**LEGACY MODERNIZATION: A SYSTEMATIC APPROACH**

|  |
| --- |
| INTRODUCTION |

Around the globe organizations have started realizing that they are stuck with legacy back – end systems. Subaru a global automaker had similar challenges and the reasons are simple

* High variable and recurring costs for maintenances, licenses, and new code implementations
* High staffing costs for architectural analysis and code conversion
* Apprehension of interruptions in business – as – usual
* Decades of overly dependence leading to resistance – to – change

Like most organizations who have invested heavily on legacy systems in early 70s and 80s, Subaru was wondering, “To be, or not to be…”

This case study details out the legacy modernization for Subaru by partnering with HTC Global Services. It also discusses successful and cost effective legacy modernization project details executed for Subaru.

|  |
| --- |
| MAKING THE FIRST MOVE |

Subaru requested HTC Global Services to convert mainframes jobs running in JCL and associated with Cobol – DB2 and Easytrieve programs into Java – spring batch technology. The mission – critical legacy system would fetch vehicle line- up data of thousands of automobiles from an external job and then processes this data for scheduling automobiles into the production shop floor.

The success of the project was dependent not just upon systems’ conversion, but also on the stringent project timelines, performance guidelines and budgetary controls. The challenge was to seamlessly convert legacy systems running for 30 years without any interruptions in production schedule. The new would not only implement the requirements of legacy but should also interface with the other legacy systems not being transformed. Even before the implementation Subaru decided to increase the production capacity on converted system by 50% compared to the legacy system.

|  |
| --- |
| DEEP DIVE: |

HTC Global Services team first focused on a solid architectural design that formed the basis for conversion of mainframes components into Java components. The architectural design also detailed out how the legacy business processes, rules and constraints imposed via mainframes will be implemented in Java. Once architectural design was finalized, HTC Global Services adopted a two phase reengineering approach.

|  |
| --- |
| COBOL to JAVA CONVERSION ARCHITECTURE OVERVIEW: |

Subaru Lineup production schedule process automated code/data migration process re-architects the code into three layers: a Data Access Layer (files and database access), a Job Layer (JCL’s), and a Business Logic Layer (steps & programs).

**Framework Components:**

1. Batch Core Engine - Execute core scheduled jobs
2. Batch Web App - For monitor and control job executions

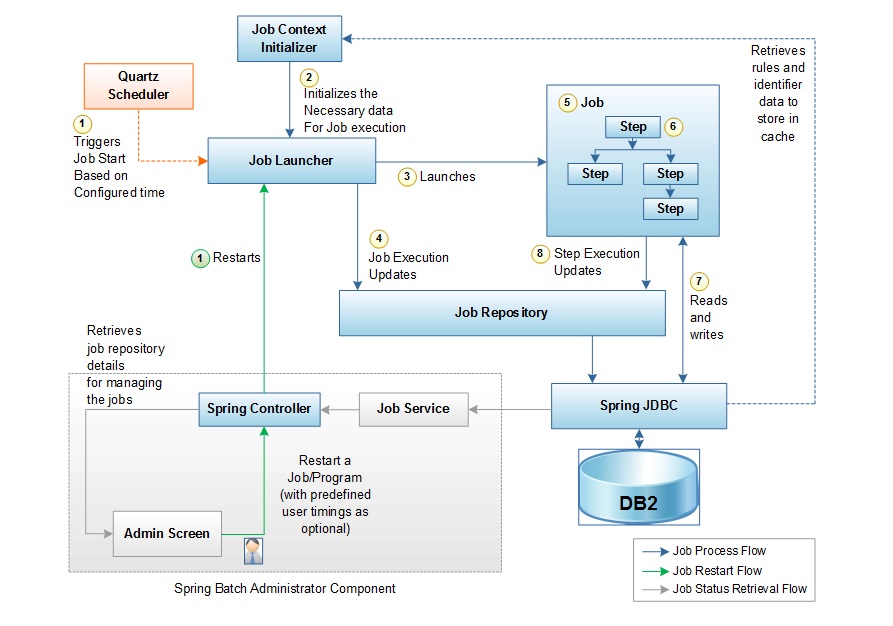
****

Fig 1.0 – Subaru Production Schedule Process Architecture Diagram

Each COBOL program will be converted to a Java class providing the same functionality exposed as a service (Business Logic Layer). All database/files operations from the original COBOL code are externalized in DAO classes

|  |
| --- |
| BASIC MAPPING |

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | **Procedural Language (COBOL)** | | |  | | --- | | **Object Oriented Language (Java)** | |
| JCL | Spring Batch Jobs |
| Programs & Sub-Programs | Spring Batch Steps & Java methods |
| Files & Copybooks | Database Tables |
| File Record Definition | Tables with Version and Record Identifier |
| Host Variables | DTO |
| Copy Members for data base | DAO |

|  |
| --- |
| ANALYSIS & CONVERSION EFFORTS: |

Phase – 01 Cobol analysis team progressively analyzed legacy sub – modules in sequence and created requirements for Java system. The requirements detailed out the business flow of the Cobol programs, and business rules implemented in the legacy code. Since the legacy code was written in over three decades, substantial redundant and archaic code was identified and extracted in the process.

Phase – 02 Cobol conversion team running in tandem with Phase – 01 picked up from the culmination point of phase – 01 team. After analysis of requirements and removal of archaic business logic, latest requirements were incorporated in the Java design artifacts. To gain performance and yet have easy of code maintainability, selected business processes were implemented as database stored – procedures avoiding any delay to bring up processing logic on the presentation layer.

|  |
| --- |
| CONVERSION PROCESS: |

* Converted Subaru Lineup Production schedule Process COBOL System which contains around 10 JCL, 50 COBOL program and 37 Copy Books into Java System with 10 Spring Batch jobs, 45 Steps and 43 tables
* Utilized Open source technologies for migrating from COBOL to Java
* Cost-effective solution significantly decreases the risk and time involved, and enables
  + current developers to continue maintaining and enhancing the migrated application
* Migrated to the new environment automated process retains business logic and application flow, keeping the same functionality. The migrated application follows object oriented principles like encapsulation, abstraction, modularization and loose coupling
* True modernization of the legacy application including replacement of terminal
  + emulation with native web browser access from a Jobs and Scheduler Admin Application to dynamically schedule jobs, start/restart failed jobs from the point, to view metric of Jobs and steps details
* Concurrent and clear Job run status notification for the end user via email (SMS) and logs
* Monitoring and maintaining the Scheduled Jobs become more User Friendly via the Admin Application
* Spring Batch an highly effective open source framework with lot of inbuilt features for executing the refactored business logics of the legacy application effectively into a newer environment
* Carrying forward the vast investments made in the legacy application
* The application can be installed on any operating system where Java runs and on any standard JavaEE server

|  |
| --- |
| WHY HTC |

HTC Global Services showed that it is really possible to convert and modernize legacy systems without a glitch. The new Java – spring batch system is running successfully producing even more cars than legacy system delivered.