A Complete Guide for Database Technology Migration Program

Database technology migration programs do not come up every year as a high priority initiative in the IT organization. However, carrying out enterprise wide database technology migration programs in an unstructured way can have a catastrophic and expensive impact on the business because once an old system has been retired there is no going back if the new migrated system breaks down.

This paper provides detailed insights and an end-to-end view of the life cycle of the database technology migration program. The author believes that the comprehensive approach provided in this paper will help IT organizations to successfully plan, design, build, test and roll-out a very large and complex database migration program in a systematic, structured fashion.
About the Author

K. Vaidyanathan

K. Vaidyanathan is an IT infrastructure consultant and part of TCS’ Global Consulting Practice - Infrastructure Solutions unit. He has around 15 years’ experience in the IT services and has delivered successful consulting engagements to global customers from various industry verticals such as banking, insurance, telecom, life science and retail. He has architected solutions covering a wide range of areas such as mainframe optimization, mainframe consolidation, mainframe exit, mainframe capacity upgrade, IT Disaster Recovery Strategy and database technology migration planning. He graduated Master of Computer Applications from BMS College of Engineering, Bangalore University.

K Vaidyanathan, GCP Infrastructure Solutions Consulting Group, k.vaidyanathan@tcs.com
Table of Contents

1. Introduction 4

2. Why migrate application systems to another database platform? 4

3. What are the systems suitable for database migration? 5

4. How is a database technology migration program executed? 5

   Stage 1: Assessment 7
   Stage 2: Proof-of-Concept 8
   Stage 3: Business Case 9
   Stage 4: Planning 10
   Stage 5: Design, Build / Migrate and Test 12
       Step I - Infrastructure Setup and Requirements Gathering 12
       Step II – Baseline Testing and Detailed Solution Design 14
       Step III – Database Migration 15
       Step IV – Application Migration 15
       Step V - Unit Testing 15
       Step VI - System Testing 16
       Step VII - Regression Testing 16
       Step VIII – Performance Testing 16
       Step IX – Retrofitting 16
       Step X - User Acceptance Testing 16
   Stage 6: Implementation Planning 17
   Stage 7: Implementation 19
   Stage 8: Post Implementation 20

5. Conclusion 20

6. Abbreviations and Expansions 21
**Introduction**

Global CxOs are focusing on technology refresh initiatives across the enterprise, and in particular, embarking upon database platform migration programs for core / mission critical legacy applications in order to gain competitive advantages and reduce the cost of ownership of IT. However, migration programs consume significant budget and investments belonging to the “Transform the Business (TTB)” initiatives in an organization and so are always prioritized and implemented in a “reactive” mode.

A vanilla database technology migration program may also have the following add-ons for improving the operational efficiency of the target solution:

- Physical database consolidation: Optimizing the number of physical database instances in the target environment
- Component Transformation: Porting of back end components such as scripts from source to target operating system platform

The overall duration of a database platform migration program depends on parameters such as size of databases (number of database objects and size of the objects) and the overall size of the application size (number of components) and other dependent or conflicting initiatives.

**Why migrate application systems to another database platform?**

IT organizations need to assess the technology infrastructure of application systems and inform business at appropriate intervals if there is a significant increase in the total cost of ownership with respect to sustaining the software products (e.g., a database product) and / or if the underlying software product becomes obsolete (e.g., database product is unsupported by vendor). The goals / key themes of CxOs in the IT organization are usually derived out of business goals and drivers for business growth. A few goals pertaining to migration could be:

**Goals / Key Themes: Reduce Operating Expense / Total Cost of Ownership**

- Current version of database platform has reached “end-of-support” and extending further will incur “special support” cost, increasing total cost of ownership
- Annual license cost of the subsequent new / next version of the current database technology has increased significantly, increasing the total cost of ownership
- Aggregated annual license cost of multiple database technology platforms is very high

**Goals / Key Themes: Risk Proof the Business**

- No clear roadmap for enhanced or subsequent versions of database technology product, posing a risk in sustaining business critical system in the “non-strategic” database technology.
- Diminishing market share and product support issues of database vendor inhibiting the enterprise’s ability to invest further in the current legacy database technology infrastructure.
What are the systems suitable for database migration?

Database technology migration is performed for three types of application systems.

1. Developed in-house or custom-built
2. Vendor supplied enterprise solution package
3. Vendor supplied application package

However, vendor supplied application packages may have challenges with database technology migration because:

- Vendor may not recommend, certify or provide support for technology migration
- Ownership of the source code of the application package may be with the vendor itself
Selecting the right candidate application system with an appropriate priority and sequence for database technology migration depends upon the following:

- Criticality of the legacy application system to the business
- Business value of the system
- Size of the system
- Complexity of the system
- Degree of Ownership of the application system

**How is a database technology migration program executed?**

In the top down approach, vision, business objectives, business drivers and business case will drive the IT organization to align and embark on database technology migration program as per the enterprise wide technology roadmap. The structured approach to large migration programs consists of eight key stages in the overall life cycle:

1. Assessment
2. Proof-of-concept
3. Business Case
4. Planning
5. Design, Build / Migrate and Test
6. Implementation Planning
7. Implementation
8. Post Implementation

![Figure 1: Database Technology Migration Lifecycle](image)

The subsequent sections describe the life cycle stages of a database migration program in detail.
Stage 1: Assessment

The Assessment stage broadly consists of 11 activities:


These activities should be performed by a core team comprising technology infrastructure consultant, application architect, database consultant, business and financial consultant. This team will deliver target solution architecture, high level roadmap, Rough Order of Magnitude (ROM) estimates and high level business case for the database technology migration program.
The assessment exercise offers CxOs appropriate inputs to create a convincing business case for the database technology migration program.

**Best Practices**

- Determine quantified benefits for the database migration program to support the financial business case.
- Leverage tools and scripts developed in-house to accelerate and improve productivity during the discovery of component inventory.
- Leverage “Solution Evaluation Framework” to evaluate and select the target solution.
- Leverage existing estimation templates and techniques to arrive at cost and effort for the entire database technology migration program. Use the appropriate productivity definition and calendar holidays during the effort estimation stage for the entire program.
- Obtain cost for the target database solution to prepare business / financial case.

**Stage 2: Proof-of-Concept**

The Proof-of-Concept (PoC) exercise sets the expectations for the entire database technology migration program. The following diagram illustrates the sequence of the activities involved.
The key expected outcomes of the PoC exercise are:

- Finalized technology solution components for the target state such as database platform, operating system platform, server hardware, storage hardware, data migration tools, schema migration tools, data comparison tools, etc.
- Refined approach for the database migration program.

**Best Practices**

- Leverage the underutilized server and storage infrastructure for provisioning and building the PoC environment to avoid upfront capital cost investment
- Connect with database and tools vendors and procure them with trial / limited period licenses for the PoC exercise to optimize procurement cost.
- Document the environment specifics, configuration, deployment lay-out, migration approach, migration techniques and procedures, PoC project results, lessons learnt, technical challenges, and their corresponding contingency actions.
- PoC should address two objectives: prove the functional correctness of the system and to validate the performance SLAs in the target state.

**Stage 3: Business Case**

The key objective of developing a business case is to prove financial / commercial feasibility of the database migration program to the business and executive committee of the enterprise. Financial business case specifies the rationale to proceed further with the initiative post proof-of-concept exercise. This business case document with financial elements such as costs, savings benefits, payback period and return on investment (ROI) is the key enabler for the business to:

- Prioritize and decide on embarking on such large database technology refresh initiative
- Provide “go-ahead” approval for funding from its strategic IT investments / budgets (IT budget allocation for “Transform the Business (TTB)” initiatives).
In IT organizations, strategic IT investments are made to support and improve the business operations directly or indirectly. For arriving at the target state scenario cost, include the capital cost heads such as cost for database migration services, software license cost - database platform, database monitoring tools, database connectivity drivers, database replication tools, operating system, cost for new hardware servers, cost for new storage and cost for training.

**Best Practices**

- To calculate Return on Investment (ROI), include the operational cost for the in-scope systems and exclude the cost for shared infrastructure and services.
- To show comparative benefit in moving towards target database system, include detailed cost elements of the current system such as penalty / additional cost from vendor for license and support for the "out of support" database technology.
- In the cost benefit calculation, include all the projected cost - quantified savings due to reduction of enterprise wide license cost for the new target database platform solution, storage reduction due to compression techniques in the target state databases and consolidation of physical / virtual servers in the target database environment.

**Stage 4: Planning**

Detailed program planning is imperative to avoid undesirable consequences from the database migration program. It is recommended to conduct workshop sessions with the business and IT stakeholders during the initial duration of planning stage to gather more insights. The planning stage consists of 24 major steps:

1. Determine business priority of the program in relation to other dependent initiatives
2. Define the program objectives and scope.
3. Define the phases in the program
4. Arrive at overall duration required for the execution of the program
5. Prepare a detailed program charter containing description of the program, cost summary, schedule summary, effort summary, assumptions and dependencies
6. Determine activities / tasks for the various phases of the program
7. Identify the dependencies between activities / tasks
8. Determine the sequence in which the tasks can be carried out to traverse through absolute critical path
9. Prepare a detailed work break down structure containing activity, start date, end date, dependencies between activities / tasks
10. Prepare a plan for provisioning infrastructure facilities for the program such as server devices, storage devices, development tools, testing tools, operating system, database technology, database tools, data migration tools, data comparison tools, etc.
11. Develop a detailed resource loading plan across the various stages of the program

12. Perform a comprehensive risk assessment and identify risks. Determine and document the risk category, risk exposure, risk impact to the program and likelihood of occurrence of the risks. Come up with well planned contingency actions to mitigate/avoid the risks. Risk categories could be business impact risk, solution architecture risk, data migration risk, data quality risk and performance risk.

13. Define communication and escalation plan for the program

14. Develop and document quality assurance plan prescribing quality standards, quality assurance approach, quality assurance timeline, quality assurance team structure, etc.

15. Design key project service metrics such as deliver on time, meeting schedule, deliver right the first time, etc., to track and assess the overall health of the program and determine outliers

16. Define governance structure with hierarchy, roles and responsibilities of various teams

17. Document the change and configuration management processes to be followed for the entire migration program

18. Freeze the technology decisions for the various stages of the entire migration program life cycle

19. Prepare an elaborate procurement plan for facilitating the technology and infrastructure components

20. Prepare, publish and agree on support and service delivery SLAs for various teams involved in the program such as vendor, infrastructure, business service team, project team, etc.

21. Prepare a detailed RASCI (Responsible, Accountable, Supported by, Consulted and Informed) matrix for the activities involved in various stages of the program

22. Define registers for logging, tracking and closing issues, risks, decisions, etc.

23. Develop documentation standards and templates for artifacts such as database design documents, database migration document, application remediation / porting document, test plans, test cases, status reporting, checklists, etc.

24. Agree and establish service level agreements (SLAs) with the product and tool vendors to fix issues that may arise during the program

**Best Practices**

- Create a collaborative project workspace for the team and create a hierarchical folder structure in the internal knowledge sharing repository system for the artifacts of the entire database migration program
- Identify and exclude all the obsolete / unused / stale objects such as tables, views, indexes, stored procedures, etc., from the scope during the inventory collection exercise
- Plan detailed knowledge transition sessions for understanding the IT systems that are in-scope for migration. This will help the project team and the testing team to design, build and test in a more efficient and effective way.
Stage 5: Design, Build / Migrate and Test

This stage has ten steps, with some significant milestones in the overall database technology migration program. The steps are: I) Infrastructure Setup and Requirements Gathering II) Baseline Testing and Detailed Solution Design III) Database Migration IV) Application Remediation / Porting / Migration V) Unit Testing VI) System and Integration Testing VII) Regression Testing VIII) Performance Testing IX) Retrofitting X) User Acceptance Testing. “Initiate and implement one change at a time” should be the guiding principle to ensure successful delivery of the migrated system to the business.

Figure 4: Design, Build/Migrate and Test Stage

Step I - Infrastructure Setup and Requirements Gathering
- Baseline the inventory of application components and database objects
- Gather non-functional requirements (performance, availability, scalability)
- Plan and provide the infrastructure (say, desktop, virtual desktop, thin client, development tools, testing tools connectivity drivers etc.) facilities to the project team
- Plan for the capacity and build the infrastructure for all the environments
- Provide access to the project team and testing team to perform baseline testing
Figure 5: Technology Infrastructure Planning and Build

- **Plan Capacity**
  - Plan
  - Decide / Finalize
  - Procure

A1 Technology Infrastructure Planning

A Technology Infrastructure Build

A2 Install → Configure → Build
- Baseline Environment
- Development Environment
- System Integration & Regression Testing Environment
- Performance Testing Environment
- User Acceptance Testing Environment
- Production Environment
- Disaster Recovery Environment

Prior to the start of
- Baseline Testing
- Development Migration
- System Testing
- Performance Testing
- Upper Acceptance Testing
- Production Roll-out
- DR Deployment

Figure 6: Project Infrastructure Readiness

A2 Project Infrastructure Readiness

- Plan
- Decide
- Deploy

- Virtual Desktops / Workstations
- Developer Environment
- Developer Productivity
- Communication Infrastructure
- Access to the Components
- Connectivity Drivers
Step II – Baseline Testing and Detailed Solution Design

- Perform functional and performance baseline testing of the application
- Define detailed solution design covering the database layout, database security, data replication / synchronization, disaster recovery, configuration layout and setup, and equivalents for the unsupported database features of the target platform
- Define detailed approach and methodology for porting / converting / remediating application components, migrating stored procedures and triggers, migrating schema and data, comparing data for verification and validation
- Document production roll-out / cut-over / go-live process in detail
- Analyze and group the existing application interface and report generation components based on technology stack, business functionality, etc.
- Analyze and group the database objects such as stored procedures and triggers based on the dependencies and SQL built in operation performed in unsupported column data types
- Analyze other database objects such as unsupported column data types, data compression, and internal data storage representation aspect such as Unicode impact and code page specifics of data, etc.
- Perform proof of concept to evaluate and select the tools / utilities to perform stored procedure migration, SQL migration, data object migration, schema migration, data masking / scrambling, data validation, data extraction / transformation / loading, performance testing and load testing.

**Figure 7: Tools Deployment**

- **C1** Tools Deployment
  - Evaluate
  - Select
  - PoC/Pilot
  - Decide
  - Procure
  - Deploy

- **C2** In-house Tools Deployment
  - Design
  - Build
  - Deploy

- **Tools**
  - Schema Migration
  - Objects Migration
  - Data Migration
  - Data Comparison
  - SQL Migration
  - Extract Transform Load
  - Performance Testing
  - Load Testing
  - Validation
  - Automation
  - Data Scrambling
  - Procure
  - Decide
  - PoC/Pilot
  - Select
  - Evaluate
  - Build
  - Deploy
  - Design

**Tools Deployment**

- **In-house Tools Deployment**
  - Design
  - Build
  - Deploy

**Tools**

- Schema Migration
- Objects Migration
- Data Migration
- Data Comparison
- SQL Migration
- Extract Transform Load
- Performance Testing
- Load Testing
- Validation
- Automation
- Data Scrambling
- Procure
- Decide
- PoC/Pilot
- Select
- Evaluate
- Build
- Deploy
- Design
Step III – Database Migration

- Configure tools and build the required scripts and utilities for the database migration tasks
- Execute the database migration tasks in the following sequence:
  1. Database schema migration - migration of database objects such as databases, table spaces, tables, views, indices and constraints (primary key, foreign key, unique key, etc.)
  2. Migration of table structure with equivalent data types in the target database platform for the unsupported column data types
  3. Database security setup migration
  4. Data migration
  5. Data reconciliation
  6. Data comparison and validation
  7. Database housekeeping processes and scripts migration (backup, defrag, etc.)

At the end of database migration step, the project team will have a full-proof tested processes, procedure and package build for the entire database migration program.

Step IV – Application Migration

- Configure tools and build the scripts and procedures for application remediation.
- Group the applications based on business functions and execute the application remediation / porting tasks in the recommended sequence:
  1. Migration of stored procedures, triggers, business user queries and reports
  2. Remediation of front end application components (languages such as COBOL, C#, C++, C, Pro C, Java, VB, ASP, JSP etc.)
  3. Porting of back end application components (developed in script languages such as perl, shell etc.)
  4. Remediation of report components (developed using products such as crystal reports, cognos, business objects, etc.)
  5. Remediation of interface application components.

Step V - Unit Testing

Unit testing ensures that the individual application components (stored procedures, triggers, front end application components, back end application components, report components, interface components, queries) are working as expected post remediation / conversion / porting / migration. Unit testing will usually be performed by the development migration team to detect and fix the defects during the component migration. The key documentation artifacts are test cases, test scenarios, test plan, and the testing approach that drives the entire unit testing exercise.
Step VI - System Testing

System testing is carried out to verify and certify the end-to-end business functionality of the migrated IT system. In addition to this, the interface systems will also be tested to prove that the entire IT system works in a coherent fashion and produces the expected results post conversion / migration. Automated testing tools and scripts can be leveraged to run the system test cycle automatically and compare the results of the test.

Step VII - Regression Testing

Regression testing is carried out using the baseline test cases for the migrated / converted / ported / remediated application components. It compares the results with the baseline testing results to

- Ensure the functional correctness of the components
- Detect and fix the defects in the components if there is any deviation

Step VIII – Performance Testing

Performance testing is carried out to certify that the performance of the target state application is either equivalent to the current situation or exceeds the defined goals. Performance testing tools accelerate the performance testing exercise. It is also important to select the right candidates such as business critical transactions and batch jobs for the scope of the performance testing exercise. Performance testing needs to be conducted in a dedicated environment which should mimic the production environment in terms of environment setup, data volume, server configuration, concurrent users, etc., to obtain the perfect performance behavior of the migrated target system.

Step IX – Retrofitting

Retrofitting is the key step in applying functional changes to the migrated application components and schema changes to the migrated database objects (component inventory baseline identified prior to analysis and design phase) which have already been implemented post baselining exercise. It is important to have a freeze period for the functional releases during retrofitting phase so that all the changes are consolidated and applied in the migrated components at one go. The testing exercise needs to be carried out to certify the functional correctness of the migrated components post implementation of changes.

Step X - User Acceptance Testing

User Acceptance Testing (UAT) is executed to validate the functionality of end-to-end business process, system transactions and business / IT user access to the migrated application system. The key focus of UAT is to identify and provide fixes for areas where business user requirements are incorrectly implemented in the system. The objective of the UAT is to obtain sign off from the business users, indicating that the delivered / migrated application system with new database technology meets requirements and is fit for production cut-over.
In addition to the listed testing steps, other types of testing which will be conducted during quality certification process are: security testing (to ensure application security of the delivered application system is preserved and maintained in the target environment), single point of failure testing and high availability testing.

### Best Practices

- Plan the infrastructure demand (storage, server and network link, for example) and procure/provision in the “Just-In-time (JIT)” fashion or reuse existing unutilized server and storage capacity as and when the demand/requirement arises to avoid and defer the delays in delivering the various environments for the program.
- Plan and include the business service delivery team during the planning and execution stage of user acceptance testing phase to ensure the delivered application system behaves and works as expected.
- Consider using utilities/scripts developed in-house to mask the critical data contents during the development and testing phases to comply with the data security policies.

### Stage 6: Implementation Planning

During implementation planning stage, it is essential to understand the:

- Current application component and database object deployment processes
- Application and database system availability requirements and availability SLAs for the business
- Pre-defined holiday calendar for the business
- Existing functional release schedule planned for other dependent projects/initiatives
- Seasonal workload processing schedule such as month-end, quarterly, etc.
- The tangible and quantified business impact due to unplanned outage events in the past
- Recovery time objectives and recovery point objectives for the application system.

Also, the following activities need to be performed in this stage:

- Conduct workshops with Business Sponsors and IT stakeholders to determine the appropriate roll-out/cut-over strategy
- Devise a detailed roll-out/cut-over strategy (big bang/straight cut-over, parallel cut-over, serial/phased/incremental cut-over, etc.) based on considerations such as scheduling, downtime, business impact, business risk, approval, technical architecture, etc.
- Agree upon and obtain approval for the roll-out/cut-over strategy from stakeholders
- Define the pre-requisites for the production “roll-out/cut-over”
- Identify the activities that can be carried out in pre roll-out and actual roll-out window
- Break down the large deployment/roll-out process effort into smaller and manageable tasks/activities in order to fix the expected issues immediately and proceed further
Plan and document detailed activity plan and schedule for the production roll-out / cut-over procedures

Determine fall back levels and define appropriate fall back actions and procedures

Plan and document activity plan for the fall back / roll-back processes and procedures

Build roll-out package with a set / bundle of scripts which will be executed in pre-defined sequence for the database migration.

Prepare deployment instructions and run book schedule for the deployment team for deploying the migrated application components, interface components, report components, database objects, etc., in the target production environment.

Conduct multiple iterations of production dry run exercises in order to:
  - Determine the duration required for the actual production roll-out / cut-over
  - Identify defects and fine-tune the roll-out processes
  - Determine the optimal number of parallel threads / tasks for carrying out activities such as data migration in the actual roll-out window

Conduct fall-back / roll-back dry run exercise at least once to
  - Ensure that there is a tested fall-back procedure available in case of major failure post go-live of the new migrated application system
  - Understand the fall back duration before roll-out / cut-over / go-live
  - Diagnose defects, rectify and improve the fall-back procedure and process

**Best Practices**

- Perform the non-impact action items such as deployment of infrastructure, migration of database schema, database structure, database security, application component deployment, interface component deployment, during the pre roll-out window to ensure effective and optimal usage of roll-out / cut-over window and schedule
- Always use source data from a production like environment (e.g., Disaster Recovery environment if available) during the production dry run exercise.
- Plan and communicate effort requirement from business service delivery team for the business check-out exercise post production roll-out / cut-over to enable “go-live/ no go-live” decisions
- Schedule the data migration tasks in parallel for both the target production database environment and the disaster recovery database environment (if any) during the roll-out / cut-over window as long as there is no dependency between them.
Stage 7: Implementation

During implementation, the following activities should be performed:

- Execute the pre-roll-out activities
- Complete the pre-requisites of production roll-out: building the production infrastructure (server, storage, etc.), halting the current production batch processing schedule, disconnecting the users from the production environment, taking a complete backup of databases from the existing production database environment, etc.
- Execute the roll-out activities
- Perform the check-out exercise to verify the correctness of key critical transaction outputs (like read only transactions) and business transaction outputs
- Decide collectively and publish “Go-Live / No Go-Live” decision for the new migrated application system and databases
- In case of failure and a “No Go-Live” decision, initiate appropriate level of fall-back
- In case of success of roll-out / cut-over, publish the “Go-Live” decision and release the new application system to the user community for the business operations
- Document the challenges and contingency actions of the production roll-out / cut-over

**Best Practices**

- Deploy a dedicated / core technical helpdesk team to address the issues that arise during pre roll out and roll out activities
- Develop roles and responsibilities for various organizations involved in the production roll-out / cut-over exercise and publish the same to the entire team
- During roll-out / cut-over window, divide the entire production package scripts into manageable sub-units of tasks and manage the sequence; this allows the team to fix issues immediately instead of waiting till the end of the data migration process, which might cause delays and an extension of the roll-out window
- Establish a dedicated support channel and mechanism for the technology vendors (database, tools etc.) to assist and provide immediate fix for addressing the problems that may arise during the implementation and the post-implementation stages
- Always create database objects by dropping the constraints such as referential integrity, check, index, unique constraints, etc., to accelerate and parallelize the core data migration tasks during roll-out. Post data migration, carry out tasks such as index creation, creation of referential integrity constraints, etc., to optimize roll-out window.
Stage 8: Post Implementation

Following are the critical activities for the post implementation stage.

- Monitor the functional correctness, stability, performance and availability of the migrated application system and resolve issues, if any, post go-live.
- Document the issues and resolution actions in detail.
- Publish a detailed training plan and impart training to the IT and business service team.
- Publish a plan for decommissioning the old database server and storage infrastructure.
- Archive the contents of the old databases.
- Uninstall the old database server instances and older version of connectivity drivers.
- Release the old database hardware servers and storage.
- Track the tangible benefits and quality metrics for the migrated application system.

Best Practices

- In case of phased or parallel cut-over, deploy data synchronization solutions in order to synchronize data between new / target and old / source database systems for a pre-defined period of time. This will help to fall-back to the old database systems in case of major failure with the new migrated IT system and database system.

Best Practices

- Plan sessions with the technology solution vendors during the training exercise to enable the user communities to work smoothly with the new database technology platform.

Conclusion

In any enterprise, successful delivery of a database technology migration program will help in achieving Total Cost of Ownership (TCO) optimization goals and risk proofing goals of IT / Business organization. This paper has elaborated the fact that having a structured and planned approach to the entire database migration program in combination with right set of elements such as people, skill sets, processes, best practices and technology solutions will help the CxOs to manage and deliver the expected results and benefits.
## Abbreviations and Expansions

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASP</td>
<td>Active Server Pages</td>
</tr>
<tr>
<td>COBOL</td>
<td>Common Business Object Language</td>
</tr>
<tr>
<td>CxO</td>
<td>Chief Information Officer, Chief Technology Officer, Chief Financial Officer</td>
</tr>
<tr>
<td>DBA</td>
<td>Database Administrator</td>
</tr>
<tr>
<td>DR</td>
<td>Disaster Recovery</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>JSP</td>
<td>Java Server Pages</td>
</tr>
<tr>
<td>PoC</td>
<td>Proof-of-Concept</td>
</tr>
<tr>
<td>ROI</td>
<td>Return on Investment</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>TCO</td>
<td>Total Cost of Ownership</td>
</tr>
<tr>
<td>TTB</td>
<td>Transform the Business</td>
</tr>
<tr>
<td>VB</td>
<td>Visual Basic</td>
</tr>
</tbody>
</table>
About TCS' Global Consulting Practice

TCS' Global Consulting Practice (GCP) is a key component in how TCS delivers additional value to clients. Using our collective industry insight, technology expertise, and consulting know-how, we partner with enterprises worldwide to deliver integrated end-to-end IT enabled business transformation services. By tapping our worldwide pool of resources - onsite, offshore and nearshore, our high caliber consultants leverage solution accelerators and practice capabilities, balanced with our knowledge of local market demands, to enable enterprises to effectively meet their business goals.

GCP spearheads TCS' consulting capacity with consultants located in North America, UK, Europe, Asia Pacific, India, Ibero-America and Australia.

Contact
For more information about TCS' consulting services, email us at global.consulting@tcs.com, or visit www.tcs.com/consulting

Subscribe to TCS White Papers
Feedburner: http://feeds2.feedburner.com/tcswhitepapers

About Tata Consultancy Services (TCS)
Tata Consultancy Services is an IT services, consulting and business solutions organization that delivers real results to global business, ensuring a level of certainty no other firm can match. TCS offers a consulting-led, integrated portfolio of IT and IT-enabled infrastructure, engineering and assurance services. This is delivered through its unique Global Network Delivery Model™, recognized as the benchmark of excellence in software development. A part of the Tata Group, India’s largest industrial conglomerate, TCS has a global footprint and is listed on the National Stock Exchange and Bombay Stock Exchange in India.

For more information, visit us at www.tcs.com

IT Services
Business Solutions
Outsourcing