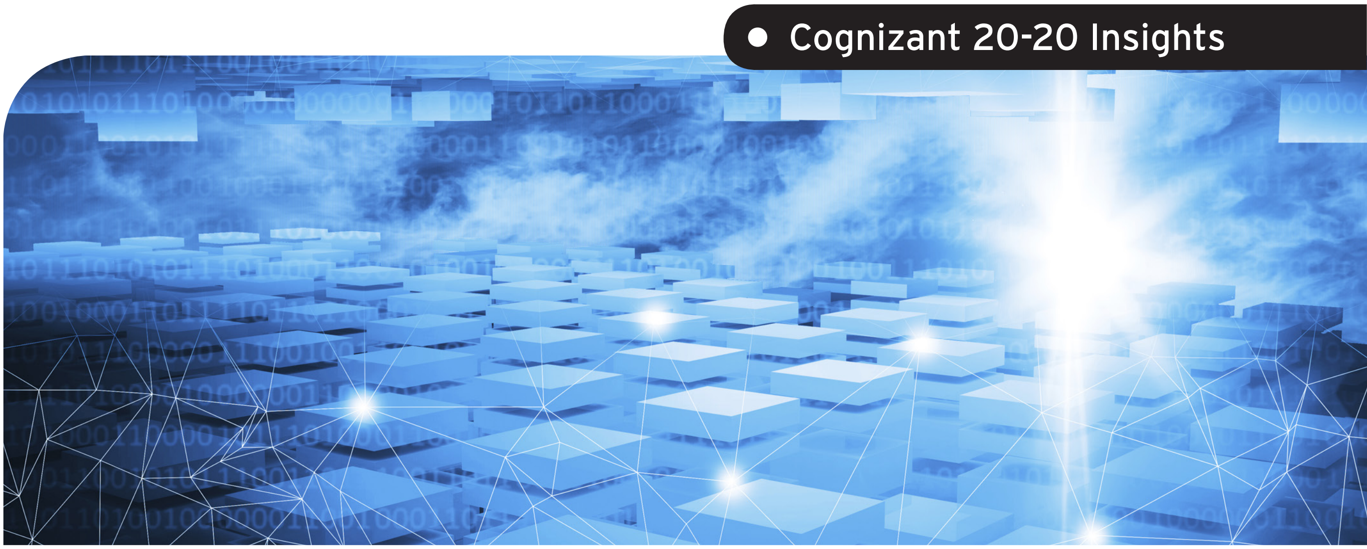
Cognizant 20-20 Insights



Legacy Enterprise Systems Modernization: Five Ways of Responding to Market Forces

Whether your organization replaces the entire enterprise application system, replaces one application/component at a time, applies localized fixes in one application or modernizes underlying infrastructure, any approach is typically better than none at all.

Executive Summary

Market forces demand a response from companies in every corner of the world seeking to maintain a modern and effective enterprise information environment. Whether it is consumerism, digitization, social-mobile-analytics-cloud (aka the SMAC Stack) or productization, the organizations that run business IT systems must be agile enough to respond to market demands in better-faster-cheaper ways.

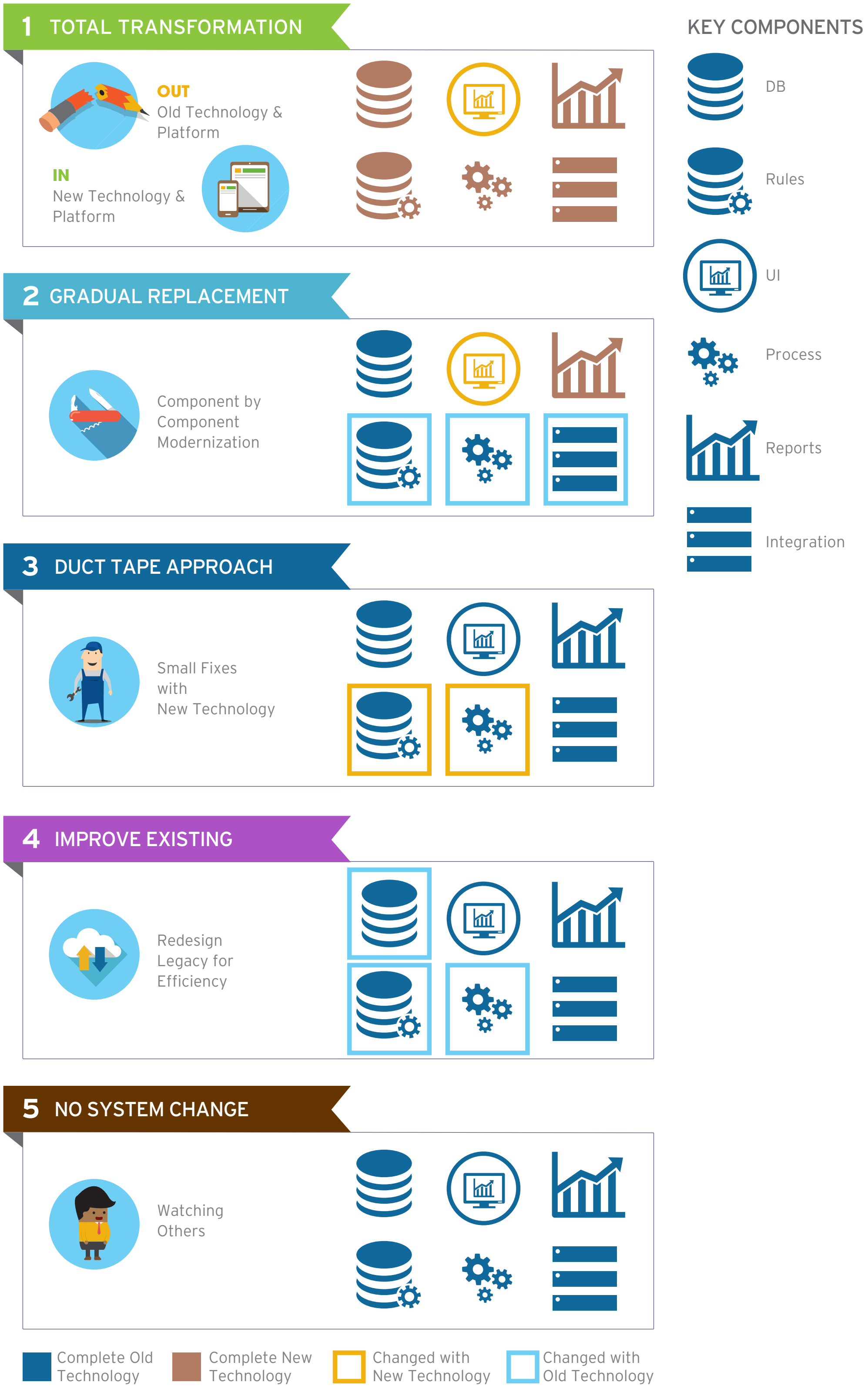
However, not all legacy enterprise applications – many of which were built on decades-old technology and use architectures that were built when IBM mainframes still ruled – can support the continuously accelerating pace of business change.

Hence, many companies now realize that they must transform their IT systems with a modern technology stack or risk irrelevance.

However, how they approach this transformation depends on factors such as their organizational culture, scope, urgency, risk appetite, existing technology stack, technical expertise, program management maturity, funding availability, etc.

This white paper describes the five ways we have seen companies approach legacy modernization, four of which, if done right, can keep them in the competition.

Modernization Approaches



1 TOTAL TRANSFORMATION

OUT Old Technology & Platform

IN New Technology & Platform

2 GRADUAL REPLACEMENT

Component by Component Modernization

3 DUCT TAPE APPROACH

Small Fixes with New Technology

4 IMPROVE EXISTING

Redesign Legacy for Efficiency

5 NO SYSTEM CHANGE

Watching Others

KEY COMPONENTS

■ Complete Old Technology ■ Complete New Technology □ Changed with New Technology □ Changed with Old Technology

Figure 1

Five Ways of Modernization

1 TOTAL TRANSFORMATION

Approach: The entire system is rebuilt using new technology, and the old system is sunsetted. It is built from scratch using standard platforms (e.g., J2EE) or built using a third-party package as a foundation layer.

Example	• End-to-end healthcare claims system replaced with a new claims package.
Applicable Situations	• Current age-old system cannot support additional changes needed by the market. • Underlying technology platform is no longer supported by the vendor. • Current technology is expensive to license. Other companies are moving to a less expensive stack (often open source). • Post-mergers/acquisitions, the company needed to build a new application with unified functionality. • New leadership was pursuing a new IT vision to support aggressive business goals.
Benefits	• Provides the highest return and best competitive edge over competitors. • Technology stack and architecture is modern enough to remain competitive for several years into the future. • Offers the maximum flexibility in architecture and design and is not constrained by old components. • Operational measures, SLAs/KPIs, business effectiveness and efficiency all receive a huge boost. • Offers an excellent opportunity to redefine/improve the business processes leveraging new systems.
Success Factors	• Proper due diligence/assessment must be conducted when selecting the technology. • Assessment phase to define roadmap with key milestones (e.g., go-live dates). • Iterative roll-out to minimize the volume and impact of change (compared to big-bang go-live). • Stakeholder management with regular communication and end-user training and acceptance. • Strong program management and governance practices must be followed. • Production parallel execution ("parallel adoption") and comparison of old and new systems. • Understanding the data structure and data quality if data migration is involved. • Capture and validate the nonfunctional requirements (e.g., availability, performance, etc.). • IT department's ability to learn the new technology and augment it with outside talent if needed.
Risks	• This is the most risky, yet most rewarding approach. • Entire functionality is at stake and any quality issue can impact the business operations. • Likelihood of failure or midway stop/hold is highest due to cost overruns or missed deadlines. • Pockets of resistance by end users due to inability to handle/accept the change. • Business sponsor may become demotivated if no incremental business benefits are demonstrated.
Verdict	• This is the best option if done properly, as it takes the systems far ahead of competitors. But only organizations with the required IT capability and maturity should go this route.

2 GRADUAL REPLACEMENT

A component/functional block of an IT system is replaced with a new technology and moved to production as a separate application while the rest of the system remains on old technology. With time, remaining components/functional blocks are replaced with separate apps and gradually the entire system is rebuilt.

Example	• Claims accumulator functionality (i.e., for calculating deductibles, etc.) is taken out of an end-to-end legacy claims systems and built as a separate application.
Situations	• Replace entire systems one component at a time, with controlled release of budget. • Only a few components contribute to most of the issues. • Externalize and consolidate scattered business rules using business rule management System (BRMS) platform. • Eliminate tight coupling of the systems by implementation of enterprise service bus (ESB). • Change the existing systems from batch to real-time/online. • Build service wrapper on top of the existing systems to provide the data to other systems. • Modernize flat files or outdated database systems using leading RDBMS systems. • Replace legacy dumb terminals with new UI, supporting Internet browser and mobile devices. • Reduce license cost by implementing an open source platform.
Benefits	• This is a low-risk way to transform the entire system by moving one piece at a time. • This requires much lower one-time budget approval compared with a total transformation. • Since one functionality is migrated at a time, the business impact and cost of failure are lower. • Since work volume is less than a total transformation, it consumes less management bandwidth. • Delivers results quicker compared with a total transformation. • Breaks one large systems into SOA-based componentized systems. • SLAs/KPIs and other measures can be tracked and improved at more granular levels. • Allows companies to choose different technologies or newer versions for later components.
Success Factors	• All relevant program management practices are followed. • Explore ways of building reusable components or other artifacts for future applications. • Follow clear technology direction for all new applications for consistent architecture and design. • Governing body to ensure all new applications are tied to common goals and roadmaps. • Long-term continuity of key resources to support multiple new application development. • Smooth communication among different concurrent application teams is key for consistency. • Nonfunctional requirements must be captured and validated for all applications.

Continued on following page »

Continued from preceding page »

Risks	• Risk of amalgamation issues among different applications if not managed properly. • In the absence of close monitoring, each app can end up using a different tech stack. • Can lead to a set of disjointed applications that do not function well as a unit. • If end state is not factored in design, future scope additions can be difficult to implement. • If architecture is not planned well, it may lead to too many components and integrations. • Concurrent development of two apps can lead to versioning issues involving common components.
Verdict	• This is one of the best approaches to follow as it slowly brings in new technology while keeping the change impact and risk manageable. Success rate is typically very high with this approach.

3 DUCT TAPE APPROACH

Localized, small-scale changes are performed using new technology to address specific issues in the application while the application core architecture and technology remain the same. A popular approach is to build a new application that will be bolted to the main application to bridge the gap in functionality.

Example	• Increase legacy claims system's auto-adjudication ratio by automatically correcting pending claims (i.e., those flagged for manual correction) using a separate rules-engine-driven application.
Applicable Situations	• Company plans to continue with the legacy app but will fix existing issues with new technology. • Focus on current problems (e.g., improving KPIs/metrics) that must be fixed ASAP. • Facing a new problem in the middle of the year with no budget for a more comprehensive solution. • Stop-gap solution to fix current issues to gain enough time for planning modernization. • Company has undertaken a large program that consumes IT manpower bandwidth and risk appetite.
Benefits	• Small-scale changes offering comparatively big returns. • Doesn't require huge investment and can be supported through ad hoc budgets. • ROI is concrete and substantial, and increases the confidence of the project sponsor. • Offers quick wins since results are often delivered in months. • A less risky approach; probability and cost of failure are typically low. • Doesn't require as much management attention as larger transformations do.
Success Factors	• Technical governance to ensure this easy-to-follow approach is not overused/misused. • Careful attention to future roadmap/strategy is required before attempting this approach. • Rebuilding the application may be cost-effective if the app requires too many changes. • No matter how small the change is, all project management processes must be followed.

Continued on following page »

Continued from preceding page »

Risks	• Too much of this approach in too many places can lead to patchwork apps and bad design. • Cost of multiple changes in an app may be higher than required to replatform the app. • A change performed without due diligence can lead to throwaway work. • A stopgap solution may not keep pace with changing business needs over time. • Future total replacement of the application would make duct tape change a sunk cost. • Due to its smaller budget, this approach is often overlooked and proceeds without management review.
Verdict	• It is one of the most common and successful approaches used, as it delivers quick wins with comparatively low investment and risk. However, it still remains a tactical solution and works best in combination with an effective overall long-term strategy.

4 IMPROVE EXISTING

The existing system is modernized to offer better results through improved design. Typically, the core technology stack remains the same or a few minor additions may be introduced.

Example	• Improve application code maintenance by consolidating scattered and repeated business rules into common components and removing dead/redundant code.
Applicable Situations	• Leadership has strong association with legacy app and wants it to continue. • IT people are highly skilled in old technology and have found ways for continuous improvement. • Technology foundation is modern (e.g., Java/J2EE) though the application may be old. • The application is already being migrated to a new version, and synergy (project management and testing) can be used for functionality enhancements. • System design and architecture are modernized using the same technology to remove bottlenecks.
Benefits	• Extends the lifespan of the application while attending to immediate burning issues. • No need to make an immediate decision on future technology. • It doesn't require procurement of any new technology, and can be used to avoid new license costs. • Existing legacy team can handle the change, and it does not immediately require new technical skills.
Success Factors	• Consult service providers experienced in modernizing legacy applications. • Make the application componentized and aligned with SOA principles. • Proper documentation of business requirements and testing needs for a future re-platform. • Review major investments in legacy systems to ensure alignment with IT vision.

Continued on following page »

Continued from preceding page »

Risks	- If the system lifespan is limited, any large investment on the application can be a sunk cost. - This might open up the floodgates for fresh development using an expiring technology. - Scarcity of experienced resources in an old technology can lead to quality issues.
Verdict	- Every system comes with an expiration date and it is unlikely that any age-old legacy application will continue for an indefinite period. An initiative like this can inject a few years of life, which can provide just enough time for the organization to get ready to embrace new technologies.

5 NO SYSTEM CHANGE

Clients are taking a wait-and-watch approach and not going for any modernization drive or systems change.

Example	- Organization found the ROI of claims modernization is not worthwhile and decided to wait.
Applicable Situations	- Company has modernized in the past or it is a new company with new systems. - Company is evaluating various modernization approaches and potential scope and roadmap. - Leadership is indecisive and not able to make up its mind. - Company is focused on absorbing the change in leadership or company structure. - Leadership believes company doesn't need technology advancements to support growth. - Lack of funding availability.
Benefits	- No need for investment, and the company can make use of the available money for other purposes. - In the absence of large IT projects, IT department has available bandwidth. - Since no technology decision is made, company can retain the flexibility to choose any new platform.
Success Factors	- Company can look to business process optimization to derive benefits and efficiency. - Company must continue to observe the industry and prepare for transformation.
Risks	- Competitive disadvantage in quickly responding to market forces. - Runs the risk of falling behind peers and losing both market and mind share. - Company's business processes are more likely to be more costly and lack the required flexibility.
Verdict	- Unless company has a definite plan to make sure its systems are agile enough to support the growth, following this approach for too long can be injurious to its future.

Approach Comparison

	1. Total Transformation	2. Gradual Replacement	3. Duct Tape Approach	4. Improve Existing	5. No System Change
Modernization	Highest	High	Moderate	Low	None
System Change	Enterprise Level	Application Level	Localized Change	Application Level	None
Risk Severity	Highest Risk	Moderate Risk	Low Risk	Moderate Risk	Risk of Falling Behind
Funding Need	Highest	High	Moderate	Moderate	Low
Duration	Long (Multiyear)	Moderate (1-2 years)	Moderate (2-6 months)	Moderate (4-10 months)	Not Applicable
Benefits	Highest	High	Moderate	Moderate	Little
Popularity	Low	High	Highest	Moderate	Low
Success Rate	Moderate	High	Highest	High	Not Applicable

Figure 2

Looking Forward

Each of the five types of transformation approaches has its own merits and applicability. Which approach works best for any company depends on its organizational requirements, risk appetite, state of its existing IT infrastructure and competitive landscape – there is no single silver bullet.

Companies can follow one enterprise approach or choose different strategies for different units (e.g., LOB, business functions). The best way to determine the optimal approach is to perform deep-dive due diligence to evaluate the engagement and develop a well-considered set of recommendations and a move-forward roadmap.

About the Author

Souvik Roychoudhury is a Director within Cognizant's Integrated Process Management Unit. He has over 17 years of experience in IT solutions, delivery and consulting, predominantly in the healthcare and insurance industries. Souvik also has extensive experience in business transformation programs and technology upgrades. He has handled inception to implementation of 15 different large-scale modernization programs. Souvik received his bachelor's degree in technology (B.Tech.) from the University of Calcutta and holds numerous certifications (PMP, CSM, MCTS, FLMI, FFSI, ACS, AIRC, ARA, AAPA, CCP, AINS, PAHM, ITIL V3 Foundation, Six Sigma Green Belt). He can be reached at Souvik.Roychoudhury@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 100 development and delivery centers worldwide and approximately 218,000 employees as of June 30, 2015, 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.

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
1 Kingdom Street
Paddington Central
London W2 6BD
Phone: +44 (0) 20 7297 7600
Fax: +44 (0) 20 7121 0102
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

Codex 1377