

“An Approach to De-risk Your Large Scale IVR Application Migration Initiatives”

By:

Rajeev Sharma & Aman Kakkar

IVR Practice,

Communication, Media & Entertainment Division,

Infosys Technologies Limited

Abstract

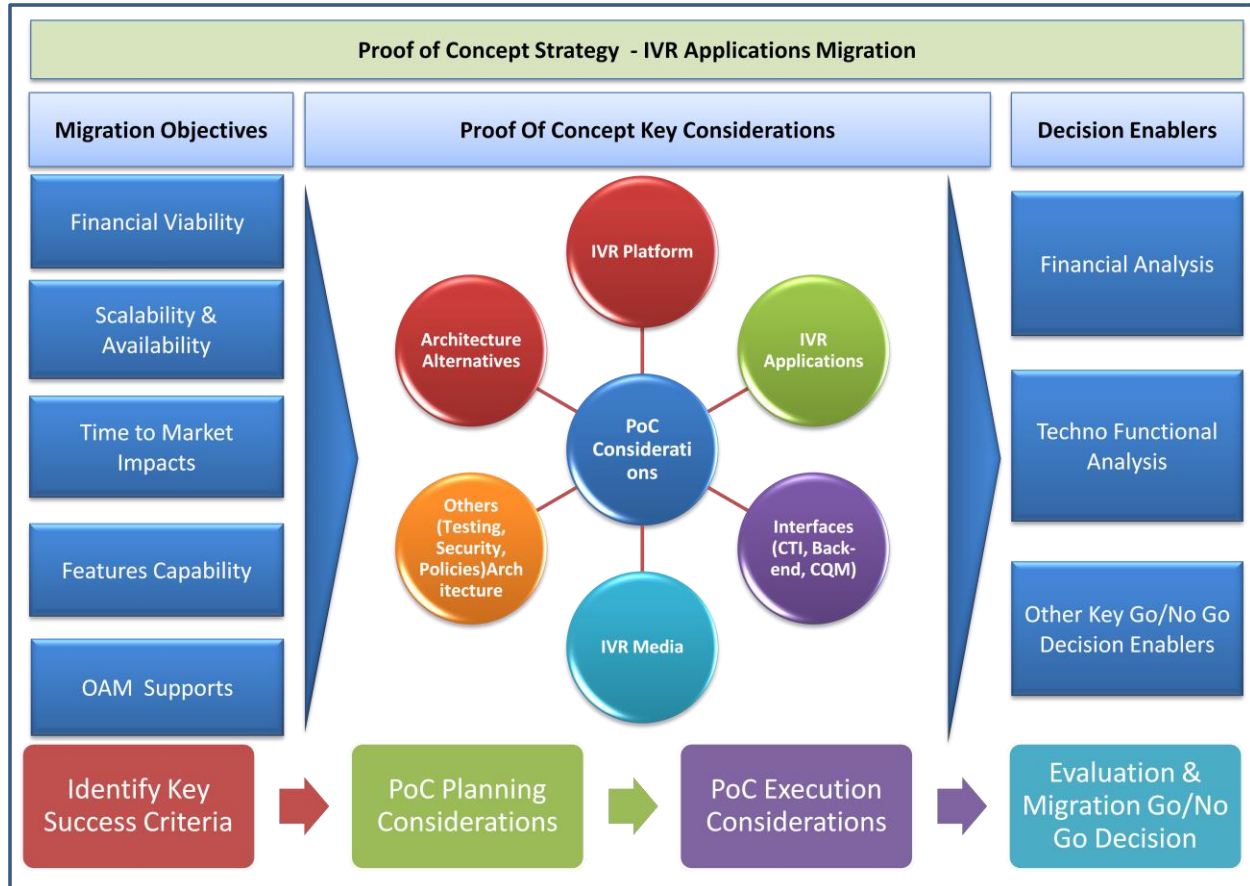
Contemporary IVR's are leveraging enormous advancements in contact center technologies and built on modern self service paradigms. These are truly representative of the brand of a company by seamlessly offering complex self-service features to their customer. In order to stay 'ahead of the curve', companies need to continuously innovate new ways to meet dynamic business demands, improve operational efficiencies & enhance customer experience. In their constant pursuit to achieve these objectives, companies often face tough choices to migrate their existing IVR applications to a new platform.

Investments involved in such large scale platform migrations are huge (~ multi-million dollars) & bear a significant high amount of risk. In order to mitigate the risks, organizations should consider undertaking proof of concept (PoC) approach to uncover unknowns, identify migration strategies and estimate level efforts; before attempting a full blown migration exercises. This White paper outlines end to end process of conducting Proof of Concept to enable organizations make “GO” or “NO GO” decision for critical IVR platform migrations for open source applications.

Proof of Concept Strategy

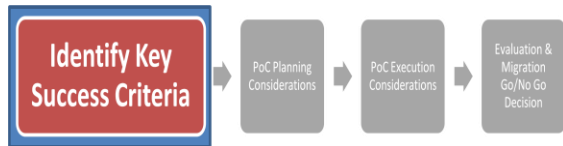
Migrating applications from one platform to another is a huge initiative with full of unknowns. It's prudent to spend some resources

and time to perform POC in order to validate expected benefits and understand possible challenges coming up your way. Following diagram presents an end to end view of a well defined strategy to recognize and implement POC is critical.



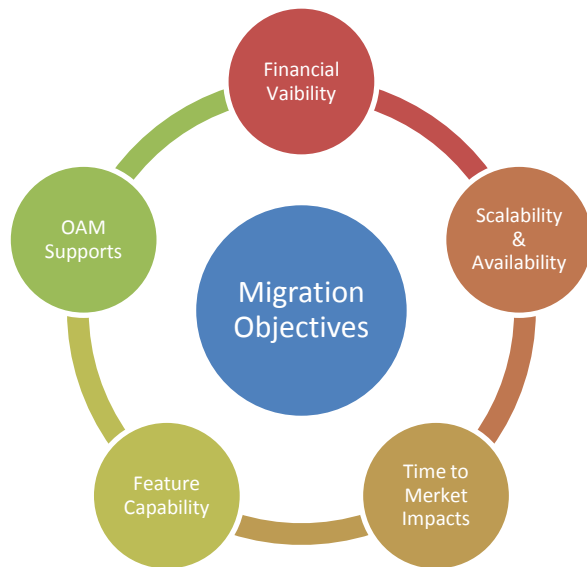
To get the most out of a proof of concept you need careful planning and execution approach. There is a recent trend of moving from premise-based IVR solution to hosted IVR solution. Such migrations pose specific challenges by enforcing policies pertaining to the hosted IVR

environment and often more complicated than a premise based solution. Decision of spending time and money in full scale IVR platform migration must be based on the outcome from the proof of concept. Follow through for the details on the approach in subsequent sections.



The primary and most important factor which determines success of any large scale migration is to understand “What is success” and “What is a failure”.

Identifying key success criteria with executive mandate is essential in undertaking large scale migration projects. Though these factors can entirely vary according to organizational needs, this paper is outlining some of the common parameters which can be used in this regard.



Financial Viability: It is highly important to understand the cost involved viz benefit achieved from such migration. Cost would include Software development (code changes

for migration), Differential Infrastructure cost. Benefits can include less license Fees, No license Fees in case we are migrating to an in-house developed platform and also customer base increment possible due to better services.

Scalability & Availability: Some organizations due to unprecedented and sudden growth in customer base face an issue of IVR application availability and Scalability. In such scenarios understanding the benefits of new platform on these parameters become critical.

Time to Market Impacts: Time to Market is always a critical parameter for success in today’s environment. Long deployment cycles on an IVR platform can be a serious threat to any organization’s marketing strategy. This factor can be a determining factor for success in such migrations.

Features Capability: Some of the IVR platforms do not support VXML, SRGS grammar, Call Recording. Depending on the application and business requirements, this could be important success criteria for application migration

OAM Support: Some IVR platforms have 24*7 supports from the vendors and it is very easy to get over any issue. Depending on application criticality, this can be a determining factor for success of application migration.

Organizations can have one or many of these as a critical success parameter for their application.



Identifying the Scope of the proto-type should be done in a very judicious fashion. Identified Scope must be representative of the complexities inherent to your existing operating environment. Ex: If the applications involve CTI interactions, some common functionality or certain unique functionalities across modules POC must be selected to involve these.

Platform Tech Considerations: In spite of maturing VXML standards and their compliance by the IVR equipment providers, there are still significant differences in the way different IVR platforms behave for same application. POC must be deigned in a fashion to bring out any such differences if there in existing and new IVR platforms. Differences can come in many forms for the building blocks of IVR platforms:

- IVR VXML Browsers (VXML Formats and Standards supported)
- ASR (Support for built-in DTMF Grammars. Speech Grammar formats (BNF, XML etc), Natural Language dialogues)
- TTS support (TTS response to complex phrases)
- Audio Support (Audio prompt format support differences e.g. u law 8 bit or MMF), Audio volumes, quality of audio)
- Platform Supported Protocols (Differences in protocols supported by the platforms (VOIP, PSTN, SOAP (XML/HTTP), TCP/IP, HTTP)

Application Tech Consideration: It's important to consider following technical aspects with respect to during proof of concept.

- Does your existing application uses any application server specific proprietary features (e.g. distributed caching, MBeans in web sphere). If yes, consider moving the dependency to the new platform.
- Caching and DB interaction considerations.
- Does your target platform have the same version of runtime environment (e.g. jvm) &

other software/API. One should never assume that the application would run perfectly if you are moving to a higher version. You are likely to get tons of warnings (if not errors!) which you need to consider removing from the plate.

Interface Considerations: There are multiple external Interfaces to any IVR application. POC Scope must consider these interfaces and ensure critical part of these is tested to avoid problems down the line. Typically you have interfaces such as follows:

Back-end Enterprise Systems Interfaces: Organizations usually have multiple types (e.g. CORBA, MQ, Web services etc) of interfaces to their back-end systems (billing, CRM etc)

- Consider migrating at least one for each type
- Does your target platform and back-end systems behind the same firewall? It's important because you may end up exploring alternative mechanisms to access your back-end systems through newer methods. (e.g. RMI-IIOP has issues with firewalls. Hence it's not recommended. You may need to explore using web services instead)

CTI Interface: It is important to consider following CTI interface specific questions while selecting POC scope

- Does current CTI working fine with the new platform or are changes needed to CTI as well? (e.g. Genesys to CISCO) – If yes you have lot of additional considerations.
- Even if your CTI remains the same, does your target platform allow you to use the existing mechanism to talk to CTI? If not explore your options.
- Consider testing one complete flow for CTI interaction (CTI init, Request/Response, CTI-end)

IVR Media Considerations: PoC should consider moving some of the IVR resources

(prompts and grammar) to assess migration effort and challenges (if any).

Architectural Alternatives: POC must also look at various architecture & migration alternatives. As long as you can commit resources – consider exploring at least 2-3 alternative architectures during proof of concept. Depending on your scenario you may want to consider options such as

- Partial re-write (migrate the IVR application, keeping the front-end IVR same)
- Complete rewrite (Migrate the IVR application and the front-end)
- Re-configuration (can you configure your app on new platform talk to media servers on the existing platform)

Based on the outcomes of the proof of concept It would help you deciding the optimal approach.

Other considerations

Load testing considerations: Determine the testing methodology and approach for the proof of concept. While its challenging to get enough resources (you require) for load testing. It's very important to surface any application/interfaces/platform scalability issues (bottlenecks). Consider doing a 40/80 channel load test depending on your needs. Monitor the resource consumptions on your target infrastructure (IVR, App Server, CTI, MQ, DB), it would help you in capacity planning

Simulating Events/failure scenarios: Consider creating some failure scenarios and generate Alarms to test the failure situations. Many a times in hosted environments have a pre-defined Alarm format. There are differences in the way other events are thrown on the platform as well.

Policy/Security Compliances: Most of the hosted IVR application platforms have well defined

policies and M&P for deploying & testing applications along with logging the errors.

Back-up/Recovery Processes: It's prudent to test the back-up/recovery process on your new platform. Typically hosted platforms impose specific processes/policies for back-up and restoration. While it might look trivial but it has significant impact on your effort to migrate the application because of additional constraints.

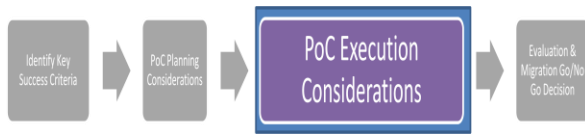
Out of the Box features Support: Determine if your application currently uses any out of the box feature, provided by the hosted IVR platform. These platforms provide advanced features like real-time recording and hot-updates to prompt & grammars, faxing, voice verification capabilities, call management capabilities (conferencing/courtesy calls).

Consideration outlined above should drive you identifying the scope of the proof of concept

Identify the applications for proto-type: Team needs to look at the importance of various applications of IVR. Based on complexity, criticality and maximum possible parameter coverage, one can select the application for prototype. Again decision can vary from scenario to scenario, but parameters discussed above can be used as a generic guideline for selecting application for prototype.

Identify Call flow: Migrating a complete application (for PoC) would not be feasible; you should look at identifying appropriate call flows. Call flow considered for the proof of concept should be true representative of the complexity involved in such a case and should assure maximum coverage of complexities involved in IVR application as mentioned above.

Above mentioned approach would help you planning out the scope of the proof of concept in due consideration to your environment.



Once you are done with planning out the scope of your PoC; go-ahead and execute it.

Line up Resources & execute the PoC

- Identify, allocate resources
- Create environment for the proto-type as close as it can be to your target platform.
- Create a plan and stick to it.

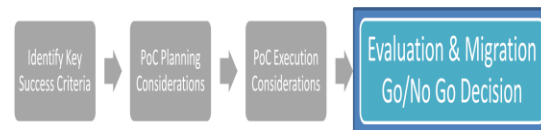
Document key leanings as you go: Identify & categorize and create a list of features with respect to target platform (NOT supported, partially supported, New features, Supported differently). Also list issues, risks and dependencies. Some of the Key things to observe would be:

- If your ASR is changing (as part of migration): Did you notice any apparent degradation in recognition of voice? Document any specific ASR tuning needed to be done as part of PoC to make it work.
- Do you see any difference in the audio levels (overall quality including pitch) of prompts when played on the target IVR?
- Does your TTS able to correctly pronounce some complex phrases correctly (like “”, “&”, “@”, “ ‘ ” etc) Does platform consistently accepting DTMF and speech inputs correctly and consistently.

- Did your system resources continued to be in normal usage limits during load tests? Did you see any slowdown of the response from ASR/TTS when the load increased?

Overall your leanings should be able to help the evaluation phase by providing enough inputs on the challenges seen during execution of PoC from all the perspectives.

Archive for Re-use: Depending on your circumstances you may choose to defer the actual migration of the application. In case you might want to return to the migration few months down the line, you would need to archive the artifacts (code, documents etc) in your repository for future use.



Evaluation is the most critical phase of the entire proof of concept exercise. As it leads to a Go/No Go decision.

Financial Analysis: Identifying and capturing all the cost headings is challenging in large organization. Specially, when these organizations have a mix of outsourced and in-house applications.

Assess As IS and To Be Costs: Determine the total cost of ownership for existing / planned & target platform

Typical Cost Items for Premise based IVR Solution			
PBX Servers	Reporting Solution	Middleware Infrastructure	Others
IVR Servers	Performance Tools	Professional Services (Vendors)	Human Resources
Speech Servers (TTS)	Disaster Recovery Tools	Productivity Enhancement Tools	Support Infrastructure
Voice Recognition Software (ASR)	Database Servers	Analytic Tools	CTI Solution
Application Servers	Licenses	ETL Solution	Miscellaneous

- Calculate the capital cost and operating cost (factor inflation/rising costs)
- Calculate TCO for couple of options 3 year / 5 years.

Alternatively, If you are already running on a hosted solution, typically you pay for what you use, hence the cost computations are rather easier as compared to a premise based solution.

Determine your Migration costs: Migration cost would involve items such as cost of migration of application code, deployment costs, media conversion cost (grammar and prompts) traffic migration costs.

Special considerations – Sunk cost, Cost of continuation (No Migration – if it's optional and just driven by reducing the costs). Once you are done with the assessment of your cost structures (AS-IS and To-BE) you can easily run some What-if simulations to arrive at optimal solution for the migration exercise.

Techno Functional Evaluation: It would revolve around on identifying technical solutions to the challenges surfaced during the proof of concept? You need to come up with mitigation plan for each of the technical issue/risk. Some examples may be:

- The new platform might mandate you to use web services instead of RIMI/IIOP communication mechanism.
- You may need to identify alternate ways to achieve an existing function (e.g. real time prompt updates)
- Clearly understand if something is not supported on new platform and why?
- Recommendation on usage of additional (out of the box/new) features from the target platform to meet your objectives

• Other Key decision enablers

- Project execution considerations (e.g. Transition Schedule, Managing two code threads v/s single code thread)
- Support to ongoing business needs
- Assessment of scalability of the target platform. i.e. Does target platform have the capability to handle your projected call volumes. How difficult/easy it would be add/remove capacity on your new platform.
- Market Leadership (in case of hosted IVR platform)

Summary

Customer satisfaction is a key goal of every organization. The real challenge comes in maintaining balance between customer expectations and investment in business to meet expectations. This is completely true for IVRs. More often than not migrating IVR applications are becomes a necessity for any organization. However, migrating IVR platform has its unique challenges. By using the right approach and proven best practices as illustrated in the paper, companies can avoid pitfalls that can result in financial losses and customer dissatisfaction

For Further Details Contact:

Rajeev Sharma (rajeevk_sharma@infosys.com) / Aman Kakkar (aman_kakkar@infosys.com)