

SOAM™ Case Study: Healthcare Benefits Management Application

Combining Agile SOA with Model Driven Development to Deliver Large-Scale Projects

Introduction

Everware-CBDI is engaged with a large healthcare agency to develop a modern healthcare benefits management application. The system is critical to the future of healthcare in America and has very tight deadlines for deployment. This application is large, complex and must support many stakeholders. It has three main user portals: one for providers to register and provide information about their plans, one for applicants to compare and enroll in plans, and one for government regulators to monitor enrollments and validate eligibility documents. The application is designed for multi-tenancy – complying with PII security concerns while housing the plans from multiple healthcare providers under state-specific regulations in a single system. It also has significant integration requirements with other agency internal systems as well as with external systems. This is a *large-scale* agile development project.

Everware-CBDI is applying our Service Oriented Application Modernization™ (SOAM™) approach to this program under the prime systems integrator and is providing architecture support, UML modeling support and code generation for the application development. SOAM combines (1) Service Oriented Architecture (SOA) to divide domains and solutions into discrete capabilities and services, (2) Agile Architecture and Development Methods for rapid delivery and frequent user feedback cycles, and (3) Model-Driven Development (MDD) to coordinate multiple concurrent teams and generate code. SOAM also encompasses Portfolio Transition Engineering to evolve capabilities from legacy applications; however, this dimension is not applied for this program because it consists of new development.

Challenges

The tight deadlines for deploying the application (the first part for registering plans is already in production and the ability for applicants to sign up for plans is required to be live in October 2013) present significant challenges for design and development. Moreover, the schedule is complicated by the fact that agency officials are still interpreting the healthcare legislation resulting in a nearly continuous stream of new or updated requirements (significant requirements churn) – causing frequent refactoring. For example, nearly half-way through the project the agency decided to adopt NIEM (the National Information Exchange Model) for all external interactions. This was accomplished largely without impacting the development teams. Additionally, every aspect of the project had to be stood-up from scratch, from staffing the team to establishing the various technical environments.

The Solution

For traditional development, these challenges would represent a major obstacle to delivering on time. Fortunately, our team recognized that the speed and agility of the SOAM approach could overcome many of these obstacles and proposed a solution that would meet the requirements of the program office. The first piece is the architecture driven process to identify the services from which the application is assembled. Using an agile SOA approach, the service architecture is developed incrementally on a just enough basis and grows over time with the application. The service architecture establishes the boundaries and specifies the responsibilities of each team assigned to develop the individual components. Early on the focus is on the touch points between the services (the service interfaces) and the specification of the services down to the data used and the methods provided. This solution is

designed as a fully service based application (that is, there is no “free range” code in the system) which provides many benefits for development and ultimately the maintenance of the system.

Managing Large-Scale Agile Delivery

This is a large project with a team of nearly 250 people. Achieving on-schedule delivery via multiple concurrent agile development teams requires co-ordination and sufficient architecture to ensure consistency across them.

SOA provides the dividing lines for the teams, which helps immensely when performing concurrent Agile Development by multiple agile teams. The SOAM agile methodology is based on SCRUM which is characterized by frequent short Sprints (usually 4 weeks in duration), along with parallel architecture Sprints to lay the groundwork. For this application, five to twelve Scrum teams (with 10 – 25 staff each) work in parallel on a continuous basis and development started almost immediately after the project kickoff. Potentially deliverable code is produced in each development Sprint, although it is typically held to be combined with other functionality for release into production.

Keeping multiple development teams synchronized presents a challenge under any circumstances; for agile development the challenge is even more daunting. The SOAM solution to coordinating across teams is based on Model Driven Development. MDD uses UML models to store and manage all of the information about the components to be developed. The models are placed under configuration management and version control, so that all changes are integrated, managed and available to other teams. This coordination would be impossible under the time constraints imposed on this application if the system were developed by hand coding each piece. The result would be chaos. MDD has an additional benefit for development: automated generation of code and other artifacts (such as documentation). For this application, Everware-CBDI’s MDD toolset, MDAgile™ is used to generate the code for all of the services (functional and data access) and extended if necessary. For much of the application, no hand manipulation of the generated code is needed or permitted – this is referred to as “Architecture Owned” code. The user interface (UI) for this system is being developed by hand, although even here some code generation is used to establish an initial framework and the required REST channel services. As a result, for the back end of the system (the services) about 80% is generated and untouched, whereas for the entire application about 50% is generated.

Case Study Results

This project has demonstrated the viability and advantages of the SOAM approach. Due to the aggressive deadlines, coupled with the dynamic business requirements, it is highly likely that the objectives and schedule could not have been met using traditional development techniques. Some of the derived benefits include:

- **Increased developer productivity** – MDD produces a significant initial codebase for each service that is deployable and executable, thereby freeing developers to concentrate on unique business functionality. Our measured improvement approaches a 4x increase over traditional coding methods, with higher quality that reduces rework.
- **Delivery cycle improvement** – The time/effort required to deliver functionality into production has been reduced significantly. The increased developer productivity and the ability to quickly refactor allows each agile Sprint to deliver larger chunks of functionality than would typically be expected from agile teams. This high “velocity” allows early and frequent business review and adjustments.
- **Agile SOA Architecture.** Rather than using the traditional ‘waterfall’ approach, the use of architecture sprints to incrementally deliver the service architecture means there is just enough

architecture to ensure consistency across development sprints and to establish the correct boundaries between them, but without delaying the start of development whilst the architecture is fully completed.

- **Reduced Agile adoption risk** – Agile is a new approach for most agencies and Systems Integrators and many would say that government contracting, staffing and pricing models are ill-suited to fully exploit agile as a methodology. In fact, most commercial organizations adopting agile do so over several years scaling from small projects to larger ones. This project required a newly formed team of nearly 250 people to build a large complex system from scratch in less than 2 years – a risky endeavor in its own right, but made even more so adopting agile at that scale. SOAM provides both the separation and the consistency to allow many Scrum teams to operate effectively – independently and as an integrated delivery team.
- **Reduced impact of refactoring** – With the high degree of business requirements churn, considerable refactoring is inherent in a project such as this one. While embraced by the agile community, refactoring is a specter for most projects and its avoidance is a principal driver behind most SDLCs. Refactoring in any approach is expensive and time consuming, but with SOAM, that impact is greatly reduced. Architectural refactoring is typically the most expensive as it is usually cross-cutting, impacting all Scrum teams. This project has experienced considerable architectural refactoring and yet we have had no need for dedicated refactoring Sprints and have rarely impacted the development teams for more than a few hours. This is almost entirely due to MDD and the fact that most architectural changes are rolled out in re-generated “Architecture Owned” code.
- **Reduced test/repair effort** – Because most “technical” code – as opposed to business logic – is generated using proven patterns and templates, testing and repair activities can concentrate on business processing in the form of end-to-end test cases. The generated code works “out of the box”, so defect research and repair is much easier because the business activity is contained. A phenomenon experienced by most software development projects is that test/repair cycles and code breakage increase exponentially with the size of the codebase. In SOAM projects, we have seen this increase linearly and from a much lower starting point due to the quality of the code produced.

Further information

Service Oriented Application Modernization™ (SOAM™)

Application Modernization should be much more than replacing legacy application “silos” with new silos, or acquiring the latest vendor technology. Modernization should mean delivering open solutions within a genuinely agile architecture designed for responsiveness to business needs - delivering a modular architecture implemented with a portfolio of services. Increasingly, application modernization is being driven by business process modernization and the need to rationalize and align the supporting software portfolio.

Application Modernization as practiced by Everware-CBDI is an architected and engineered process that ultimately produces a smaller overall software footprint which is more responsive to the business. Ongoing, our modernized solutions can then be continuously modernized in a high-value/low-risk manner rather than high-risk/high-cost “big bang” replacement.

SOAM combines

- Service Oriented Architecture (SOA) to divide domains and solutions into discrete capabilities and services
- Agile Architecture and Development Methods for rapid delivery and frequent user feedback cycles
- Model-Driven Development (MDD) to coordinate multiple concurrent teams and generate code.
- Portfolio Transition Engineering to evolve capabilities from legacy applications

See <http://www.everware-cbdi.com/soam> for further information on Everware-CBDI's Service Oriented Application Modernization services.

MDAgile™ Software Development Accelerator

MDAgile™ provides the UML-based tooling that supports SOAM and consists of a *Service Specification Portal*, the *Construction Engine* and the *Transformation Asset Library*. By building an integrated suite of models, all under configuration management, multiple teams can apply agile methods to work on aspects of the overall system without concern that what they are producing will negatively impact other teams.

The flexible *Construction Engine* supports a variety of target environments (language, dbms, middleware, etc.) and the application of standard patterns and frameworks

The use of a UML 2.0 modeling environment improves communication and is an Industry standard (OMG)

See <http://www.everware-cbdi.com/mdagile> for further information on MDAgile™.

Everware-CBDI is a specialized consulting firm dedicated to bringing Agile IT to Federal agencies and large commercial organizations. Our comprehensive approach to IT transformation, known as Service Oriented Application Modernization (SOAM™) leverages our proven, advanced Service Architecture and Engineering (CBDI-SAE™) methodology, agile development methods and model-driven techniques. These techniques are combined with a business capability focus to evolve organizations from stove-piped, hierarchical systems to modern service-based, loosely-coupled solutions that enable responsiveness and agility. We offer transition planning guidance, along with practical, on-the-ground IT development support, to assist organizations successfully accomplish their transformational objectives.