Smart Core

Case Studies - Data Landscape

The data landscape uses logical mapping to measure quality and provide tactical outcomes

Due to a long time-to-value on projects an initiative was approved by a leading energy company to explore alternative methods to measure the quality and assumptions of current deliverables.

The initiative had three phases - discovery, automated test and tactical solution.

The objective at each phase was to provide a set of agile deliverables that allow for the process to be iteratively developed with assumptions tested at regular intervals.

Deliverables

The deliverables consist of:-

- *A logical map and diagram that provides audit quality lineage of the process flow broken down into stages*
- *A term map that describes the key rules and implementation code for the target platform*
- *Code that can be implemented onto a database server (SQL Server, cloud or physical)*
- *Logging of the execution of the process for performance and volume tracking*
- *Automated deployment*
- *Visualisation of results via a dashboard for installation to an Apache Web Service*

The Approach

Discovery

Through interviews with SME’s, business terms were captured in a logical map showing relationships between terms. Redundant relationships were then trimmed away until a consistent view of the business processes emerged.

The exercise was performed on the following subjects:-

- Overview – core applications and functions
- Entity – data entities and hierarchies
- Analytics – areas of interest for analysis
- People – for purposes of hierarchy, role and data stewardship
- Workflow – demonstrating a how a workflow would be mapped

The resulting diagrams were then used to frame a discussion as to areas of improvement.

Automated test

A key area of interest is test assurance. The existing test methods have limited access to aged test data and test cases are performed manually. The objective of this phase was to create an automated testing system to map the (scalar only) data from the raw head end to MDMS (Meter Data Management System) and from the MDMS to customer data delivery.

These specifications were logically mapped in Smart:-

- Data Translation. Master Data
- Data Translation. Data Delivery

Where the specifications were ambiguous the implementation code (Mule java script) was referenced. The existing test documentation was then referenced and the test lead consulted for accuracy.

Up to date data was then accessed through approved channels and, using code generated by Smart, run through the logical model. The resulting output files were then checked against the specification. Where data was required from the master data management system, default values were used from the examples in the specification.

The resulting output files could, with the addition of MDMS data, be compared to actual customer files for the same period. This allows for precise testing of output on volume data.

Each stage of the process was completed entirely within a database, with the results of each specification stage being stored in a matching database table. This approach allows for easy tracing and allows for direct comparison of the specification, element by element, to the implementation.

Tactical Solution

Due to the customer need to begin the billing process as soon as possible, the customer indicated that early delivery of un-validated data may be acceptable. A tactical solution to this need is to process readings as they arrive and provide a view on the progress along with a customer output file at any point in the day. Minimum validation can be performed with additional validation tests added to the engine to replicate functions of the MDMS.

Building upon the automated testing phase, the existing test engine was enhanced in the following ways:-

- Batch processing implemented to enable file tracking
- Incremental loading implemented to allow for frequent updates
- Aggregation layer added to provide a progress overview
- Visualisations to allow for slicing and dicing of available information.

This solution allows for a regular (every 15 minutes) snapshot of the un-validated data to be available for delivery that can be tracked through a dashboard. The dashboard and file delivery are low footprint and are based in the Cloud.

A solution was developed through agile practices to meet specific customer requirements and work alongside the existing delivery mechanism. The system continued to function as an automated test platform which grows in capability as each requirement is met.

Conclusion

These phases show how an agile process can iteratively develop a minimum viable product to meet a need, with the outputs from each phase contributing to the next. A system created using these principals regularly delivers under an agile practice whilst maintaining high standards and increasing capability.

CAD4Data is a methodology, linked with a code generator, which allows lists of logical relationships between stages of a business process to be rendered as a system flow, a logical model and executable code. An analyst can explore a requirement using CAD4Data and have data available for testing at each step of the analysis process. A data analyst using CAD4Data can perform the functions of an analyst, developer and tester in one. This allows for rapid change within an agile development.