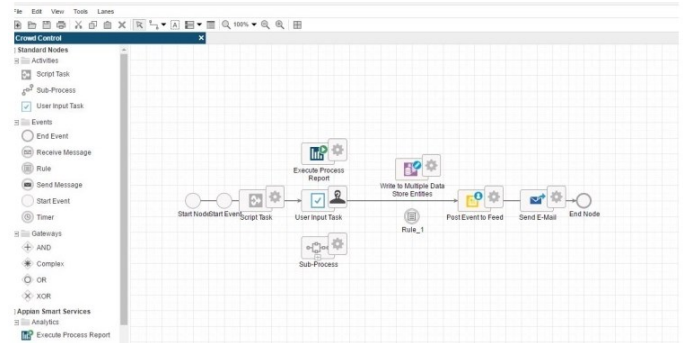


# Federal Low-code and No-code Solutions

6/02/2023

Analytica supported HHS's Centers for Medicare and Medicaid Services to address challenges limiting their ability to exchange data and build analytics supporting a federated operating environment. In collaboration with the client, our team supported architecting, developing, and implementing a federal low-code solution using Appian, AWS RedShift, and Amazon. These federal low-code and no-code solutions provide federal client numerous benefits, including:



- Higher productivity with the ability to build applications quickly
- Better customer experience, and build once, run anywhere experience
- Effective governance and the ability to integrate data seamlessly



## The Challenge

A key organization within CMS was experiencing significant technical debt due to being locked in on a legacy vendor technology platform. This technical debt created issues across the enterprise, from being able to leverage their data as an asset to more advanced needs such as exploring robotic process automation or artificial intelligence. It also forced the agency to use more manual processes and people-based solutions, creating additional costs and operating constraints. Our team engaged at the initiation of this project to oversee operations and management (O&M) under the existing platform as we designed a low-risk, incremental approach to [modernizing federal applications](#).



## Our Approach

In collaboration with the client, Analytica's consulting team defined user requirements and designed a roadmap for transitioning from the legacy system by supporting the development of the new system in Appian, AWS RedShift, and Amazon QuickSight. The program incorporated various components:

**Requirements Definition:** Requirements were defined by capturing them as user stories aligned with SAFe agile practices. Our team utilized techniques such as story mapping and backlog refinement sessions were employed to prioritize and organize requirements.

**Data Modeling:** Data modeling was focused on designing a data model supporting the Agile incremental delivery principle. The data model was broken down into smaller, manageable components implemented and validated in iterations. Appian provides a built-in database called the Tempo Data Store, and our team leveraged this along with AWS Redshift for reporting.

**User Interface Design:** While one team was designing user interfaces using Appian's drag-and-drop Interface Designer, another team focused on designing reporting dashboards using Amazon Quicksight. Usability testing was conducted within Appian's development environment, and feedback was actively incorporated to continuously improve the user interface throughout the development process.

**Business Logic Implementation:** Business logic implementation followed an iterative and incremental approach using Appian's Process Modeler. Complex logic was broken down into smaller components or features represented as process models, decision rules, and task models within Appian. User stories and acceptance criteria were used to drive the development process actively within Appian's Agile Delivery feature.

**Iterative Development:** Align with the iterative nature of Agile by working in short iterations or sprints. Conduct regular sprint reviews and retrospectives to gather feedback and make necessary adjustments to the application. Embrace the Agile mindset of continuous improvement and adaptability.

**Integration:** Appian offers various integration capabilities, leveraging pre-built connectors, web APIs, and Appian's Integration Designer to connect with databases, web services, legacy systems, and more. Our team developed web APIs to exchange information with Amazon Quick Sight and embed reports into a central user portal.

**Security and Access Control:** Security and access control measures were actively incorporated early in the development process using Appian's security framework. Security requirements were identified and actively implemented using Appian's security rules, groups, and permissions. Security testing was conducted actively within Appian to ensure the presence of security measures throughout the application.

**Testing and Debugging:** Testing and debugging were performed continuously using Unit testing, integration testing, and system testing were actively conducted within the testing environment. Test-driven development (TDD) and behavior-driven development (BDD) techniques were actively used within Appian to ensure comprehensive test coverage.

Automation tools within Appian were utilized actively to support continuous testing and debugging.

**Documentation and Training:** Our team developed various documentation and training documents covering the application's architecture, design choices, and configuration settings. Provide user guides and training materials to help users understand how to utilize the application effectively.



## The Solution

Through this work, we have helped the client migrate from their legacy infrastructure to a modern Appian low-code application environment that supports leveraging their data as an asset across the organization. This will help drive greater data quality and operational efficiency while leveraging scalable and lower-cost infrastructure. This success has helped the client also reduce reporting latency over 10-fold on key reports that are used across the organization. From an executive value, our modernization efforts have provided a more strategic view of the data and correlations across the data to improve decision-making and outcomes. As a result of this project, our client is maximizing its business intelligence and analytics investments, enabling enhanced financial and operational governance, reduced risk, and compliance with intergovernmental reporting requirements.

---

## Contact Us

**Phone:** 202.470.4806

**Fax:** 1.888.265.4689

**Email:** bdfederal@analytica.net