LEGACY MODERNIZATION

Strategic Decision

Application modernization/reengineering is considered when existing legacy applications enjoy properties worth preserving. They are already deployed and have usually undergone significant scrutiny by their users. A long history of maintenance has resulted in "stress-hardened" code and a wealth of test and validation capabilities. Nonfunctional properties such as performance and accuracy have been fine-tuned. Application history exists in the form of original designers, current and past maintainers, and (potentially) in bug reports and change order records.

However, the same history of maintenance that stress-hardened the applications, cause a problem: the system becomes 'brittle" and increasingly resistant to change. For example applications may lack many of the features of more modern ones, such as

- *Support for data abstraction*
- *Modularity*
- *Information hiding*
- *Concurrency support*
- *Data modeling capabilities*
- *Hard coded Assumptions about the application environment*

TECHNOLOGY DRIVERS FOR MODERNIZING

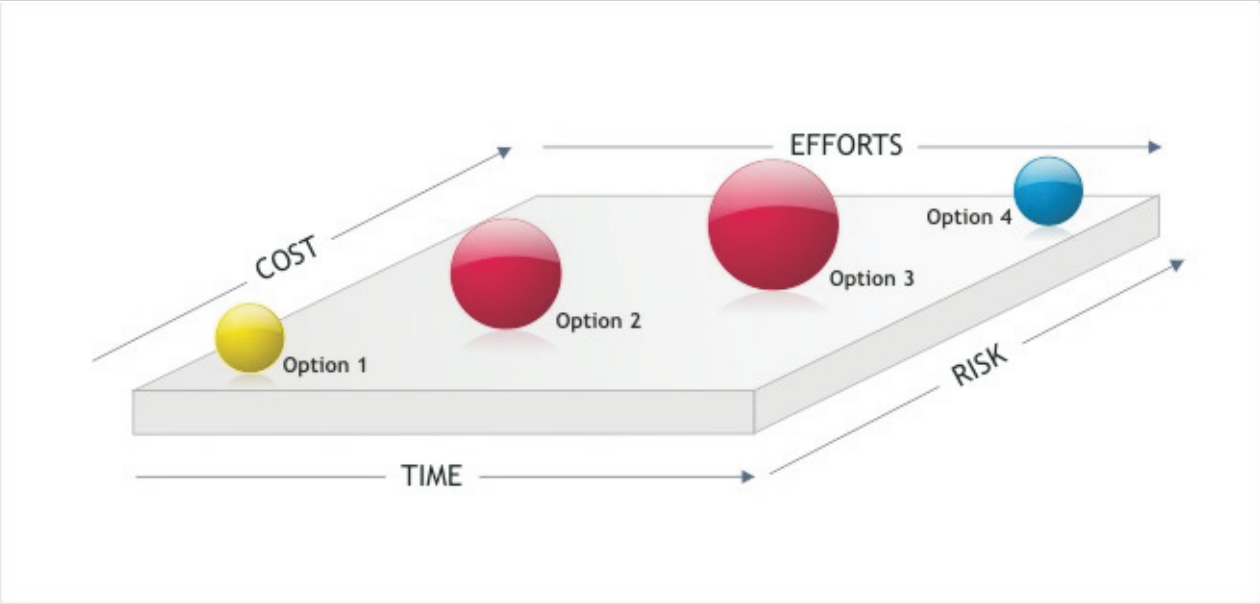
Drivers for modernizing applications vary depending on the organizational needs and business goals. These drivers may range from a simply business requirement to represent the information from say a Legacy green screen onto a Web browser or a need to access the legacy application code via the internet, or to a requirement to build true interoperability across databases and applications (including the need to persist, monitor and audit data as it moves around the enterprise).

MODERNIZATION STRATEGY EVALUATED

For modernizing applications, organizations broadly have four strategy options, apart from re-engineering/re-writing all the legacy applications.

- The first strategy with least effort, cost, time and change to applications is to simply update and optimize performance of the current system. This includes evaluating all system software (operating system, OLTP etc), databases and applications currently deployed in production to ensure that their advanced features are used to enhance usability, productivity, performance and interoperability with other applications/platforms.

- The second strategy is to keep the business logic of the applications on the legacy application, while providing the user with a more "friendly" interface to the data such as GUI screens or Web browser.



- The third strategy is to integrate the legacy applications including the business logic with new technologies while (most of) the data resides on the legacy platform.
- The fourth most intensive approach (short of re-engineering or rewriting the applications which is described separately), is to migrate part of the legacy applications to a different platform with little or no change to core business logic. Here the data may/may not be migrated from the legacy application.

Constraints of budget, time, resources and risk management compel businesses to modernize legacy applications by implementing one of the four strategies mentioned above. Option 2 or 3 or combination of both has been found to be the popular strategy. The strategy of re-engineering/re-writing the whole set of applications is not covered in this document.

A very significant point to note in favor of modernization is that the legacy platforms themselves are undergoing changes to conform to emerging technology trends. These include moving away from proprietary technologies to open systems, interfacing with popular third party products and platforms, and even having products provided by legacy system manufacturers that allow seamless integration with newer technologies. Example of this is BEA with Weblogic suite of products and IBM with the newer Z/OS operating system.

Additionally, it is worth keeping in mind that 70% of the world's data about customers, business operations and financials resides on legacy systems. These systems excel in high-speed transaction processing and reliability. As such, they are ideal back-end engines for newer Web and e-commerce applications.

The current competitive market demands that new business processes and functionalities be brought to market in months – not years. Cost and time are relevant issues. Within such an environment, leveraging off existing legacy systems seems the practical and effective solution.

ACHIEVING MODERNIZATION THROUGH SOA

Service-oriented architecture (SOA) is an ideal means of implementing modernization strategies described above. It offers the capability to develop a collection of services that can communicate with one another. Such architecture can be used to divide larger applications into discrete modules and make those modules available to other applications. This process addresses one of the key modernization challenges, that is to build interoperability between databases and applications.

SOA is not new, but a wide range of new technologies based on Internet standards have made the concept much more technologically and economically feasible. Technologies such as Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL) provide a simple yet effective way to exchange XML documents and describe how to invoke a Web service. Standards such as Universal Description, Discovery and Integration (UDDI) provide a way to register Web services so that one Web service application may locate and invoke another.

As SOA develops, it will be able to tackle large organizational challenges such as providing various business units with a customized "single view" of all the relevant data that resides throughout the enterprise, supporting the ability to persist, monitor and audit data as it moves through the enterprise from system to system.

ANALYSIS

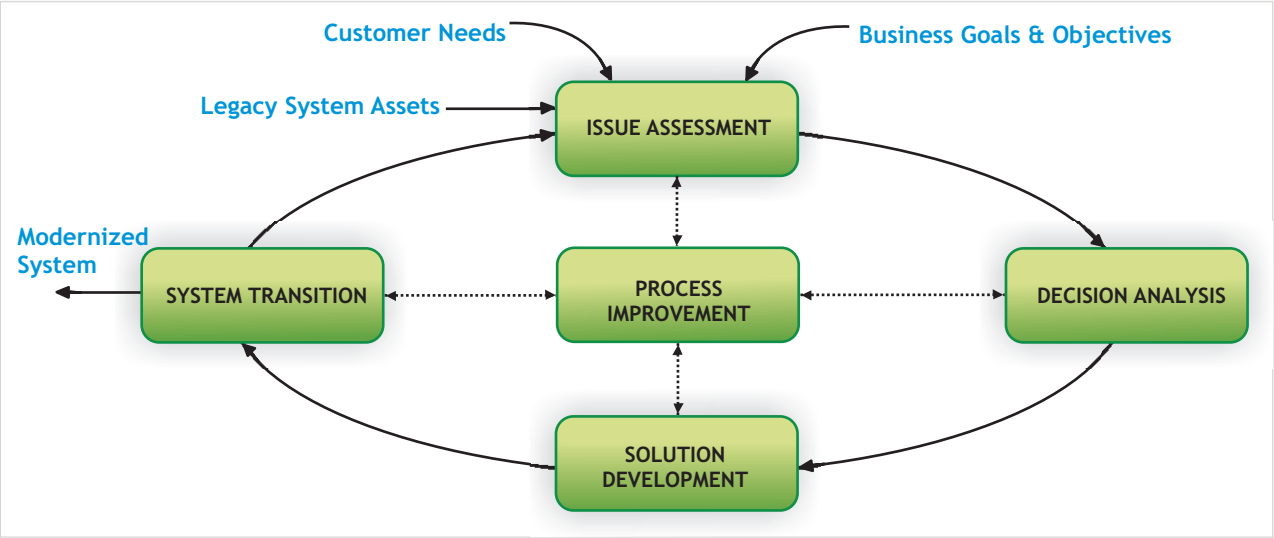
Whatever the driver for modernization, Web based architecture is an excellent foundation for extending the life, functionality, and value of legacy systems. By allowing data dynamically into multiple presentation formats, the technology provides a means to deliver standard access to both structured and unstructured information. Furthermore, the implementation of a Web services-based architecture enables universal access to all information systems.

This is not to imply that Web services are required in all cases. In many instances, a simple 3270 screen-scraper system may be all that is needed to accomplish an organization's goals. The difference today is that screen-scraper technologies can be combined with XML technology to quickly and inexpensively create easy-to-use Web interfaces. The underlying COBOL or PL/1 source code need not be changed, but the user experience is modern. Web extends the life of a wide range of legacy systems with minimum effort.

APPLICATION RE-ENGINEERING FRAMEWORK

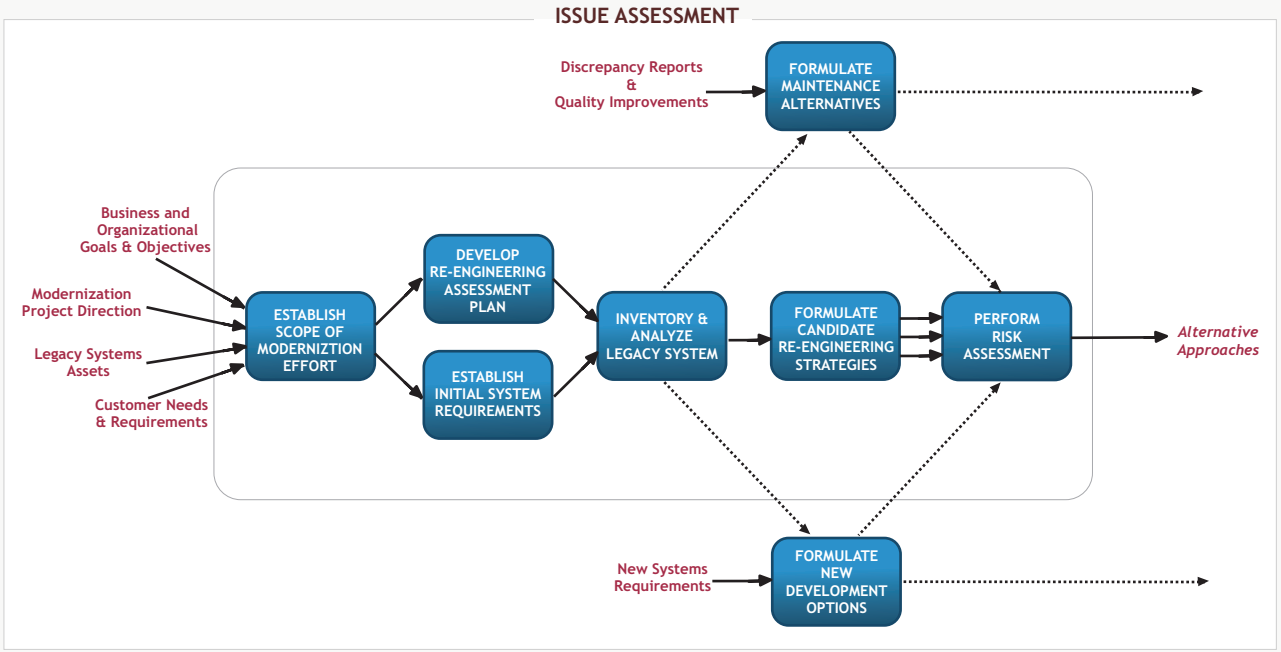
Modernization using the re-engineering process framework broadly encompasses five major phases:

1. Issue assessment
2. Decision analysis
3. Solution development
4. System transition
5. Process Improvement



ISSUE ASSESSMENT

The initial activity of this phase is to establish the scope and direction of the modernization/reengineering effort. Another activity is centered on developing the preliminary plan to guide and direct the initial phases of the modernization/reengineering effort.

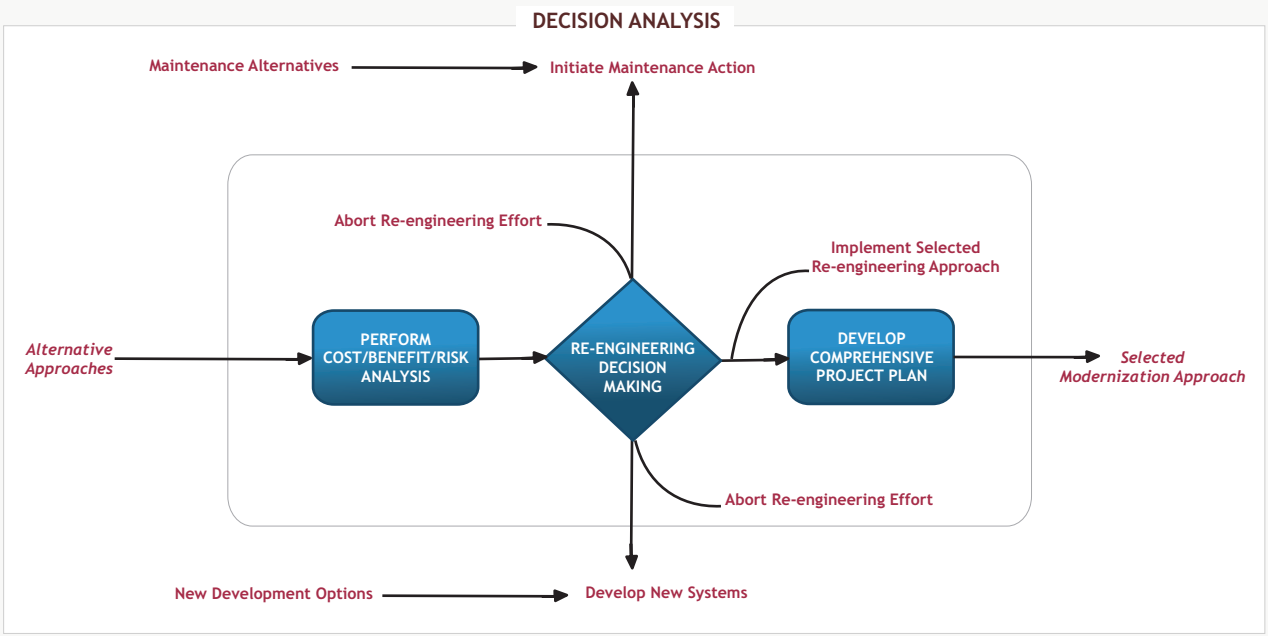


Successfully completing the issue assessment phase will establish the following

- The true scope and extent of the modernization effort.
- The explicit modernization goals and objectives.
- Quantitative and qualitative data describing the legacy system
- The feasibility and practicality of modernizing the legacy system
- Candidate modernization strategies and technical approaches.
- Alternative solutions to modernization

DECISION ANALYSIS

Following the issue assessment phase is the decision analysis phase. The decision-making process takes into account all the technical, economic and program issues that can influence and affect modernization. An integral part of the decision-making process is performing a cost/benefit/risk analysis and comparing re-engineering with other approaches or maintenance alternatives.

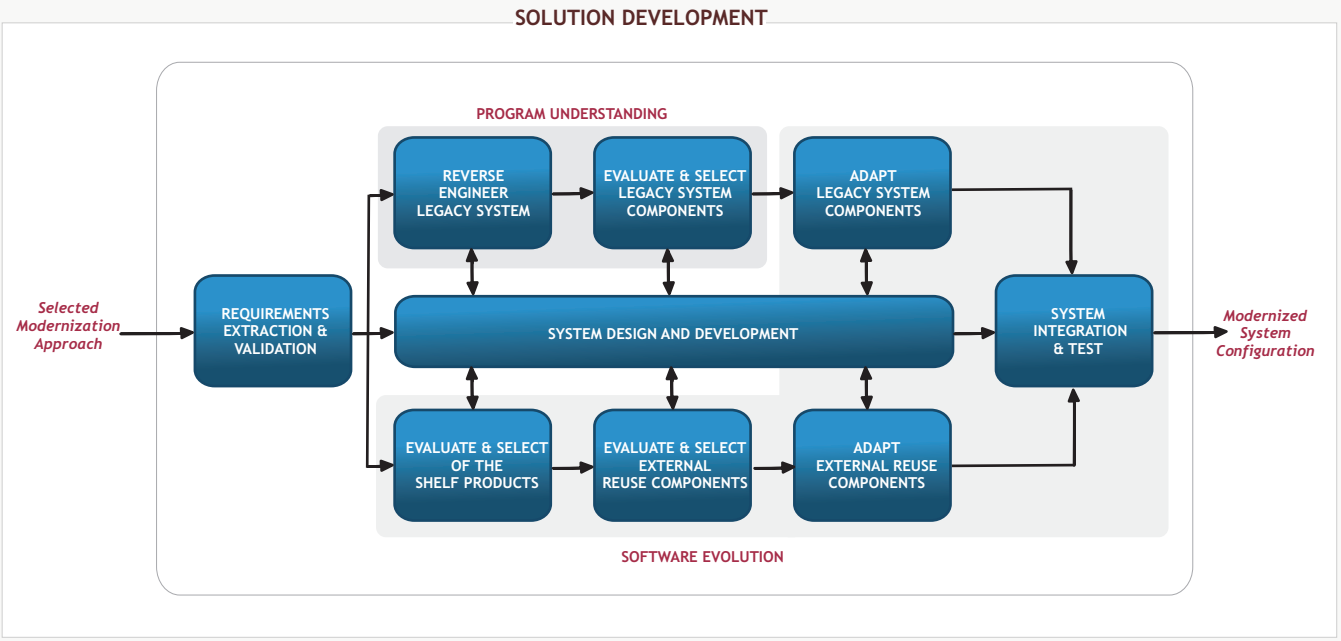


Successfully completing the issue assessment phase will establish the following

- Gathering comprehensive analysis data for the decision-making process
- Producing an approved and validated modernization/reengineering strategy and a high level technical approach
- Projecting the impact of modernizing/r-eengineering the system. Also risk management.
- Estimating the cost, resources, schedule, and risk management for modernizing/re-engineering the system.
- Forecasting the expected benefits of modernizing/re-engineering the system.

SOLUTION DEVELOPMENT

The third phase of the re-engineering process framework is the solution development phase, which constitutes the core element of the modernizing/reengineering effort. The initial guiding activity of the solution development phase is to formally elicit and validate the detailed system requirements for the re-engineered system. A reverse engineering activity, which is unique to re-engineering, is focused on analyzing the existing system to obtain an in-depth understanding of the software system.

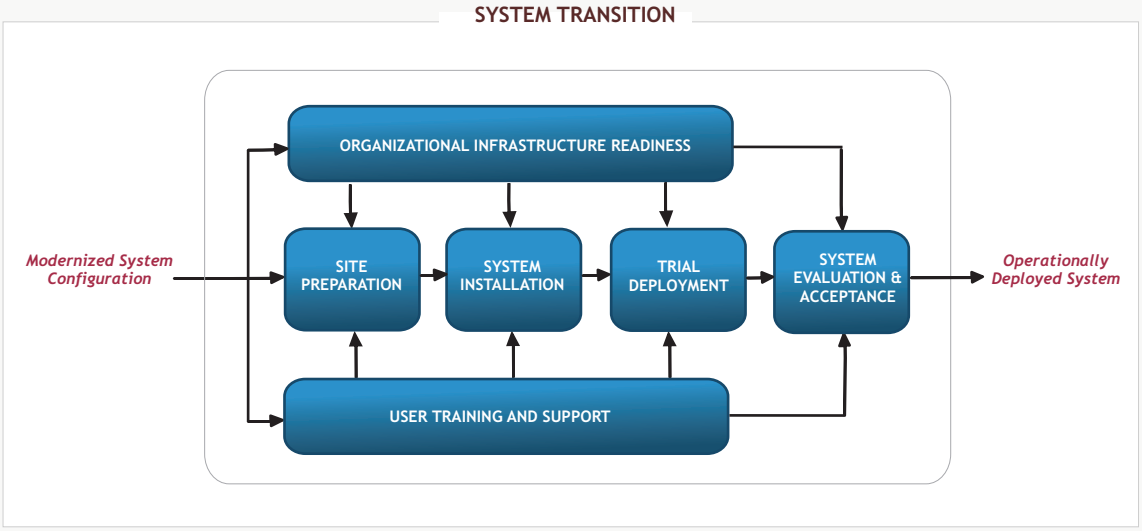


Successfully completing the solution development phase is essential to meeting the following goals

- A re-engineered system that is tested, demonstrable and complies with the specified requirements
- Delivery of a modernized system within the cost, schedule, and risk envelope

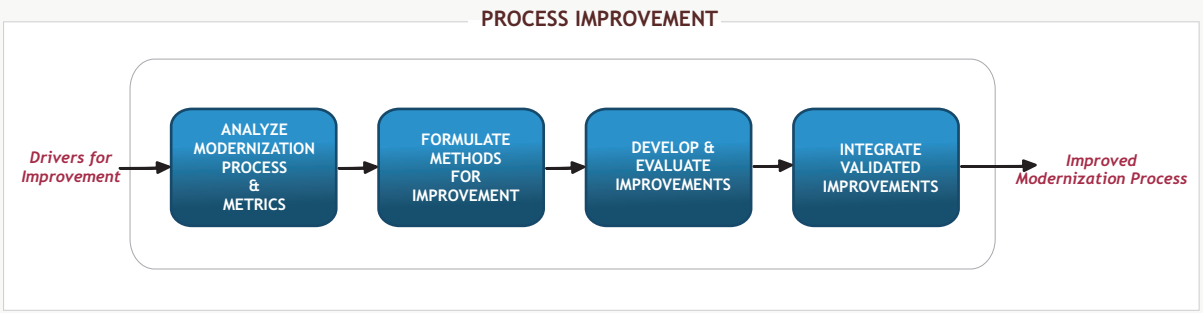
SYSTEM TRANSITION

The fourth phase of the re-engineering process framework is the system transition phase, which focuses on deploying the re-engineered system into its operational environment. The activities go beyond the normal site preparation and system installation and implementation. Ensuring that the users are properly trained and the organizational infrastructure is in a state of readiness are high-priority items. Another important aspect of the re-engineering transition phase is performance testing and allowing for trial deployments to ensure that all the “glitches" in the system have been fixed. This not only alleviates and circumvents many of the problems normally encountered in deploying a system, but greatly facilitates achieving customer satisfaction and user acceptance.



PROCESS IMPROVEMENT

The bottom line for a successful modernization project is effectively using the past experiences in modernization, re-engineering and development to ensure continual improvement for the benefit and advantage of the customer. The benefits include reducing costs, increasing productivity and gaining a competitive advantage by



BITWISE EXPERIENCE

By leveraging over a decade of software development experience, and proven delivery methodologies, mainly in the financial services domain that includes insurance, BitWise is uniquely positioned to meet the demands business for these initiatives.

The model of delivery practiced by BitWise is a customer focused, flexible, low risk and cost effective one that will deliver rapid return on investment to businesses around the world.

BitWise Application modernization approach is based on:

- Understanding the functional strengths of existing applications
- Understanding the additional business needs/gap analysis
- A well planned, best fit approach considering all requirements
- Drawing out the intended architecture
- Continual risk assessment and mitigation
- Mapping business processes to application capabilities
- Consolidating repetitive processes across business units
- Rationalizing business and application data
- Consolidating and redesign common cross-functional data
- Migrating/consolidating redundant applications to existing/new architecture
- Web-enabling / integrating user interfaces as required

CASE STUDIES

WEB ENABLING

A leading insurer that retails personal and commercial insurance products was required to find solutions to modernize its COGEN applications to enable its brokers and direct customers to access information that resides on COGEN. In the existing scenario, data could be accessed only by designated employees of the insurer.

The insurer initially looked at replacing its COGEN applications. BitWise evaluated their business processes workflow, existing applications and business requirements and after doing a gap analysis recommended options to the insurer to leverage their existing applications at a fraction of the cost to replace them, with increased functionality.

The project involved developing Web interfaces for insurer's six customer groups. The solution was to deploy message-oriented middleware that acted as broker between the range of legacy applications and newer Internet technologies. It involved having COGEN on the legacy platform that suited it best, and then accessing the data from it. The web applications were deployed on network servers. legacy applications were executed and the data was retrieved from the legacy databases. Some of the legacy applications generated an XML Data Stream to pass down to the Web applications to render internet-displayable format.

Using this technology BitWise was able to develop the web interface in seven months. Today customers - including 1,200 brokers, 400 account managers and 260 group administrators, are using the system daily. Data access is in real time and response time beats the sub-four-second goal set by the deployment team. The system is scalable for future growth.

FUNCTIONAL ADDITION

This case study depicts functional additions made to the existing COGEN Application to provide a Quoting system for use by the Compulsory Third Party Insurance Underwriters.

The CTP system did not contain a facility for the underwriters to use process and store customer quotes. A manual system was in place and added inefficiencies to the system as a whole and allowed loss of quotes.

A new screen was added to the existing system as shown in the next page

Quote Number:

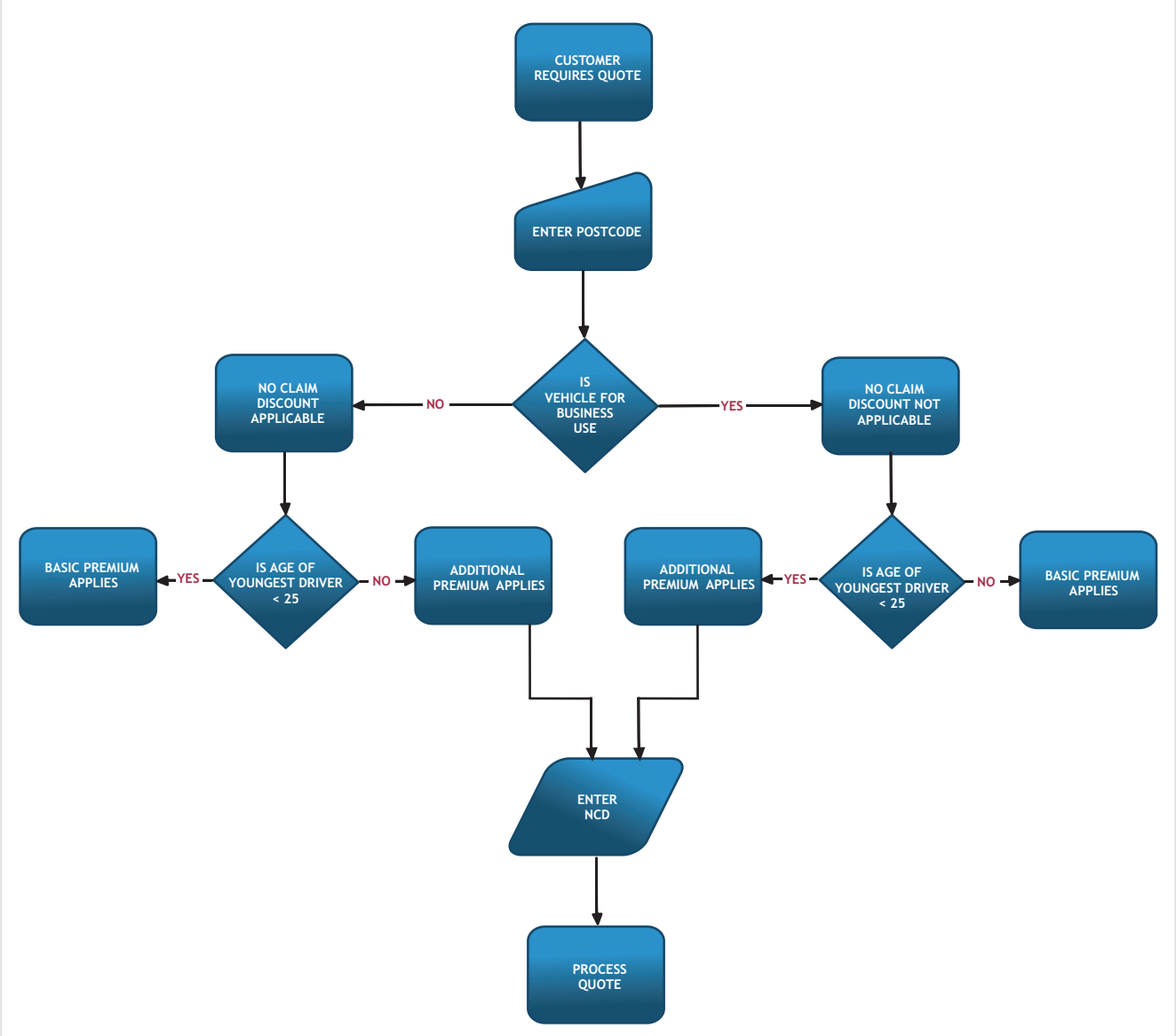
Suburb	:	_____	or
Post code	:	_____	
Business use	:	_____	(Y/N)
Date of Birth	:	_____	or
Age in Years	:	_____	
Registration	:	_____	
NCD	:	_____	
Date	:	_____	

Selection : _____ A = Add, P = Print, R = Reprocess

The business requirements were as follows

- If Date of Birth is entered, total in years is to be calculated and if less than 25 years additional premium is to apply.
- If Date of Birth is greater than 25 then base premium is applicable.
- If vehicle is to be used for Business purposes then additional premium is to apply, however if business use is yes and calculated age is less than 25 then only one additional premium is to be added.
- If vehicle is for personal use and calculated age is less than 25 then additional premium is to apply.
- If vehicle is for personal use and calculated age is 25 or over then base premium to apply.
- When selection is to add, quote number and date are to be generated.
- Quote number is to be 12 characters commencing with a letter.
- Quotes are to be deleted after 15 days from date entered.

The process flow of the added functionality is depicted below



BiTwise

Many Minds...One World

CONTACT INFORMATION

BitWise Inc.
1515 Woodfield Rd.
Suite 930
Schaumburg, IL 60173

Phone : 847-969-1544
Fax : 847-969-1500
Email : info@bitwiseusa.com

BitWise Australia Pty Ltd.
Level 39, 2 Park Street
Sydney,
NSW 2000

Phone : 61 2 9004 7887
Fax : 1300 790 860
Email : info@bitwiseaustralia.com

BitWise Solutions Pvt. Ltd.
BitWise World
Off Int'l Convention Centre
Senapati Bapat Road
Pune - 411016 - INDIA

Phone : 91 20 40102000
Fax : 91 20 40102010
Email : info@bitwiseindia.com

SUPPORTING PARTNERSHIPS

Microsoft GOLD CERTIFIED Partner | IBM Business Partner | expressor tm | talend open data solutions

© Copyright 2010 BitWise believes the information in this publication is accurate as of its publication date; such information is subject to change without notice. BitWise acknowledges the proprietary rights of the trademarks and product names of other companies mentioned in this document.