center Magazine as an "Original Pioneer" of the call center industry and has also been named to the Call Center Hall of Fame.

Joseph Katz is product Manager of MCI's WebCenter, a hosted contact center service provided by MCI to both internal and external clients.

Eli Borodow is the CEO of Telephony@Work, a leading provider of carrier-grade contact center infrastructure for contact centers of all sizes since 1998. Telephony@Work's comprehensive CallCenter@nywhere solution has received over 37 industry awards and has been validated by large-scale deployments at fortune 500 companies and top-tier telephone companies delivering hosted services at scale.

Deploying contact center communications technology requires more 'homework' today than at any other time in the history of the contact center. What began as value-additions to traditional PBXs has evolved into a diverse set of mission-critical communications technologies that encompass the phone, fax and internet. Today's call centers, now redefined as 'contact centers,' are challenged by the need to implement diverse communications technologies with seamless integration that spans across all communications channels. What is required is nothing less than the re-definition of 'the call' itself to include the many different ways that customers can interact with the contact center, including e-mail, voice-mail, chat sessions supported by on-the-fly web-collaboration and suggested response templates, voice-over-web calls and web-callback requests. Complicating technology implementations is that customer priority business rules and weighted skills-based algorithms must now be universally applied across all media channels to ensure consistent service regardless of the medium of communication. Of course, this integrated infrastructure must also include quality recording, supervisor monitoring and 'barge-in' capabilities that extend across all channels.

Accomplishing these objectives has historically been expensive, time-consuming and risky from a project success perspective. Mainstream business publications, analysts and academic research consistently report that 60-70% of deployments never achieve their stated objectives. This white paper focuses on the differences between the three core contact center technology deployment paradigms - and the dramatic impact of those differences on success ratios.

System Paradigms: Formulas for Success or Failure

The architectural paradigm in which multi-channel infrastructure is deployed can have far-reaching and often unintended implications. These will impact on initial deployment costs and on enduring re-integration requirements and service costs (as changes are required or as vendors release updates and new versions). They also determinatively impact on go-forward training costs, employee productivity and resulting manpower requirements, reporting quality, and system reliability, scalability and extensibility. Perhaps most importantly, they will determine your company's ability over time to provide superior or even comparable service to that of its competitors.

A framework for paradigm differentiation exists in the vendor and consultant communities that can help buyers understand core differences. This framework suggests that contact center solutions can essentially be divided into three camps, with each representing a differentiated approach to the problem. Understanding the distinction between these paradigms provides a lens through which companies can assess proposals under consideration.

"First Stream" Technology: The Multi-System Custom-Programming/Systems Integration Paradigm

The first of these paradigms is represented by "multi-system solution vendors," who offer various point solutions that are in fact separate, non-integrated systems that are usually provided by different technology-vendors; or by a single technology-vendor that has acquired separate, non-integrated systems through mergers or acquisitions. In this paradigm, each point solution consists of programming tools that require professional programming services in order to deploy each technology. Professional services are also required to 'glue together' the different systems that provide those capabilities, but the result is an infrastructure without any overriding software architecture or a common administration interface. A by-product of this lack of architecture is the cost and difficulty of providing cross-media, historical customer-interaction information on the agent desktop (or even transferring data along with calls); an issue that generates significant customer frustration. In addition, the absence of integration-by-design often results in having to dedicate agent seats to different technologies. This requires foregoing the manpower efficiencies associated with a common agent pool and deploying technology-specific agents. Cross-media licensing is also unavailable since all the vendors involved need to be paid. As a result, you cannot license based on overall communications capacity but must instead license each capability separately – and ultimately 'overbuy' licenses to address constantly shifting traffic patterns between media types.

This paradigm requires large-scale systems integration and involves many consultants, integrators and programmers. As a result, implementations in this paradigm often take over a year to complete. The familiar argument against this paradigm is that its solutions are expensive to deploy and expensive to own. Changes are expensive and time-consuming to implement. In addition, every time one technology solution provider issues a product upgrade release, it must be re-integrated with the other point technologies in the “integrated” solution. Because every vendor issues product-upgrades on different release cycles, each new technology increases maintenance complexity and costs exponentially. There is also the element of danger: who *hasn't* heard of system-wide crashes resulting from simple upgrades?

Fault tolerance is another area of concern. Each subsystem in this paradigm represents a separate point of failure. Instead of “hot back-up,” this paradigm relies on “fail-over” or “warm-backup.” This is an important distinction. “Hot backup” ensures that if any software process crashes, another server on the network can instantly take over without the loss of any information or the disconnection of calls – even on calls in progress. “Fail-over” is an older technology that works simply by disconnecting all communications-in-progress and switching over to a backup system that takes over after ten to thirty seconds of downtime.

“Second Stream” Technology: The Unified, Single-System / Custom-Programming Paradigm

The unified, single-system has attracted a lot of attention over the last decade as an alternative to multi-vendor systems integration. Vendors in this paradigm develop and market unified sets of programming tools that are typically implemented by systems integrators via professional programming services. As a result, while deployment costs and time-to-market are reduced, their solutions still involve many of the same “one-off” quality disadvantages as those in the multi-system paradigm.

Since all instantiations involve custom programming, the quality of deliverables is inconsistent. In addition, since custom programming is required by individual programmers who may not work for the vendor several months down the road, clients must ensure that their code is extremely well documented. Reliability is also an issue here because these vendors rely on fail-over rather than real-time “hot-backup” of live software processes. As with multi-vendor solutions, follow-on programming services are routinely required for implementation of patches and upgrades and programmer one-client focus is essential getting through upgrade debugging cycles with minimal operational disruption. There's another important caveat to keep in mind: a single vendor does not equal a single, unified system. In fact, most single-source solutions are deployed in the first-stream multi-system paradigm; because their suites were the result of mergers and acquisitions or separate team development rather than integration-by-design.

One of the most commonly understood points of differentiation for these vendors is that most single-system vendors deploy their technologies on communications-server platforms; which replace traditional PBXs with server-based switching (to reduce costs and fully integrate communications infrastructure). Many of these “CommServers” typically rely on telephony cards produced by Intel or others – and these telephony cards live inside of Intel-based servers that replace the PBX. The fundamental architectural problem that plagues these second-stream CommServer vendors, however, and the core reason for this paradigm's reputation for unreliability, is that most of these vendors deploy their custom-programmed/custom-compiled software *on* the telephony server – such that any application crash in any media type also brings down the embedded phone system. These vendors typically lack the ability to run or mirror processes in real-time across separate blade servers to maintain the stability of the telephony server. Since every deployment in this paradigm is custom-programmed and must go through multiple debugging cycles, downtime and frustration are generally inevitable. In addition, most vendors in this paradigm rely on the operating system to communicate with the telephony cards in the server – and as such their applications cannot communicate with each telephony card individually but can only communicate with the cards as a group. Why does this matter? Because the failure of a single telephony card in this paradigm will take the entire contact center offline – while software communication with individual cards would enable the server to dynamically adjust to the loss of a telephony card and the contact center's operations would continue, albeit with reduced capacity (i.e. only the capacity associated with the failed telephony card would be affected – and only until its capacity could be cut over to backup resources in the same server).

Most second-stream vendors also fail to leverage their single-source roots and don't provide cross-media licensing; choosing instead to bill separately for licenses of each media-type.

The common concern about CommServers regarding scalability is often justified but is not universally true. Vendors who cannot scale across multiple servers (i.e. to link multiple CommServers over a backplane to work together seamlessly as a single system) tend to overstuff servers with telephony cards to scale, which generates excessive heat and results in excessive hardware failures and reduced hardware life for expensive telephony cards and servers.

Those vendors who *can* link servers together to work as one generally do not suffer from these limitations - particularly if their software is designed with a network-based architecture. Many vendors *claim* to be network-based, but what they often mean is that their technology is distributed across multiple servers that perform specific tasks. True network-based software architectures can eliminate traditional scalability limitations entirely – by spawning processes that can be spread across an unlimited number of servers; re-defining scalability as a flexible barrier limited only by the physical processing resources of the communication infrastructure network. Here the network becomes the computer with processes communicating with each other over that network to form the application. Need more processing resources? Plug in another server. No limits. This is where the industry is headed. For the next paradigm, this vision has already been realized.

“Third Stream” Technology: The Single-System, Menu-Driven Customization Paradigm

The questions that all customers inevitably ask is: “why haven't vendors matured their technology into commercial out-of-the-box technology that is fully customizable through drop-down menus and parameter selections instead of custom programming? Why is it still necessary to examine dozens of technologies and vendors in order to make good decisions? Why is it still necessary to retain and manage consultants, system integrators and software developers for multiple point technologies?”

Those were the questions that engineers began asking in the mid-1990's. Why hadn't contact center technology matured such that it could be fully customizable from menu-selections rather custom programming? – and why, understanding the scope of the issues, hadn't anyone decided to white-board the problem and create an architecture integrated-by-design to work seamlessly and flawlessly?

This question might have gone unanswered but for the fact that the two prevailing paradigms couldn't deliver on the ambitions of large telephone companies such as MCI. What these telephone companies wanted was something that enterprises had also wanted but no one was yet willing to provide: technology that deploys quickly without sacrifice in customization capabilities, that wouldn't require man-months or man-years of professional services to implement or maintain. Since none of the traditional technology vendors willing to abandon their old business models for this new approach, a new class of vendor and technology paradigm had to emerge address this need. This “third-stream” paradigm delivered scalable, comprehensive multi-channel infrastructure in a true, network-based architecture.

“Third-stream” technology deploys easily and predictably through menu selections – without compiling unique software – and delivers sophisticated customization capabilities at no-cost in a browser-based environment. This “third-stream” approach eliminates all of the architectural limitations of the “second-stream” approach – providing both carrier-grade scalability and “hot-backup” reliability. Cross-media licensing is also an attribute of this new approach. Over a period of many years, engineers at third-stream vendor Telephony@Work (who's CEO co-authored this white paper), and at phone companies such as MCI (whose product manager is also a co-author) collaborated to develop this menu-driven technology, which has been deployed as the core technology of MCI's WebCenter hosted contact center service as well as for on-premises technology deployments in companies of all sizes, including a diversity of Fortune 500 companies.

Funded by carrier revenue, this “third-stream” paradigm became an option for enterprises only in the last few years, delivering the unique value proposition of carrier-grade reliability and scalability at enterprise-level

pricing. In 2003, Purdue University's Center for Customer Driven Quality and Benchmark Portal analyzed the technology and presented its conclusions in a variety of webinars and in its academic research. This research yielded clearly compelling results in both the technology's ability to dramatically increase revenue and slash contact center operating costs by as much as 90%. In one case, simply by migrating to third-stream technology, a study subject increased their revenues by 3500% and became a top-50 outsourcer in the United States. From the perspective of dramatically reduced cost-of-ownership, the replacement of legacy technologies can be justified even early in legacy lifecycles; on the basis of almost immediate operating cost savings that exceed the cost of technology acquisition. The conclusions of this research were clear in that third-stream paradigm clearly offers carrier-class infrastructure that installs faster, at lower cost and with greater reliability than traditional first-stream or second-stream approaches. Reflecting the new sensitivity to provisioning costs, particularly for carrier-hosted enhanced services, customization is implemented through parameter-based configuration menus instead of professional services. This enables low-cost deployment and no-cost changes and updates without sacrificing functionality, algorithm customization, customer priority business rules or any of the elements associated with custom integration and programming.

Third-stream technology can be deployed on legacy CommServers (with two or more blade servers performing off-box processing in a "hot-backup" configuration to bring carrier-grade reliability to previously unstable environments), in a distributed network or in incarnations that support legacy PBXs. It also deploys as a pure softswitch solution. Also compelling is that it enables multi-site centers to become service providers for their regional centers and telecommuters – with skills-based routing to the best-available person, site-based or telecommuter, regardless of location. This paradigm's overlay capabilities also solve another major problem specific to multi-site contact centers making the multi-channel leap: load balancing all media-types across locations. (Remember, Signaling System Seven only load balances circuit-switched voice communications. Web-enabled multi-site operations must also implement multi-channel load balancing across locations).

For those who prefer to buy services rather than equipment, third-stream technology also enables and empowers access to 'virtual' infrastructure from companies such as MCI, which eliminates traditional time-to-market and cost barriers with customized deployments from browser menus – without sacrificing routing algorithms or business rules. Bundled with expert advice on best practices, no set-up fees and no minimum commitments, companies like MCI are democratizing contact center technology for those companies that lack the budgets, competencies or the will to manage their own multi-channel infrastructures.

Conclusion:

Responsible decision-makers must look beyond the surface level of feature/functionality to find answers to the really big questions: about core architectural efficiencies, stability, realistic up-time, future scalability and extensibility. It is system architecture that will ultimately either unleash or severely limit your contact center's operational efficiency for many years to come. The decisions you make up-front will inevitably determine your range of possible successes over time.