P E R S P E C T I V E

Legacy Transformation

Finacle from Infosys®

YOUR INNOVATION PARTNER

Universal Banking Solution | System Integration | Consulting | Business Process Outsourcing

The following article presents a high level view best used by a CTO/CIO of a bank when deciding on large scale system transformation projects such as core banking replacement. The article begins with an analysis of various parameters for finding problem areas in the current architecture after which they are analyzed for maturity levels of the target architecture. It then addresses the subject of measuring indicators for level of complexity of transformation. The decisions and analysis are indicative and are quite sophisticated, calling for some deep thinking. The user of this paper is expected to know the strengths of his/her organization, be able to evaluate them and compute maturity levels for the various parameters mentioned in the paper.

Large Scale Legacy Transformation like Core Banking

Today, most banks are investing in technology and technology based transformation on a large scale. It is every CTO/CIO's dream to create a brand new technology platform housing numerous specific systems either built in-house or purchased from product vendors. It is possible to leverage existing systems and create many layers of software on top of them; in that case, the business logic gets distributed across various systems and probably technologies. At the same time, banks can use niche market products providing a common platform and technology, as the first step towards transformation. This article makes an attempt to help CTO/CIOs take decisions pertaining to large scale transformation requiring technological progressions.

Often, the CTO/CIO finds it difficult to decide whether the existing architecture requires a complete make over or whether it can be made sustainable by just building more need-based software components. The task becomes even more difficult with the need to justify the cost and build a convincing business case. The immediate cost of building new business functionality over the existing infrastructure is considerably lower than that of replacing the system landscape completely. But huge long-term expenditure on maintenance and lack of business flexibility wipe out this advantage. A centralized new platform architected to form a robust and scalable foundation for the bank in the long run is therefore a better option. However, it is absolutely essential for the decision maker to do a self-check before taking the big plunge.

Defining the goal is the first step when evaluating the need for transformation. A few basic questions must be answered:

- What is the target landscape for the systems?
- Which of them need absolute replacement and which can be retained and maintained?
- Can significant technical benefits be achieved by moving multiple systems on a single packaged product?

Secondly, the question of how to achieve that goal can be tackled in multiple ways based on the program management and risk taking abilities of the organization.

Determining the current Maturity Level of the Organization for Transformation

As part of decision making, it is essential to pinpoint the current problem areas for the bank. Multiple pain points exist in a distributed environment comprising a large number of individual systems. An example typical of large banks is as follows: The core back end is on mainframe (or similar technology) and the front end is separately built on an open systems technology. The core back end is split into multiple smaller systems on the mainframe. A system on the mainframe technically enforces a logical boundary, since the entire software physically resides on a common server accessible to all. In case there are two distributed systems running on different technologies, both need to be integrated using standard mechanisms in order to access each other. Such technologically non-centralized architecture is usually associated with the following problems which can be used to estimate the overall need for transformation.

1. Code logic duplication in various systems: Even business logic that is common to different systems and technologies may be repeatedly written due to legacy reasons. This pattern, if common to different areas, indicates that the overall code distribution is decentralized and requires significant maintenance to implement continuous

changes to the existing business. To understand this better, let us look at an example of decentralized logic. Bank account balances and transactions are visible on various channels indicating that the logic of deriving the same is repeated in individual channel systems like Internet Banking, ATM, IVR and core backend. This duplication is not scalable for growth, as the logic has to be maintained at multiple places.

2. Data duplication in various systems: Business data may be stored within multiple systems because of historical reasons. Data duplication is one of the most common and important reasons for increased maintenance cost. Defining data centrally and making it available to the most commonly used systems reduces the need for storage space as well as maintenance overheads. It also trims excessive interfaces and improves their quality.

3. Disparate technologies: Although various technologies are capable of providing similar results, it is extremely important to have a single view of the system and the technology landscape of the organization. Having too many technologies and interface hops for a particular business service can add to the cost and is very difficult to maintain in the long run. Reducing the number will definitely lessen overall complexity, enabling the CTO/CIO to concentrate on the benefits of each and create standards for the organization to follow. For instance, parameters like authentication, security, reconciliation and consistency can be defined and exercised centrally to benefit all J2EE based systems.

4. Excessive data transfer between various systems: A large bank may have too many interfaces due to lack of enterprise level standards; the right number can be assessed based on an SOA guideline. A simple comparison of the total number of business services of the organization and the average number of interfaces from a single system will help to compute the ideal ratio that must be maintained for each business line and system. Too much of variance will mean increased maintenance overheads. It is also important to estimate and decide the optimal number of technology hops - from mainframe to Open Systems and Open Systems to mainframe or others - to achieve one single business service. Exceptions can exist, but establishing standards and maintaining control at the central level is absolutely essential.

5. Excessive reconciliation: In complex systems architecture, there is a strong possibility that massive reconciliation will be required due to reasons like inefficient operational processes, manual work arounds for technological solutions or voluminous data movement from one system to another. A lot of this can be avoided by replacing multiple systems on a single technology with a common packaged product, which also leads to greater technological benefits and ease of integration.

6. Relational data model: It is important to examine the layout of the relational data model of the organization by asking:

   - How is the business data addressed at the system and technology level?
   - Is there a single definition for all the business fields and are the same definitions reused?
   - Are the data constraints and relationships within and across systems clearly defined?

   Answers to such questions can decide the maturity level of the model. The higher this level, the lower the need for user education, training, mapping of business requirements to technical design, procedure building etc.

7. Batch architecture: If the central architecture of the core system is of the batch type, online needs can be met by writing additional logic. For instance, memo posts are built into a lot of mainframe-based systems to achieve real-time results in a batch-posting environment. It is important to move to an online real-time single system as it provides additional business and technology benefits like statements, general ledger and integration, all in real time. This also reduces the huge overheads associated with building a batch system at the End of Day.

8. Large offline end of day processing: Batch architecture necessitates daily system downtime. Banks employ additional software solutions in order to provide uninterrupted service to customers during the time that their systems are offline. In case there are multiple individual core batch systems, reconciliation after offline hours becomes extremely complex. The resultant increase in maintenance cost and reconciliation can be avoided with a true 24x7 online real time core system.

9. Resources and skills: Many banks have large systems built in-house for most of their business lines. But resources and skills could be lost over a period of time making modifications extremely difficult. Also, it is not easy to bring specialized resources or skills from the market for in-house system development. This problem can be solved by using packaged products.

10. Business process definition: Clearly defining all the business processes of a bank and mapping them to technology is vital for maintaining large scale system architecture. Any change in the process can thus be easily translated into an impacted system. Well-defined business processes also reduce human intervention through automation, making it easier to introduce changes in business process or logic. Answers to the following questions will help ascertain the maturity level of this parameter.

    - Are all business processes well defined?
    - Are they linked to operational processes?
    - Are the impacted systems linked to business processes?
    - Are the business rules and external integrations captured in detail?

11. Parameterization and business flexibility: A highly parameterized system will allow plenty of business flexibility. If the current systems are hard coded and have very poor maturity on parameterization, effecting any change in business requirements will call for technical support.

12. Performance and non-functional areas: Every bank follows standard SLAs for different transactions. There are various technical parameters to measure the performance of infrastructure. High degree of standardization, parameterization and process orchestration will lead to additional overheads; therefore, achieving the right balance between the technical and non-technical parameters is very important. For some time-critical services, it is absolutely essential to achieve the result as fast as possible, whereas other asynchronous or back office processes can take longer. The maturity level of this parameter is derived from the availability of standard SLAs and the bank's ability to optimize the same.

13. Documentation capability: Documentation plays a critical role in preserving know-how and sustaining an organization's ability to execute large transformations even after key personnel leave. Current system functionalities, business requirements, data models, business processes, operational guidelines etc. need to be well documented.

14. Tools and automation: Tools which automate operations reduce excessive human intervention during the maintenance of large scale IT systems. A high maturity level on automation will definitely make daily maintenance easier and enable error-free changes to business requirements.

If most of the above indicators have poor maturity, it may be time for a complete transformation. Although there is a significant cost attached to it, this make over is easily achievable. A simple way to do this is to score the above parameters on a scale of 1 to 5 (where 1 signifies the lowest maturity level and the largest problem, and 5 signifies the highest maturity level and the smallest problem).The sum of the scores can be used to determine the overall technology strength of an organization. If it is lower than 30, it is time to look for a strategic replacement. However, this technique must be used only as a general guideline and further assessment of individual parameters with high ratings must be conducted. Alternatively, the

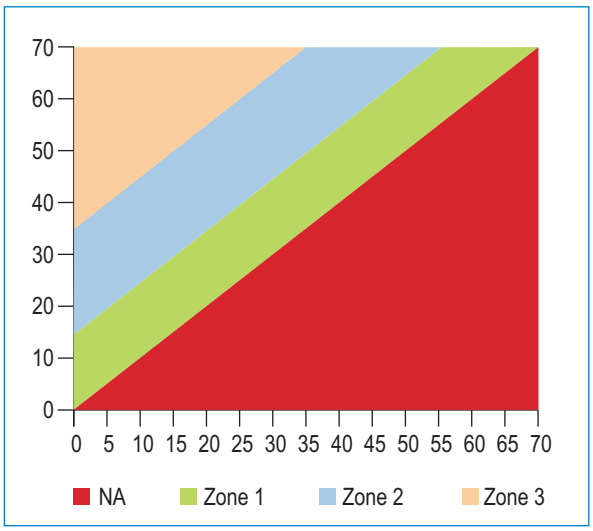
weighted sum can be computed taking into consideration the ultimate goal and the importance of each parameter.

How difficult is it to get to the target?

Once the decision of a strategic system transformation is taken, it is extremely important to understand its challenges. Various indicators should be analyzed in detail to evaluate the current situation of the bank as a whole and its readiness for transformation. Often, there is a lot of pain involved in the process. Also, the associated effort and cost should be quantified before commencement.

The complexity of transformation can be determined using the factors discussed above. No doubt, core banking transformation is challenging and achieving a perfect balance between all the parameters is quite difficult. A perfect result may be a CTO's goal, but the organization's cost and skill constraints can play spoilsport. Based on their current maturity levels and need for improvement, banks may be segregated into the three zones defined in the subsequent section.

Determining the zone that a bank belongs to is a matter of assessing the existing and target maturity levels for each of the 14 parameters and calculating the difference between their cumulative sums.



X - Current Maturity Level
Y - Target Maturity Level

Zone 1: Difference between the current and target maturity levels <=15

Zone 2: Difference between the current and target maturity levels >15 and <= 35

Zone 3: Difference between the current and target maturity levels >35

The characteristics and recommended approach for each zone are indicated below:

- Zone 1 - Implement with patience: A bank in this zone has already matured on most, if not all parameters. With patience, even large transformation can be implemented successfully. Every process and project management technique to be followed should be laid down clearly before initiation. In depth solution analysis will ensure that the end result is accurate and achievable. The decision of whether to go for a big bang or a multi-phased rollout depends on project management capabilities and risk appetite.

- Zone 2 - Implement with caution: Transformation is possible but due to poor maturity in the existing infrastructure, achieving the end result would need a lot of focus and involve some cost as well. It is highly advisable to de-risk implementation by first testing it in small phases and gradually executing it on a larger scale. Modules considered least risky because of their limited business impact can be taken up first. Once the bank completes a few small phases (ideally 1 to 3 in number), it can take up large scale conversion. Phases may be divided according to region, customer segment, business line, product, branch, system etc. Along with implementing core banking, the bank can improve overall maturity as well. If the bank does not have prior experience in large scale implementation, it must seek external expertise. It must ensure coordination between all its units (Business, Operations, Technology) and specify cost constraints right from the beginning.

- Zone 3 - Prepare now and implement later: The organization is not yet ready to take

the big leap and must spend enough time and resources to pull itself up on those parameters where it falls short. Once the required maturity is achieved and the bank lands in Zone 2 or Zone 1, the transformation can be implemented.

Identifying Inner Strengths of the Organization to Aid Decision Making

Before launching this exercise, a CTO/CIO should also leverage historical data to assess the bank's capabilities. The following questions need to be answered:

1. How many large transformational projects has the bank implemented in the past?
2. How many of them were successful on time?
3. How many projects were not successful because of the following?
   a. Non-availability of funding
   b. Lack of skills and execution ability within the existing resources
   c. The business' inability to assume risk
   d. Lack of coordination between technology and business teams

Some banks are highly risk averse, whereas other, more dynamic banks may be willing to take on high risk projects, but lack the necessary skills. A study of historical data and the zonal analysis described earlier can help a bank arrive at an informed decision on whether or not to transform.

Author

Mukund Galgali
Lead Consultant - Finacle
Solution Architecture and Design Group
Infosys Limited

Finacle from Infosys®

YOUR INNOVATION PARTNER

PERSPECTIVE

Universal Banking Solution | System Integration | Consulting | Business Process Outsourcing

Infosys Limited

Plot No. 44, Electronics City, Hosur Road, Bangalore - 560100. India
Tel.: + 91 80 28520261, Fax: + 91 80 28521747, e-mail: finacleweb@infosys.com
www.infosys.com/finacle

Join us on Twitter, LinkedIn and Finacle Whiteboard at www.infosys.com/finacle/networking.asp

"COPYRIGHT NOTICE: Copyright ©2011 Infosys Limited, Bangalore, India. ALL RIGHTS RESERVED."
Finacle logo is a registered trademark of Infosys and Infosys acknowledges the proprietary rights of the trademarks and product names of other companies mentioned in this document. Infosys believes the information in this publication is accurate as of its publication date; such information is subject to change without notice.