

Doubling Your Productivity When Maintaining Self-Service Phone Portals

A Productivity Study Prepared by Dr. Ingo Bors

Dr. Ingo Bors

 **ibstar** consulting services

Table of content

- Executive Summary 3
- Introduction 4
- Common Maintenance Tasks 6
- Average Effort for Each Maintenance Task 8
- Productivity Assessment Model 9
- Results 9
- Conclusion 10
- Appendices: Detailed Descriptions of Each Maintenance Task 12

Target Audience: IVR Managers, Development Managers, and IT experts who are:

- 1) responsible for delivering state-of-the-art voice applications with flexible dialogs and personalization options, comparable to personalized dynamic web sites of today, and
- 2) facing high support and development costs associated with the maintenance and management of existing phone portals, or are developing new, Java- and or VoiceXML-based systems installations.

This paper focuses on the maintenance productivity advantages of phone application servers when used for building self-service phone portals.

Executive Summary

Companies have long recognized that forging personal ties between the revenue-generating customer and the service-oriented enterprise frequently yields the most successful business outcomes. Personalized interactions create long-lasting relationships and drive repeat business.

Savvy enterprises have turned increasingly to self-service phone portals to engage their customers in a cost-effective manner, yet most companies have nevertheless neglected opportunities to harness the fundamentals of personalized service and transform their phone portals accordingly. Many customer satisfaction surveys demonstrate that poorly designed and managed self-service phone portals often lower customer satisfaction and increase customer frustration — the opposite result that companies strive to achieve.

Even as pressure builds to improve customers' phone self-service experiences, enterprises also must react deftly to changing business conditions. Gauging market dynamics and leveraging new customer behaviors require organizational flexibility, and IT infrastructures must adapt quickly to changes in business processes and keep costs under control. Thus, the adaptability of the self-service phone portal is essential to successfully negotiate an ever changing business terrain.

These market conditions demand technologies supporting the highest possible productivity that companies can achieve through their change management processes.

In the realm of software development, businesses have rightly paid attention to, and profited from, personalized self-service systems and optimized application change management. Companies have not paid nearly enough attention, however, to these concepts in the context of phone portal development — and just at the critical moment when consumers are encountering self-service phone portals based on VoiceXML technology. Even though companies can now benefit from sophisticated VoiceXML-based self-service phone applications written by Java programmers, they nevertheless lack an environment in which to create scalable, personalized phone applications that are easy to maintain. And those same companies adopting VoiceXML for their phone applications today often do not predict the high cost of maintenance associated with the increased complexity of their phone portals.

A new technology called the phone application server promises companies a clear productivity advantage in updating and changing their phone portals. As we have found in our experience and demonstrate in this paper through the close study of a representative company and its investigation of maintenance efforts for the VoiceXML-based voice portal, companies with a phone application server, such as VoiceObjects Server, can achieve at least twice the productivity or more over current methods of managing updates and changes to phone portals based on Java programming.

Introduction

Most companies implement self-service phone portals to increase efficiencies in their customer service operations. They leverage phone self-service to improve the routing of calls, provide customer account information, or enable customers to execute transactions over the phone. In addition, companies often seek to supplement existing agent-based services by offering a self-service option available 24 hours a day, 7 days a week to their customers.

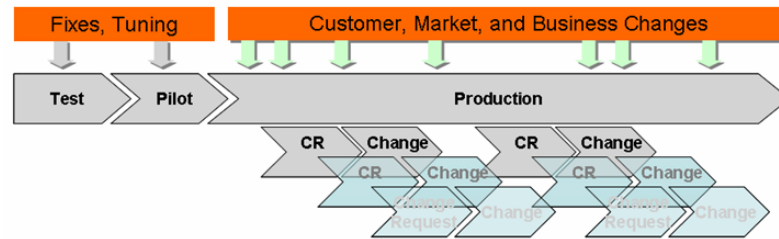
In our experience, we have found that customer service operations often focus first on the initial design and development of the phone application, and then, once the phone portal goes live, operations will shift focus to improving the portal's service quality through regular updates to the application. Many companies regularly make several simple changes to their self-service phone portals, sometimes daily changes. In addition, they may make more complex revisions to their self-service phone portals in less frequent intervals. Common application change requests include:

- Modifications driven by market or business conditions:
 - Emergence of new competitors whose appearance prompts companies to differentiate services by customer segments or to create new product or service offerings;
 - Changes to retail store information, such as changes in hours of operation;
 - Alerts and announcements for transportation companies or network providers.
- Modifications driven by customer behavior:
 - Changes to needs for marketing campaign support, such as enhancing dialogs, to accommodate inquiries about a new offer;
 - Changes to product availability, such as addressing inquiries about non-deliveries or about expected availability of popular products;
 - Changes in needs to respond to public events, such as countering negative press on product or company issues.
- Modifications driven by business processes:
 - Requests to analyze customer behavior in new ways;
 - Organizational changes in customer service operations to support pre-qualification, routing, or prioritization of calls;
 - New or altered self-service processes such as the addition of SMS or email notification, user authentication, and up-selling or cross-selling of related products.

We have found that companies, whether they have traditional hardware-based IVRs or VoiceXML-based phone applications, usually take a multi-phase approach to implementing and enhancing their self-service phone portals. In the first phase, companies focus on achieving a solid initial launch of their portals, and therefore develop and deploy phone applications with a limited number of dialog options and adopt IT and business processes associated with the least risk to the launch. Then, as companies acquire greater familiarity with their phone portals, they add more sophisticated dialogs and increasingly complex IT and business processes. This approach is not a mere academic exercise; it has in fact proven very successful in many projects,

whether they be large or small. However, this multi-phase approach — from cautious simplicity to increased complexity — implies continuous evolution of the phone portal to produce on-going success (see Figure 1).

Figure 1:
Production change management represents most of the personnel investment in self-service phone portals.

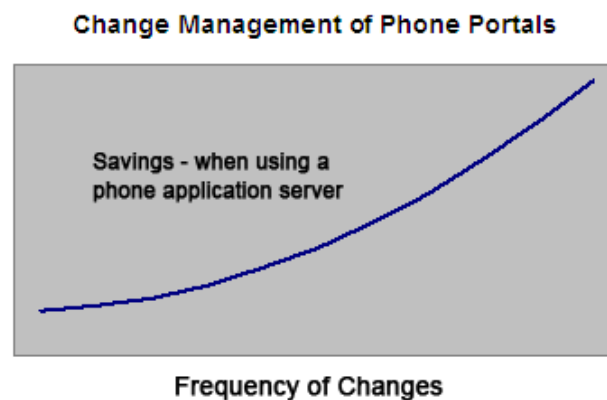


Ideally, all phone portals are dynamic. In our experience, we have often found that the transition of phone portals from simple to complex systems proves quite brief. We have observed that some production systems can become extremely complicated to maintain – so much so that companies determine that they cannot afford to make essential updates and changes. At this point, the self-service phone portal turns into a quasi-static system. The operations team may fall prey to thinking it should never touch a running system, which is counter productive to fulfilling the needs of customer service centers.

The focus of this paper is to demonstrate that optimized change management processes, achieved through use of a phone application server, produce significant productivity gains. Generally, the potential savings benefit increases considerably the more frequent the changes to the phone portal (see Figure 2).

Figure 2:

General Effect of Change Management on Savings and Productivity. Companies that make more frequent changes to their self-service phone portals running on a phone application server will experience greater savings.



We have conducted this paper's study in the context of a specific company's use of VoiceObjects phone application server technology, and we included the assumptions of the company's underlying operational model as well as evaluated the total cost of application maintenance. We first identified and analyzed the most frequent and most important tasks for making changes to self-service phone portals, and then, using the findings of that identification and analysis, we compared the costs associated with making those changes to phone portals that include VoiceObjects phone application server technology to phone portals that do not include any

phone application server technology. Of course, results may vary based on how often a company updates its self-service phone portal and what kind of changes a company makes, but based on our experience, we expect that the overall conclusions of this paper remain valid.

In subsequent sections of this paper, we describe the following steps in more detail:

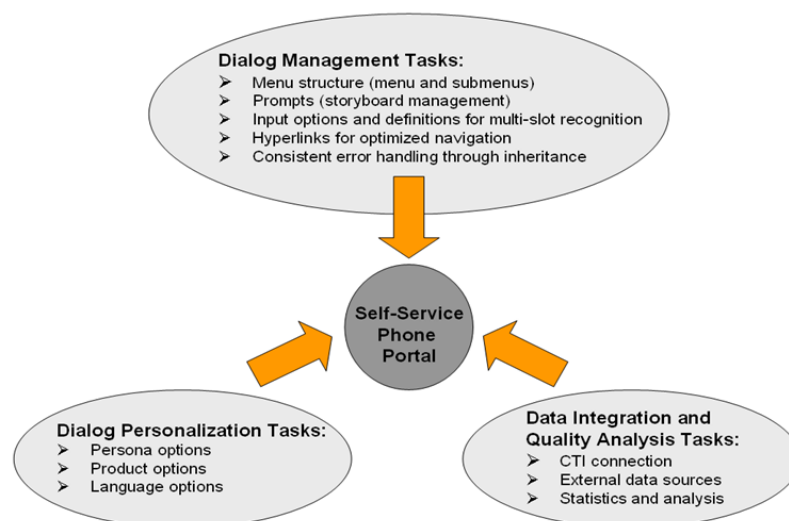
1. Identifying common maintenance tasks associated with self-service phone portals and their update frequency, as determined by the investigations of the reference company we studied (a mobile phone carrier) with a self-service phone portal of average to high complexity
2. Assessing the average effort each task requires (with VoiceObjects technology and without VoiceObjects technology) based on the assumed operating model of the company under study
3. Compiling the data and deriving a yearly productivity model from the data
4. Summarizing the results in terms of overall productivity gains for the company examined

Common Maintenance Tasks

For this study, we have examined a company's phone portal experience and used it as a source to identify and describe in general terms the change management requirements for phone portals. We then analyzed the impact of several of the most common tasks associated with phone application change requests. To simplify the data summary, we have clustered change request tasks into three categories: dialog management tasks, dialog personalization tasks, and data integration and quality analysis tasks (see Figure 3).

Figure 3:

*Summary of
Common Tasks
in Updating and
Changing Self-
Service Phone
Portals.*



The first change-request task group (Appendix 1: Dialog Management Tasks) relates to the management of the dialog itself and produces the greatest frequency of change requests. Companies need to change dialog prompts or tune the phone portal's error handling. The self-service phone portal may, for example, ask callers for additional input to authenticate themselves, to receive special offers, or to select new menu options. These new inputs may directly affect the menu structure, but because the navigation logic must remain consistent, application developers will need to update hyperlinks to define which hyperlinks are global and which pertain only to specific submenus. And, of course, if new items require similar attributes, application developers must maintain consistency throughout the entire dialog structure.

A second change-request task group (Appendix 2: Dialog Personalization Tasks) involves the personalization of dialogs, which allows callers to make use of personalization features in the phone portal or allows the phone portal itself to automatically adapt dialogs by accessing databases with customer profiles or by assessing and responding to real-time customer behavior. For example, expanding the reach of the phone portal to a broader customer base may require introducing the most suitable persona for different customer segments, adding a new product or brand, or offering the same dialog in a different language. While these kinds of changes may occur at a lower change frequency, they nevertheless present a significant challenge to the service organization.

The third change request task group (Appendix 3: Data Integration and Quality Analysis Tasks) includes communication with external systems and analytics. Changes of infrastructure, content, or business processes may directly impact the integration with the phone application. Modifications of the data exchange with the CTI-environment can drive significant system changes. Finally, the examination of system statistics and customer behavior reports may drive further changes.

Average Effort for Each Maintenance Task

As a next step in the study, we summarized each of the tasks described above and evaluated the productivity gain of using the VoiceObjects phone application server versus coding the application in Java. Please see the summary in Table 1 below.

Table 1: Productivity Gains with VoiceObjects.

Task	Task Description	Productivity Gain: VoiceObjects vs. Java Coding
1	New menu item	Application developers can build a new menu item in a single step. Developers simply create a new object and assign attributes including output, input, and error handling as well as persona and language.
2	Additional submenu	Application developers can add or change a submenu which can then automatically inherit previously defined attributes from the parent object. Developers save time in managing attributes across the application.
3	Prompts (storyboard management)	VoiceObjects places all prompts and all announcements in a central repository so that application developers can achieve consistent management of all prompts and announcements and thus save time in application maintenance. Prompts can include text-to-speech and pre-recorded elements.
4	Input options	Application developers can take advantage of speech-platform independence, assigning pre- & post-input prompts, and defining the complete error handling protocol. Through use of a consistent format, application developers can save time adding, deleting, and updating input options.
5	Hyperlinks	Developers need fully flexible administration of hyperlinks, anywhere in the application, at any time, whether global to the application, specific to a customer, or confined to particular submenu dialogs. Application developers save time by easily identifying hyperlinks and making essential updates.
6	Error handling	Developers can maintain consistent error handling at all times during operations and cut development time. Developers can easily adjust all input-related objects and pass general attributes to other objects through inheritance.
7	Additional persona	Application developers save time on implementing a new persona because once they define the global application settings as a "layer," every object inherits all persona-specific additions and updates.
8	Additional product	When companies add new products or brands, developers can extend the phone applications according to the new portfolio or they can add customer-specific call logic. Developers save time because they can reuse entire applications, significant parts of applications, pre-defined modules, or even previously defined objects.
9	Additional language	Application developers save time on implementing new languages because once they define the global application settings as a "layer," every object inherits all language-specific additions and updates. Using VoiceObjects, developers can easily build "multi-lingual" dialogs that are independent of both the VoiceXML and speech recognition platforms.
10	Adding CTI connection	Application developers can use flexible application programming interfaces (APIs) to synchronize caller preferences and profiles with the CTI software. Proper synchronization helps route callers to the most appropriate agent.
11	Connecting additional data sources	Application developers can integrate self-service phone portals with almost any data source through XML, JAVA or SOAP (Web Services). Developers save time because they can easily access the interfaces and use standard protocols.
12	Customizing dialog statistics	By editing the VoiceObjects log object, application developers can add new metrics or new parameters to customize the data collection. Because a relational database stores the statistics, developers can easily access the entire set of data.

Next, we determined the effort in person days to accomplish each change — with the VoiceObjects phone application server and without a phone application server. Then, we estimated the frequency of changes belonging to each task, respectively.

When reviewing Table 1, it is important to note that the productivity increase we discerned with VoiceObjects relates to any phone application whether the application uses voice, video, text, or Web channels as interfaces. It is also worth noting that many of the change request tasks will occur in high frequency during final implementation, testing, and tuning of a phone application.

Productivity Model: Assessing Annual Savings

In the final step of our study, we estimated the potential savings for a given period of time when using a phone application server:

$$\sum_{task1}^{task12} \frac{changes_per_year(task) * efforts(task) * [prod_gain(task) - 1]}{prod_gain(task)} = Savings_per_year$$

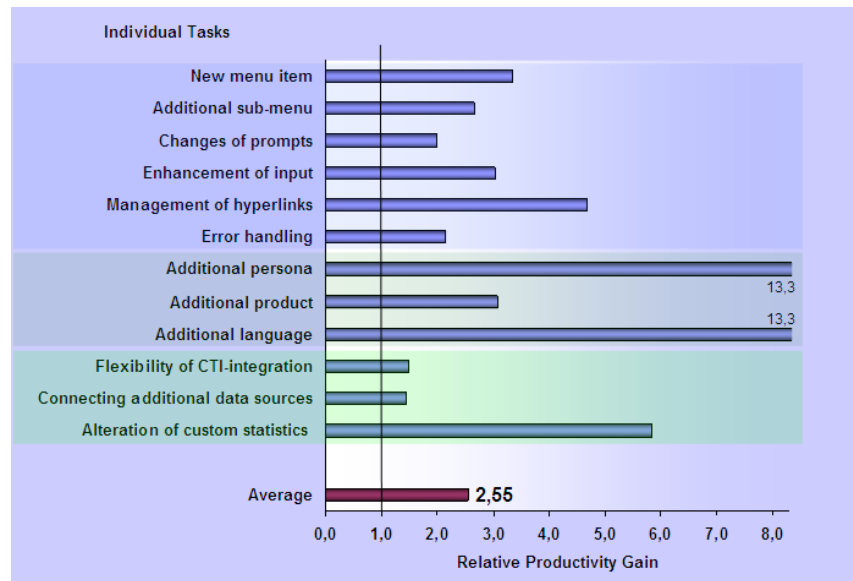
For example, the first task listed in Table 1 is Menu Structure: Add new menu items or new submenu. This task requires about 2.5 person days in a Java-based environment. Frequency of change is once per month, hence a total of 12 changes per year requires 30 person days. Using a phone application server would require only about 9 person days per year – a savings of 21 person days which represents a productivity gain of about 3.3. The derived productivity gains for all tasks listed in Table 1 are shown in Chart 1 below.

Results: VoiceObjects Yields High Productivity Improvements

In our study, all tasks resulted in significant productivity gains when using a phone application server (see Chart 1). Adding a persona or a new language produced the highest productivity gains, reducing by more than 10 times the effort needed to code the same update in a Java environment. Productivity improvements of approximately 50% relate to changes in the areas of database connectivity and computer telephony integration (CTI). This stems from the overall integration efforts which do not directly depend on the phone application.

On average, a company using a VoiceXML based phone portal proved more than 2 ½ times as productive with a VoiceObjects phone application server than without it.

Chart 1:
Productivity Gain
by Task (Average
Application
Complexity).



Conclusion

When planning a self-service phone portal, particularly one that leverages the rich possibilities of VoiceXML, we recommend that companies consider the processes for change management and their impact on ongoing operations. We further recommend identifying the drivers for changes and the frequency of change details as fundamental to the overall calculations of the Total Cost of Ownership (TCO).

For today's self-service phone portals, choice of platform will determine portal flexibility as well as the amount of future operational investment required for managing systems. The project data presented in this paper clearly show that choosing a platform based on a phone application server like VoiceObjects provides fundamental and significant advantages with:

- lower cost per change
- faster change completion
- reduced risk of change

Companies can experience twice the productivity for frequently requested application-specific changes such as menu items, prompts, and error handling. Ten-fold increases in savings can result during the process of adding languages or personas. Customizing statistics during operation — a task for which the effort required is often underestimated — becomes five times more efficient.

Our research shows that several VoiceObjects design elements account for the increase in productivity. First, the VoiceObjects Server, a phone application server, uses a modular approach providing clearly defined entities or objects allowing significant flexibility to accommodate new

requirements. Second, VoiceObjects' support for Service Oriented Architectures (SOA) provides for standard interfaces to distributed data and applications, which creates significant flexibility to access information stored through business processes in related business applications such as CRM systems. Third, the personalization engine embedded in the VoiceObjects product allows for the real-time generation of dialogs that adapt according to customer profiles or behaviors. Finally, the ease with which changes can be added or rolled back gives significant control to phone self-service operations.

Phone application servers like VoiceObjects are becoming central to the evolution of phone self-service, providing the link between the phone system front end (phone application) and underlying business logic. When combined with a strong operational model, investment in a phone application server will yield a substantial gain in productivity for companies' phone self-service operations.



About the Author

Since 2005, Dr. Ingo Bors has collaborated with enterprises to optimize their processes for improving customer service. Before that he managed projects in the area of customer relationship management (CRM) and was responsible for product launches, market analysis, and sales trainings.

Dr. Bors' primary focus is raising efficiency of the contact center and of service automation through voice dialog systems. Another key focus is the close cooperation with SAP for multi-channel integration with mySAP CRM.

After his studies of physics, astrophysics, and modern China, Dr. Ingo Bors continued his career at Siemens AG. As technical project manager he was responsible for the launch of one of the first pre-paid services in German mobile networks.

Contact: Ingo.Bors@ibostar.net – +49 178 478 62 39

ibostar consulting services – Sperberweg 15 – D-51427 Bergisch Gladbach

Appendices

Detailed Descriptions of Common Maintenance Tasks

Appendix 1: Dialog Management Tasks.

Task	Description	Use Case
Menu structure	Adding new menu item or new submenus.	Introduce a new function through a new menu item. Enable access to new information or services through dedicated submenus and link the submenus to menu options through hyperlinks. Ensure appropriate dependencies on the menu tree's logical structure.
Prompts (storyboard management)	Changing audio, announcements etc.	Change prompts to incorporate customer feedback, to optimize dialogs, to improve messaging etc. Most frequent task.
Input options	Providing new input including error-handling and definitions for multi-slot recognition.	Introducing new input options, which require many attributes for error handling, random prompting, grammar design, language detection, pre- and post-input prompting as well as additional help prompts. Each result must have a defined reaction of the system. The logic around an input should be regarded as an entity of its own.
Hyperlinks	Optimizing navigation.	Support hyperlinks, which are a central element of an intelligent dialog. They allow the caller to efficiently navigate within the dialog and provides the caller control over the application. Hyperlinks need to be managed carefully because they link information across the entire application.
Error handling	General updating for consistent error-handling; changing the error-handling for specific input-items during runtime.	Consistent error handling is required throughout the dialog. New error handling will be required for new input and logic and to address results from quality monitoring. Intelligent error handling in most cases can significantly impact the customer experience so we advise great care design and implementation. Error handling increases in complexity as the number of dependencies in the application increases. Ideally, the dialog inherits global attributes throughout the dialog, which ensures consistency and efficient management of updates and changes.

Appendix 2: Dialog Personalization Tasks.

Task	Description	Use Case
Additional persona	Adding separate "Hear&Feel" as necessary for specific customer segments.	Provide a choice of personas as an extension of an existing dialog. Any prompt-related parts of the application requires change, while the overall logic and call flow remains identical. Changes and further evolution of a persona should not jeopardize application consistency.
Additional product reflecting personalized dialogs	Adding new products, often in conjunction with a different persona.	Adapt the initial application to provide specific dialogs for customer segments or to address new brands or products. The dialog will often intelligently draw on call- and caller-related data to control the dialog. Adding support services for a new brand may reflect the existing dialogs, but these services are accessible only for customers of that brand.
Additional language	Adding dialogs in multiple languages.	Extend existing dialog to accommodate several languages. All prompts and inputs are language sensitive, while the overall logic of the application does not change.

Appendix 3: Data Integration and Quality Analysis Tasks.

Task	Description	Use Case
Backend: Adding parameter for CTI connection.	Passing new parameter to aid in improving CRM processes or to provide additional information on screen-pop for agents.	Achieve required integration of phone portal with essential enterprise systems to drive new services on the self-service portal. CTI connectivity is fundamental for the transition from the self-service portal to subsequent assisted service. This comprises aid in routing calls to a live agent as well as providing additional information on the agent's screen for optimized call handling.
Backend: Connecting new data sources	Integrating with a database, a content management system (CMS), or new web services for data exchange.	Optimize the link between customer and business data that drive personalized dialogs. Information portals retrieve data from a variety of content sources. Content Management Systems can be used to manage the content. Flexibility in connecting data sources is key for any application which depends on external information.
Customizing dialog statistics	Generating application and caller-related statistics.	Give operations teams insights to enhance data-gathering related to portal performance. The core of quality management is based on the perfect stack of reporting data. As management requirements for reports typically change with growing knowledge and often as frequently as every quarter, operations teams will need the flexibility to implement the new reporting requests anytime during operations.