



Multiple Country Implementation – Influencing Factors and Approaches

Executive Summary

Micro- and macroeconomic global challenges have forced large multinational companies to embrace a model where infrastructure and resources can be standardized and reused across various business units, in many cases spanning multiple countries. The goal: standardize processes and bring efficiency to resource utilization.

This approach has had significant impact on the way IT systems are conceptualized, developed and maintained. In today's large enterprises with operations that span various countries or regions, companies need applications that singularly address the needs of different business units in various countries. Organizations adopt this approach to control costs, standardize operations and practices, consolidate hardware and software and, last but not least, apply common governance practices. Insurance is an industry where we see many commonalities in the core business processes followed across various regions/countries. As such, there is a high degree of interest in maximizing the commonality and reusability of many IT applications among large multinational insurance companies.

Most global organizations are trying to achieve this by migrating to new technologies or by developing new applications or product implementations as a program for multiple business units (BUs) spread across countries or regions. This, they hope, will help them achieve the aforementioned goals of creating standard business processes and, to

a large extent, consolidate disparate software and hardware resources across multiple BUs. The regional programs are often run by a single overarching team, commonly known as a "regional team."

As large organizations migrate to a model where business processes, infrastructure and resources can be standardized and reused across various BUs, they must also recognize a multitude of factors that influence the foundational elements of this approach. Many of these factors are unlike traditional green field projects and are normally not seen in any single-country implementation.

This paper examines the key factors which influence a multi-country solution development and implementation, along with the industry best practices and their key features.

Key Factors Influencing Process and Systems Standardization

Most core implementations are large, challenging transformation projects. In this section we analyze the factors and approaches influencing the implementation of a standardized solution and group them in three broad directions.

- **Development:**

- These factors influence the development process of a regional application. There are multiple ways the development of a regional application can be approached; choosing the right one is the key to the success of the program.

- **Implementation:**
 - These factors influence the best way to implement the software and maintain/enhance the system on a continuous basis, considering various country-specific requirements and keeping language in mind.
- **Budgeting:**
 - These are the factors that influence the creation of a win-win “financial model” for all country businesses and the regional unit.

Development: Key Influencing Factors

These are factors which we must understand and react to before we conceptualize the business application and begin development.

These factors are mainly related to requirement management, architecture definition, development, governance, etc.

As country-specific customization is performed on a common codebase, regression testing becomes extremely important. In many cases, having an automated regression bed is very useful to reduce implementation time as well as enable more effective testing.

1. **Governance:** This is a key factor, as different BUs may have different ways of working and systems in place. A model with representation from both the regional as well as BU team is mandatory, along with transparent and well-defined roles and responsibilities. Depending upon the approach adopted for the multi-country development, governance may be either regional-centric or federated within the BUs.
2. **Differing local business practices – requirement management:** BUs may have different business practices to satisfy a particular business need, hence it is of extreme importance that the regional team initiates and oversees the gathering of requirements from all the BUs, either at once or in sequence depending upon the development approach adopted. There should be a requirement control board comprised of senior business analysts and architects who will make the final call on which requirements can be classified as regional and which are BU-specific.
3. **Integration with BU applications:** Different BUs may have differing legacy systems as part of existing architecture landscapes. Inter-

facing with legacy sub-systems may also be different. A standard data exchange format (mostly XML-based) should be decided upon and shared with all BU teams well before the interface development starts.

4. **Reusability and extensibility in architecture:** The intent of a multi-country implementation is to achieve a high level of reusability and extensibility, in terms of both technology building blocks and functional systems. To gain maximum reusability, a flexible application architecture that enables various features such as language compatibility, feature on/off, database call abstraction, etc. is a must. That is why the architecture of the core application is very important in order to ensure the extent of reusability across various BUs.
5. **Testing optimization:** As the multi-country implementation requires multiple rollouts in various BUs in a short span of time, having an optimized testing strategy plays a crucial role. As country-specific customization is performed on a common codebase, regression testing becomes extremely important. In many cases, having an automated regression bed is very useful to reduce implementation time as well as enable more effective testing.

In addition to the above, several other factors and considerations are made during the development phase. They include:

6. **Legal and regulatory differences:** This typically arises out of differing and sometimes conflicting laws and financial practices across various BUs. There may be a restriction on hosting data outside the borders of a BU. Cross-border payments across BUs may need approval or may not be allowed at all.
7. **Geographical and cultural differences:** This includes support for local languages and local practices. The key requirement of global implementations is the deployment of a core regional team in different BUs. The collaboration among teams, and their timely presence, is crucial for a successful implementation.

Along with these factors, choosing the right execution model is equally important. This paper presents the common industry practices or models for multi-country implementation.

Development: Approaches and Models

The most important part of any multi-country build and development program is the approach

taken to define the solution architecture and the way it is shared across BUs. These can be of the following two variants.

1. **The Core and Country-Specific Model:** Core is developed by extracting the common business requirements from all affected BUs.
 - > Start with requirement gathering for all BUs. A thorough requirement analysis is carried out to determine the requirements from all the BUs.
 - > Abstract the common requirements and define the boundary for the core and segregated BU-specific requirements. After a detailed analysis is performed with inputs gathered from different BU stakeholders, a common requirement set is developed.
 - > Develop core as a common application based on the common requirements.
 - > Developed primarily as part of a regional initiative, these system requirements are tested as an individual project/product. This development of core application is run and managed by the regional team.
 - > BU-specific layers are built on top of core, separately.
 - > Plug in country-specific functionalities or changes to ensure alignment with the overall solution road map and follow the same principles.

- > Individual business unit applications are tested separately as per local requirements.
- > The purity of the core application is maintained during the BU implementation. Any change in the core application is made only after a thorough analysis of the new requirement and after ensuring the suitability of that requirement in the common application. This must be approved by the regional team and added to the core by the same team.

The pluses and minuses of this approach are as follows.

Pros:

- More standardized towards a common business and technical solution.
- Less effort required during BU implementation.
- Controlled and effective governance.
- Subsequent BU implementation may happen in parallel.

Cons:

- Initial investment is high.
- ROI is realized very late in the process as most of the initial time is spent in gathering requirements for all BUs and then building the core.
- Risk is higher as actual BU implementation is performed late.

Core and Country-Specific Model

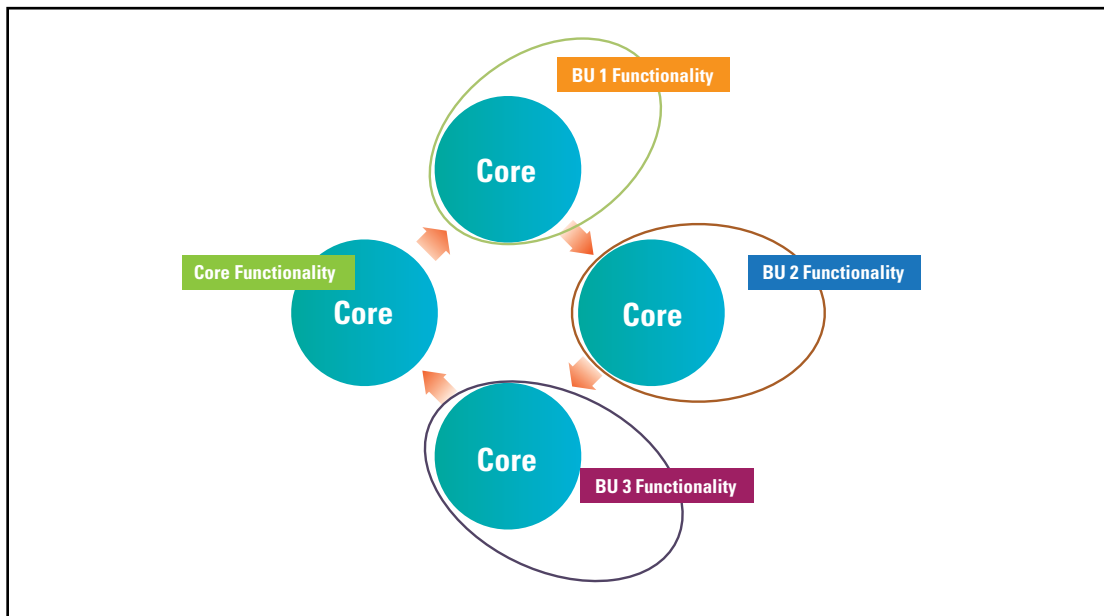


Figure 1

2. **The Base and Customize Model:** One BU is selected for the first implementation and the end-solution is considered and standardized as the base to be implemented for subsequent BUs.

- Select the first two BUs to get near full functional coverage.
- Start with requirement gathering for the first BU.
- Design and develop the application for maximized reusability and extensibility.
- Test and implement the application as per first BU requirements.
- Share the codebase with the next BU; that BU then builds its specific requirements on top of a shared codebase.
- Codebase will expand depending on the nature of the functional requirements encountered during the first two or three BU implementations.
- After two to three BU implementations, cut off codebase development and baseline it as the regional codebase (though it is recommended to cut off the codebase as regional codebase after the first two BUs).
- Share regional codebase as executables with subsequent BUs for extension.
- Changes and additions should be analyzed and approved for induction to the regional codebase.
- The regional codebase is managed regionally as a single repository. This helps the movement of this codebase to the subse-

quent BU and controls the changes effectively.

- Configuration management is done using a centrally managed repository with the trunk representing the regional codebase and the interim BU development written on the branches.
- Each country should have stand-alone instance with selected or all features from the regional codebase.

The pluses and minuses of this approach are as follows.

Pros:

- The ROI for the first BU is achieved faster than the previous model as a working solution.
- Minimum code changes to speed up rollout and reduce risk.
- Less risky, as BU implementations are sequential.
- Initial investment is low.

Cons:

- Complex governance after the first implementation.
- Greater chance of deviating from a standardized business and technical solution.
- Due to sequential implementation pattern, time taken to implement all the BUs is greater.
- Complex configuration management.

Implementation: Key Influencing Factors

Similar to development, there are multiple factors which influence a multi-country implementation during the post-development phase, and based on

Base and Custom Model

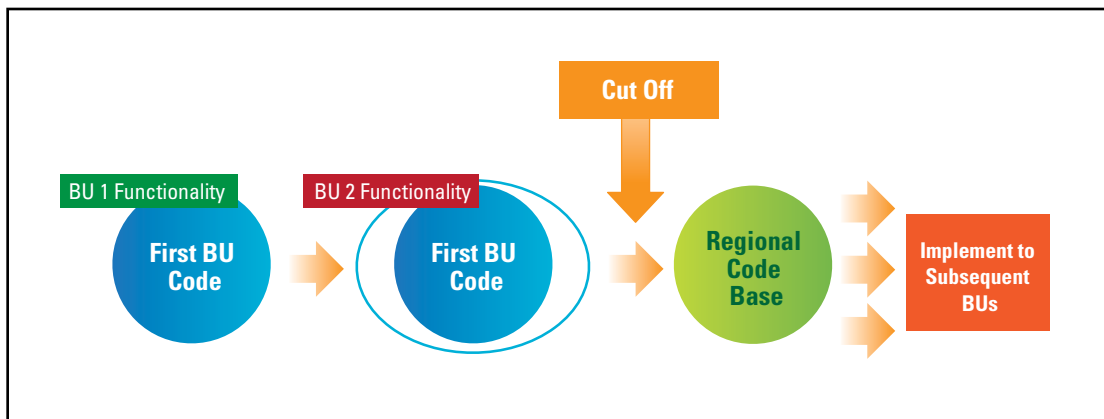


Figure 2

those factors it is crucial to adopt to any one of the industry practiced models. These are factors mainly related to the post-development and implementation phase. These include:

1. **Governance:** This is a key influencing factor even during the implementation phase; maintenance and enhancements of the developed solution may be either done by the regional team or by the BU team depending upon the approach adopted for implementation.
2. **Hardware/software/infrastructure and connectivity:** The procurement of hardware and software should be conducted either regionally or locally. Connectivity is an important factor for consideration and highly influences the model adopted.
3. **Change and release management:** Based on the development approach, the change and release management can be done centrally in the regional center or locally by individual BUs in accordance with an organization-stated standard process. Any change in the core application or core codebase needs to be handled extremely cautiously as it will impact all the instances.
4. **Hosting and disaster recovery environment:** This may happen either regionally or locally and varies depending upon factors that lead to adoption of any one of the implementation approaches or models.

5. **Different regulations and data privacy:** Legal and regulatory challenges arise out of differing and sometimes conflicting laws and financial practices across countries. There may be restrictions on hosting data outside the borders of a country. Cross-border payments across countries may need approval or may not be allowed at all.
6. **Multilingual support:** A support team consisting of multilingual people is required to support the regional application. Various challenges need to be overcome (as a result of geographical and cultural differences), including support of local languages and local practices.
7. **Configuration management:** Configuration management and code promotion is complex for a regional implementation as compared with local instance deployments. This complexity varies largely depending on the architecture of the application and development methodology.

Based on their enterprise constraints and working model, organizations may be influenced by either all or any number of the aforementioned factors. Based on their degree of influence and priorities, organizations should adopt any one of the proven approaches or models as listed below.

Implementation: Approaches and Models

There are two primary ways in which a multi-country solution can be implemented and managed.

1. **Regional Implementation:**
 - Deployment of a single instance accessed by multiple BUs (mostly separate instances of the same solution for each BU, hosted regionally).
 - Configuration management and code promotion handled regionally.
 - Deployment of the complete solution to a specific location (mostly the regional center) and accessed via network by different BU users.

The pluses and minuses of this approach include the following.

Pros:

- Provides enhancement of reliability, a reduction of hardware acquisition costs, an acceleration of movement towards standards-based servers and application consolidation and the leveraging of existing development skills.
- Better regional control.

Regional Implementation

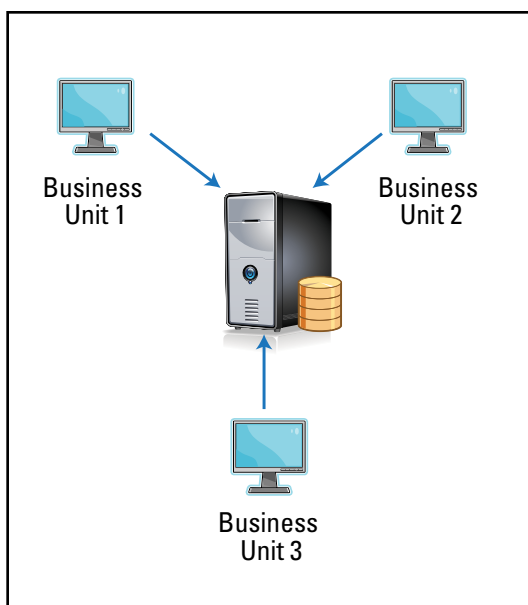


Figure 3

Cons:

- Requires the availability of hardware and software at the regional level only, but has a risk of availability and disaster recovery as all BUs will be dependent on a single infrastructure.
- The varied country-specific rules and regulations binding data portability across country boundaries can pose a risk to this approach.
- Performance is a key concern.

2. Local Implementation:

- Deployment of the same core solution to individual BUs as stand-alone instances.
- Requires the availability of hardware and software in individual BUs.
- Configuration management and code promotion is handled by BUs.

The pluses and minuses of this approach include the following.

Pros:

- Federated control of hardware and software locally.
- Existing peripheral devices can be used without much change.
- Reduced risk of disaster recovery as all BUs may operate separately.
- Control of changes and releases management is quite simple and done by local teams.
- Better performance than a regional model.

Cons:

- Hardware and software procurement costs may be redundant and high.

- Infrastructure consolidation is not achievable, resulting in high maintenance and running costs.

Budgeting: Key Influencing Factors and Approaches

Structuring the program cost between the BUs and the region is often a difficult task, and can affect the program outcome significantly. As different countries/regions will have varying degrees of reliance and importance placed on these shared applications, and also as the customer base and business volume will not be the same for different business units, it is important to build a financial model which is proportionate and justifiable to all the stakeholders. In most of the cases, the businesses for different BUs vary in size and complexity, so the cost allocations to build applications for different BUs need to be handled in a very reasonable and equitable manner. There is no easy solution for this challenge, but typically it is handled in one of the following two ways.

1. Sharing the cost for each implementation:

Typically for these types of program, a regional budget is allocated along with a budget for each BU. The implementation cost for any BU is shared by the regional and BU teams. In this method, each BU implementation is taken as a separate project and cost is shared between BU and region.

2. Core and customization cost:

In this model, the regional team sponsors the development cost for the core application or codebase (depending on the development methodology). Here the initial core application development cost is completely borne by the regional team, then the BU implementation and BU-specific

Local Implementation

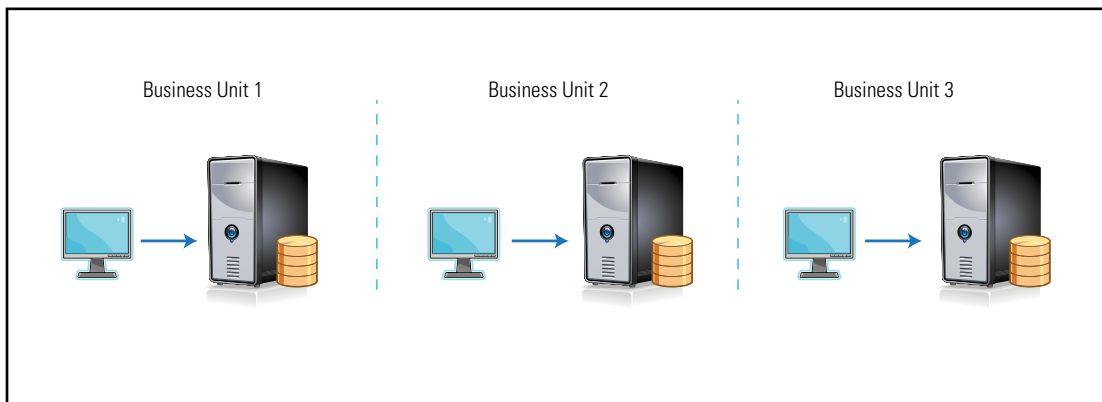


Figure 4

customization cost is left with the individual BUs. In this method, the BU has the liberty to customize (with BU-specific functionalities) the application based on its need, budget and business criticality.

Conclusion

Regional implementation of various applications has become a necessity for most large organizations doing business in multiple countries across BUs within a single region. This approach is becoming increasingly relevant in today's increasingly interconnected global economy, where organizations deploy varying methods and technologies to achieve maximum efficiency from a single initiative. Different approaches will make sense to different organizations based on their business situation and future road map.

This paper has acquainted the reader with key influences and business factors based on our real-world experiences. One key takeaway from

our vantage point: It is extremely important to have a proper due diligence phase, to first understand necessary organizational behaviors and weigh the applicability of key influencing factors. Based on the influencing factors applicable for the organization, a suitable model or approach for multi-country implementation can be adopted. Once an organization adopts the right development and implementation approach, the subsequent processes fall in place with little or no difficulty. On the other hand, without giving due considerations to the key influencing factors organizations may run into thorny problems during the development and implementation phases.

It is extremely important to have a proper due diligence phase, to first understand necessary organizational behaviors and weigh the applicability of the key influencing factors.

About the Authors

Goutam Kanjilal is a Cognizant Client Partner working with insurance industry customers across Asia. A 10-year Cognizant veteran, Goutam has worked on numerous large consulting and strategic development projects with multiple clients worldwide. His primary responsibility is to provide strategic solutions and build strong executive-level relationships with insurance clients throughout Asia. Goutam has a bachelor of engineering degree from Calcutta University (India) and holds various certifications on project management, technical and insurance-domain related subjects. He can be reached at Goutam.Kanjilal@cognizant.com.

Sabyasachi Chowdhury is a Senior Architect with the Insurance Technology Consulting Group within Cognizant. He has experience in SOA, multi-country rollout, JEE-based insurance applications and enterprise architectures. His interests include architecting, multi-country implementation and large-scale technology modernization programs. Sabya has a Bachelor's degree in electrical engineering from NIT Rourkela (India) and is a PMP and certified in TOGAF 9 and SCRUM. He can be reached at Sabyasachi.Chowdhury@cognizant.com.



About Cognizant

Cognizant (NASDAQ: CTSH) is a leading provider of information technology, consulting, and business process outsourcing services, dedicated to helping the world's leading companies build stronger businesses. Headquartered in Teaneck, New Jersey (U.S.), Cognizant combines a passion for client satisfaction, technology innovation, deep industry and business process expertise, and a global, collaborative workforce that embodies the future of work. With over 50 delivery centers worldwide and approximately 118,000 employees as of June 30, 2011, Cognizant is a member of the NASDAQ-100, the S&P 500, the Forbes Global 2000, and the Fortune 500 and is ranked among the top performing and fastest growing companies in the world. Visit us online at www.cognizant.com or follow us on Twitter: Cognizant.



World Headquarters

500 Frank W. Burr Blvd.
Teaneck, NJ 07666 USA
Phone: +1 201 801 0233
Fax: +1 201 801 0243
Toll Free: +1 888 937 3277
Email: inquiry@cognizant.com

European Headquarters

Haymarket House
28-29 Haymarket
London SW1Y 4SP UK
Phone: +44 (0) 20 7321 4888
Fax: +44 (0) 20 7321 4890
Email: infouk@cognizant.com

India Operations Headquarters

#5/535, Old Mahabalipuram Road
Okkiyam Pettai, Thoraipakkam
Chennai, 600 096 India
Phone: +91 (0) 44 4209 6000
Fax: +91 (0) 44 4209 6060
Email: inquiryindia@cognizant.com