

SoftwareMining – Legacy Transformation Case Study

US State Tax Office

In 16 years of implementing complex tax systems, the MERITS project has been by far the most successful implementation to date. The translation of the COBOL code to JAVA using SoftwareMining's translation software was a tremendous success. RSI is an advocate of reengineering legacy systems that meet the majority of the user community's business needs, rather than recommending the wholesale replacement of a working system. Our experience of working with SoftwareMining reconfirmed our view that reengineering enables a business or government agency to reap the benefits of modern technology without the significant risk and cost associated with wholesale replacement

Danny Reeves, Project Manager

A US Local Government agency has a legacy IBM COBOL Application with proprietary extensions. The agency was looking to modernize the system in order to bring the system up-to-date with new technology and reduce maintenance costs. The agency had standardized on Java language for all new development.

Migration of the Legacy COBOL application to Java was seen as an essential part of a much larger picture of modernization activities – with the following benefits:

- Migration of existing application will not result in changes in Business Processes
- The system can continue to be maintained by existing personnel
- Moving from Proprietary to Open system
- Service Oriented Architecture
- Reduce Maintenance costs
- Easier integration of the Legacy Application with the more recent Java developments

The technical objectives were as follows:

- Maintainability of the java code: The new designs should be maintainable by java developers with little or no training
- All further functional enhancements to system should be achieved seamlessly in java.
- The translation framework should easily be adopted to non-functional enhancements: SOA, Changes to screen design,
- The translation should support deployment to large user base.

SoftwareMining CORECT Translation Toolkit was identified to meet all above objectives, and was used for translation of a total of **5 Million** lines of code in phase-1 of the project.

Once the new java system **tested** and **base-lined**, a 2nd iteration consisting of functional enhancements, screen redesigns, and implementation of SOA layer were performed incrementally. The benefit of the approach was that each incremental change could then be tested quickly against the based-lined system.

Implementation Timescales

The following resources were used for the translation and enhancements (Screen-redesign, SOA, and functional changes) of the 5 million line of COBOL code into a new Java application:

- 8 Java developers: used for iteration-1 (Translation, Compilation, Build) and Iteration-2 (SOA architecture, re-design of screens, integration with older parts of system),
- 8 Testers – involved in Unit Testing of new system

An additional 4 testers were involved in User-Acceptance Testing.
The project was successfully completed and delivered in 15 months.

A manual migration of the project was estimated to take some 10 times longer, and 4 times more expensive.