MODERNSYSTEMS

CASE STUDY

DESJARDINS GENERAL INSURANCE GROUP

INSIDE THE INSURANCE INDUSTRY’S LARGEST AND MOST COMPLEX MODERNIZATION PROJECT

Introduction

Desjardins Group (DG) is the leading cooperative financial group in Canada and the sixth largest in the world, with assets of over $175 billion. Desjardins General Insurance Group (DGIG) is a leading Property and Casualty insurer with a portfolio of over 2 million active policies, $2 billion in revenues and $4.3 billion in assets. This project completely migrated DGIG's systems from an MVS/IDMS platform to a new Windows/Oracle platform. This project took 4 years to complete and according to DGIG, currently stands as the largest successful modernization project done in both North America and in the history of the insurance vertical.

DGIG: Why Modernize?

The main objectives of this project were to achieve:

- A technology platform promoting the convergence of distribution channels
- Efficient, cost-effective infrastructure
- Performance and scalability improvement over legacy systems
- Elimination of obsolescence risk associated with legacy MVS/IDMS systems
- Reduced cost and better scalability for supporting increasing volumes of business
- Advantages in terms of staffing and recruitment, easier to find skills and people to support newer technologies

Project Highlights

Annual operating costs were reduced by **$4.5M.**

Performance improvements of **30%** were achieved in daily processing. Performance improvements of **50%** were achieved in weekly processing with **99.5%** of transactions processing in **3** seconds or less.

All Migration Project Deliverables were delivered on time and on budget. Data conversion was practically perfect with a rejection rate of nearly **0%**: 61 errors on 2.6 billion migrated occurrences.

Over a 30-day testing period, only 66 variances in approximately **1.5 million** transactions, performed daily by 2,500 users (agents), were identified.

Over the course of the migration, **273** business-driven change requests were implemented on the old platform, then integrated onto the new platform with no impact to users or product development teams.

ModernSystems' Automated Conversion and Portfolio Optimization Solutions were used with complimentary partner technologies to migrate the following technologies to a modern, Windows and Oracle environment:

- IDMS
  - over 2.6 Billion records
  - 237k files, over 14k fields
- 4,330 COBOL programs
- 400+ ADS/O dialogues
- 1100+ Easytrieve
- 10,000+ JCL

Overview: Modernization In The Insurance Vertical

The Challenge of Legacy Systems

Most insurance legacy systems have been in place for decades. Most are based on aging development standards with limited functionality and usability. Despite their limitations, many are still relied on to provide mission-critical business functionality. Business challenges perpetuated by these legacy systems can include:

- **Disjointed development and maintenance, leading to "silo-based" usage**
- **Maintenance and staffing costs are high and growing**
- **Compliance and regulatory challenges**
- **Limited functionality and integration of legacy systems limits innovation and growth**
- **Data-driven architecture makes business intelligence efforts either impossible or prohibitive**

Conversely, it takes a flexible, nimble organization to compete in today's aggressive insurance landscape. Reliable technology that is scalable and integration-friendly for growth will dictate an insurance organization's success in this environment.

Because legacy applications were built in a different time, often to serve a single purpose, they rarely meet the criteria necessary to help an organization for the long-term. It's important to understand the specific impact of legacy systems on competitive advantage and business growth in the insurance vertical. The following chart outlines this impact with a focus on core insurance business processes.

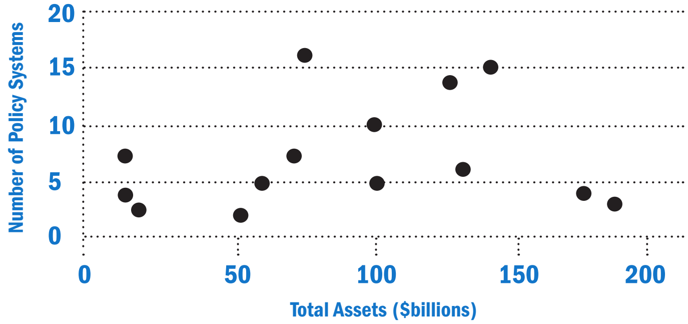
Product Development	Distribution	New Business	Policy Owner Services	Claims
· Long dev cycles · Expensive testing/QA · State-specific regulatory/compliance regulations · Lack of long-term enterprise viability	· Low adoption · Low retention · High cost of support	· Manual processes · Lengthy application processing cycles · Lower quality application processing	· Poor response times · Extended time to resolution · Lack of Single Customer View	· Loss leakage · Problematic benefit disbursement · Long cycle times for benefit processing · Customer dissatisfaction

Overview: Modernization In The Insurance Vertical

Repetitive, Disconnected Systems

According to benchmark data collected by CSC (Figure 1), a typical mid-to-large U.S insurer maintains four or more policy admin systems, and many have portfolios of 10 or more.

Fig.1, Policy Administration Portfolio Size



Sources: AM Best 2011 and CSC Sample of 14 US insurance companies

Multinational insurers are facing similar challenges. Many multinationals have redundant policy administration platforms across countries or regions, owing to earlier growth through acquisition strategies and subsequent governance of each region as a standalone profit center.

Organizations reap the biggest returns when they are able to reuse data and avoid redoing or completely recreating work that's already been done. Without adequate interfaces linking systems, reusability is greatly limited, if possible at all.

Legacy applications simply were not designed for this. With their unique and hard-coded interfaces written with rigid functionality in mind, legacy systems make integration difficult. This further perpetuates existing silos and prohibits sharing of information across departments and line-of-business (LOB) applications.

Disparate Systems Destroy Advantage

SAP released a survey in June 2013 of over 200 insurance executives, conducted at the IASA Annual Conference. Modernizing legacy insurance systems was top of mind for many of the attendees.

Forty-three percent of believed that advanced analytics for greater insight into the business, including predictive and fraud management, are essential for competitive differentiation. Mobile workflows and apps for employees, agents, and customers came in second at 22 percent, followed by delivering a single view of customer and creating a great customer experience across every touch point (19 percent).

In an industry as interconnected as insurance (e.g., customer information first recorded on a policy application can be reused later in completing a claim submission and later in a renewal form), reusability of data should be a top priority.

Without major time and dollar investments (not to mention serious risk) to current systems, integration and reusability may be impossible for companies wishing to remain in a legacy state. Programmers spend so much time maintaining and rewriting legacy code in an attempt to keep up with user demands that they have little if any time to create new or unique functionality to improve business processes.

On the business side, with their hard-to-learn interfaces and lack of functionality, legacy systems hardly facilitate productivity. The outcome is that business users often spend more time trying to decipher the system or actually avoid using the system at all, opting instead to complete tasks manually.

Not only does this cost the insurer time and increase employee frustration (which leads to staffing and morale concerns on the business side as well), but when employees revert to manual processes, information security and access become crucial concerns.

Overview: Modernization In The Insurance Vertical

Compliance Quicksand

Legacy systems were not designed to live up to today's stringent compliance standards like internal audits, Sarbanes-Oxley (SOX) and Health Insurance Portability and Accountability Act (HIPAA).

Many insurers react by making customization after customization to their legacy systems, hoping these changes will push them into compliance. Often, these changes aren't well documented and the ability to maintain them is restricted to "tribal knowledge" amongst a chosen few.

With governmental interest in insurance affairs growing and regulations becoming stricter, it seems clear that sooner or later insurance organizations of all sizes will be impacted. Without systems designed for oversight in place, insurers are in an immediate position of liability.

The Institute of Internal Auditors linked obligations under the Sarbanes-Oxley Act to a variety of specific recommended IT practices, including the upgrade of operating systems and other software "to stay current with security patches and to ensure continuous support for all software in use." Failure to patch systems promptly or continued use of unsupported software can be viewed by courts as negligence.

In many cases, legacy systems have been updated by many hands over the years with relatively minimal testing and release note documentation. Customization of these systems can easily generate an operating deficiency—in SOX terms, this occurs when a control is not operating at it should—or there might be a technical fault present in the system. If any one of these is directly involved with the administration of a company's finances and official support is no longer available, then the company is at risk of a serious SOX violation and C-level management must bear the legal consequences by default.

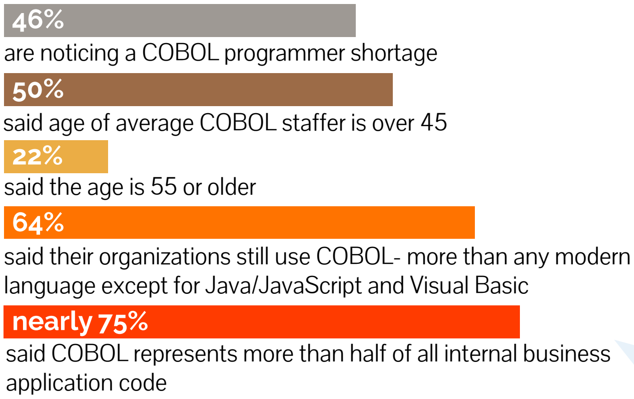
Who Will Help You Climb Out?

For most insurers, legacy systems are responsible for core business transactions and capabilities that represent the livelihood of the organization, so proper care of them is critical. But that care is hard to find- and doesn't come cheap.

Mainframe systems administrators, for example, who are responsible for job schedules, systems security, operating system upgrades and the like, have different skills to the application developers creating the company's business logic in languages like COBOL, Natural, CA GEN, and ADS/O. These languages are the domain of an older generation, many of whom are retiring with no replacement in sight.

Today's technology professionals are focused on modern languages, like Java or C#, taught in rich detail at universities and reinforced globally through online user communities. According to research published by MicroFocus, the largest volume of skilled developers introduced to the job market in 2013 was Java programmers (32%), followed by C# and C++ programmers (16%). In comparison, only 5% were COBOL developers.

Another 2013 survey from ComputerWorld IT further reinforces the reality of the legacy skill shortage:



46%
are noticing a COBOL programmer shortage

50%
said age of average COBOL staffer is over 45

22%
said the age is 55 or older

64%
said their organizations still use COBOL- more than any modern language except for Java/JavaScript and Visual Basic

nearly 75%
said COBOL represents more than half of all internal business application code

Moving to a modern configuration can alleviate compliance challenges and reduce risk around increasingly rare legacy system resources.

Overview: Modernization In The Insurance Vertical

Tackling The Legacy Challenge

So where is the best place to start? Karen Furtado, Partner at Strategy Meets Action says her firm's research shows insurers place highest priorities on addressing the critical areas that may be impeding the business progress. These areas vary for each insurer. "For some, the challenge involves the ability to service the distribution channels. These insurers are focusing on user experience. For others, it is the inability to respond quickly to rate and rule changes, and they are more focused on core rating capabilities. Others face challenges in rolling out new products and programs and bringing them to market, so they are concentrating on core systems and how to deal with product configuration."

The key, she adds, is to approach the modernization in a strategic and planned manner rather than as a tactical project. The following are the choices insurers have for modernizing their systems.

Status Quo

The first choice for any scenario is always to do nothing, to "let it play out". The sole benefit to ignoring the problems of your legacy systems is saving money in the very short-term. As time passes, dealing with legacy systems is more and more difficult – less available expertise, more application development backlog and more money thrown away that could be spent on projects to improve IT efficiency. Doing nothing to improve your legacy system drives up costs in recurring license fees, hardware maintenance, facilities costs and staffing costs.

Third Party "Off The Shelf" Solutions

This approach focuses on replacing legacy application functionality with packages and components available from third party vendors. The positives of this approach include reduced maintenance of source code, as the vendor is responsible for fixing production bugs and implementing new functional enhancements.

However, commercial packages offer standard domain business processes that often differ from the homegrown legacy application. Reuse of existing business logic isn't possible, so some level of business process re-engineering or customization/rewriting of the third party solution is required. Both of these processes can be time consuming and expensive.

Application Re-Engineering

Known as the "big bang approach", re-engineering is the most expensive and risky solution for modernization. It includes requirement capturing, coding, debugging, testing and refining. To recreate the legacy solution's wealth of functionality, with newly written applications requires significant time and effort. Impact on end users and their adjustment cycle is also a major factor. Many analysts consider the failure rate of these projects to fall between 70-80%. The cost, time and risk involved with re-engineering are so high, few organizations choose to embark on large-scale rewrite projects.

Application and Database Conversion

This approach preserves the benefits of the legacy environment while empowering customers to leverage the advantages of newer platforms.

Modern Systems' automated conversion technology guarantees 100% like-for-like functionality and generates fully maintainable Java and or C# code. This means once the legacy application and database is converted, developers can extend application functionality directly. Precious business logic from the legacy system is preserved while enabling deeper integration and customization to meet business requirements. It also opens up a whole new world of scalable resources for system support.

DGIG: Project Risk Factors

An Immovable Deadline

DGIG had to migrate by a specific date to avoid substantial costs from extending its outsourcing contract, to reduce its annual operating costs by $4.5M and maintain the implementation budget. This migration, which included 2 implementations, had to be done in the same fiscal year. If not, this would have had a very negative impact on customer service, reputation and implementation costs.

A Budget Set In Stone

DGIG set a project budget of $45M, a hard number with no upward flexibility. Establishing an accurate inventory of the legacy system was critical for establishing scope. However, legacy artifacts were spread over several departments and applications- many of which were never properly documented.

Operational Consequence

DGIG users had to experience a "Business As Usual" feeling throughout and after the completion of this migration. Any operational impact would slow the project's delivery and create negative perceptions internally about the project's viability. This necessitated an on-going complex data conversion strategy, using the classic metaphor of "changing the tire on a moving car."

No Point Of Reference for Project (or Vendor)

DGIG worked with analysts, vendors and partners but could not find a comparable project with respect to planning, development, implementation and training. Additionally, this was the largest project ever taken on by Modern Systems.

Major Public Visibility

DGIG's project budget approximately $2.7M in tax credits from the Canada's Scientific Research and Experimental Development Program (SR&ED) for:

- Research and development of migration solutions for all components (data, programs, screens, reports, JCLs, schedules, ACF/2 Security).
- Development of a methodology to integrate business project modifications during the migration
- Creation of an automated test environment allowing results to be compared (initial vs. target) to guarantee that the data and functions are identical
- Design, development, research and integration of tools required to ensure that the management of the new platform is equal or better than MVS
- Research and development to provide strength and performance that is equal to or better than MVS
- Creation of an environment whose security is equal to or better than MVS without affecting the code

Code Conversion

Multiple procedural languages, intertwined to deliver complex business logic, needed to be converted into a fully maintainable, object-oriented model. The new model had to keep the business logic intact while meeting performance targets with no user impact.

These requirements ruled out replatforming solutions, as emulators and code containers could not deliver on the requirements of maintainable code or complex integration.

Many code transformation solutions create structurally similar programs in the new language, but then need an additional re-engineering effort to yield desired results. Scoping this additional work was widely considered a risk factor for meeting project deadlines.

DGIG: Engagement Detail

Integrated Partners For Delivery

DGIG teams ran the entire project, from planning to milstones execution to communication. However, we had to work with several other partners on this complex project. One was WAZ Informatique, a consulting services company based in Quebec.

WAZ Informatique is a longtime technology services partner to DGIG, working together since the late 1990's. Over that time, WAZ Informatique accumulated significant knowledge and experience with DGIG's business process and systems, handling projects across multiple systems (infrastructure, applications and data tier) impacted by the modernization project. This uniquely qualified WAZ Informatique to ensure no critical information or requirements were missed during the modernization engagement.

Modern Systems' approach added efficiency for WAZ Informatique. "Their methodology supported our process perfectly, enabling us to work in parallel without confusion," says Frederic Gingras, President at WAZ Informatique. "This alignment eased stress around planning and eliminated role confusion. It also added clarity and simplified knowledge transfer throughout the course of the project."

Assessment

We worked together with the DGIG team to establish a baseline inventory for translation. The next step was aligning our delivery strategy with the progressive migration plan developed by DGIG.

The DGIG team instituted a "bridge analysis" system, developing specific programs to facilitate data sharing between legacy and target systems. This was a cross-functional IT effort across DGIG and enabled systems to be migrated with minimal to zero impact on end users and production systems.

For Modern Systems, it provided additional visibility into how the legacy systems worked together and the expectations end users had for their performance and functionality.

Data Conversion

Once the baseline inventory was established and agreed upon, we worked with WAZ Informatique and DGIG to prepare for the modernization. DGIG had invested heavily in the presentation layer of their applications. The Modern Systems solution exposed C# classes to enable DGIG's technical team to map screens to the business logic layer.

Modern Systems gave DGIG the flexibility of using embedded SQL or I/O routines for the IDMS conversion. DGIG's team chose to convert the IDMS DML commands to embedded SQL to retain the logic of the legacy state in the new relational database. However, not using full SELECTs (i.e. the full ordered data buffer that comes back from the Database Table row) meant the system was doing a field resolution (searching for each individual field in the buffer). This slowed performance to a level that would not meet requirements.

"Our technology is flexible and can support customization as appropriate," says says Pratik Dalal, Modern Systems Project Manager. "We developed an automated solution that leveraged full selects, eliminating field resolution processes running behind the scenes."

"With legacy code now migrated to .NET platform, DGIG was in a position to expose their systems over web using web services," adds Dalal.

DGIG: Project Risk Factors

Converted Application Testing

Alignment between Modern Systems and WAZ Informatique was critical in this stage. “We were unable to test offsite due to DGIG’s security policies,” says Dalal. “Plus, there were the challenges of testing against several integrated systems, many of which were also being built as part of the overall modernization project. WAZ’s experience both in testing and with DGIG systems resulted in well evolved, relevant test scripts and problem descriptions.”

WAZ Informatique’s testing methodology was fairly straightforward, involving Unit Testing, Integrated System Testing and Business User Acceptance. However, project change controls and customization requests challenged the team’s ability to meet the project deadline.

“We would get the converted code, test it, and if there was a change request, add the desired functionality and resume the testing process,” says Gingras. “So having quality converted code that was easily modified was essential.”

High code quality eliminated manual fixes and rework, kept the project on schedule and enabled the WAZ Informatique team to properly test across over 30 integrated systems that leveraged the converted code.

WAZ Informatique heavily leveraged the Business User community as well. “User experience and system performance were the ultimate benchmark for our success,” says Gingras. “Getting a holistic view from the business users rounded out our testing approach. We collected anomalies and folded them back into the Unit Tests. In the end, we delivered an “as-is” solution to the business, meeting transaction time requirements of 3 seconds or less.”

Converted Code Delivery

Desjardins code was not frozen during delivery, so even during migration, code was changing and testing and delivery needed to adjust accordingly. Therefore, modernization, testing, and delivery of converted code was broken into Work Packets. Each packet was broken down into subsystems representing different parts of the business- and were delivered according to the impact the converted code had on that business unit.

“For example,” says Dalal, “work Packet 1 could include General Infrastructure system, because none of the systems could work without it. Work Packet 2 could be the Accounting System and the HR System, and Work Packet 3 could be Payroll.”

For each Work Packet, Modern Systems started with the entry points of those systems. For batch it was all of the JCL. For online, it was the main menu(s) and a list of transactions. “WAZ Informatique verified presence of all required components, and if any were missing, we worked together to track them down and fold them into the plan. In all, there were 7 Work Packets, some containing up to 26 logical systems,” says Dalal.

By the end of the project, Desjardins had 5 full scale code refreshes to support changes requested during modernization. Having technology that could support such requests in an expedient manner was essential for project success.

“When we say we modernized 4300 COBOL programs for DGIG, a big part of those programs were not converted once, but twice, 3 times, and some even 5 times, accommodating all the changes DGIG made in the original mainframe system while we were modernizing the system,” says Dalal.

Each of the 5 code refreshes were validated and tested by WAZ Informatique and DGIG, adhering to the original process and meeting project deadlines.

Best Practices Critical To Success

Executive Buy-In

The management approach relied on the accountability of those responsible, the accountability of IT resources and the involvement of managers and upper management, which ensured the mobilization of resources throughout the project.

Proactive, Participative Management

DGIG mobilized stakeholders during the entire project by setting up interactive means of communication (management committees, newsletters, master plans) as well as during the transition period. Rigorous management of the following activities was implemented:

- Management committees encouraging synchronization, convergence and communication
- Structured accountability at every decision-making level
- Formal information mechanisms (newsletters, minutes from meetings, progress reports during by project team meetings)
- Recognition of successes by IT resources

All change strategies (communication, transition and training plans) were implemented, allowing new business projects to be developed on the new platform even before the end of the migration work.

Strategic Evaluation Of Legacy Functionality

Extensive resources and timelines were applied to assess the project's impact across the business. Legacy artifacts and logic were measured for relevance to future business goals and requirements. Business logic and processes relevant to the business were added to the project plan. Those no longer essential were either refined or removed.

Proof-Driven Vendor Selection

An extensive Proof of Translation and Proof of Concept was required to validate vendor technology. This process also identified aspects of the project that would require internal DGIG resources. It was important DGIG resources maintain aspects of the environment moving forward. A straightforward process of delivery, testing, refinement and training was developed to ensure proper transition.

Committment To Timeline

Change requests during the late stages of a modernization initiative could significantly delay or even derail the project. A 3-month freeze was imposed on business project deliveries during the migration implementation period. Escalation management procedures ensured alignment around critical issues, avoiding ambiguity and misalignment during the various implementation phases.

Thoughtful, Exhaustive Testing

Cross-departmental representation was essential in developing test cases/scenarios for integration and system testing. Testing was started early and involved end users. This proactive approach reduced risk by identifying and solving unexpected integration challenges, underestimated efforts and undetected functional gaps.

Committment to Performance

A core tenet of the project was a seamless migration experience for users. Requirements went beyond duplicating legacy functionality and business logic- the target system had to meet or exceed performance benchmarks of the MVS environment. The new environment had to improve productivity for developers. Solution architecture prioritized performance with attention to redundancy, recovery and database availability. This focus enabled implementation of Agile Software Development and created opportunities for IT to deliver new, value-added services to the business.

Customer Feedback & Conclusions

Business Benefits of New Systems

"The business experienced significant savings and from an IT perspective, we now enjoy simpler upgrade and maintenance cycles," says Serge Grenier, Principal Director of IT and VP of Insurance Technologies at DGIG. "The flexibility of the new platform enables integration of modern technologies, which helps the business retain advantage and prepare for the future."

Engagement Highlights

"Modern Systems invested heavily in communication at the onset of the project," says Grenier. "They committed an executive sponsor and had that sponsor on the floor with our team. This made a big difference not only in establishing a clear process, but in making the teams feel like true partners."

"Modern Systems also committed onsite resources to help with testing and code refinement," continues Grenier. "This came in handy when we experienced performance challenges- their resources were right there to understand our need and customize their solution."

What Could Have Been Better

"The project was a success, but there's always things you look back on," says Grenier. "For us, it would've been organizing our team to better train vendors on DGIG-specific scenarios for testing. A lot of our knowledge share was manual. Having this area streamlined and potentially automated would've saved time."

On Modern Systems: Would You Do It All Again?

"I can definitely recommend Modern Systems with confidence; in fact, I already have," says Grenier. "This was a high-profile project and several entities here in Canada and in the insurance vertical have inquired about our success. Modern Systems invested in understanding the unique aspects of our business and went above and beyond in their committment to the project's success."

"Technically, the company has it together. They were able to respond to unexpected challenges quickly with skilled resources. Therefore, mistakes and rework didn't kill our timeline," says Grenier. "From a communication and professionalism perspective, their process and expertise in doing projects like this assisted in getting over the usual issues international teams might face in a large project like this. From the outset, their executives had visibility into the project and the delivery team made it work."

About Modern Systems

Modern Systems, Inc. is the leading provider of legacy language and database modernization. Leveraging over 30 years of best-practice domain expertise, Modern Systems works closely with its customers to minimize risk and provide a clear path from legacy platforms like COBOL, Natural/Adabas and others to modern solutions like SQL, DB2, Java and more. Modern Systems was chosen by Walmart to modernize the world's largest order system. We've also modernized the world's largest trading platform. Modern Systems has offices in the USA, UK, Italy, Romania, and Israel.

Visit http://modernsystems.com to learn more