M365 Upgrade

Client: Anonymous
Business Size: Corporation
Industry: Insurance Company
Country: UK
Technology: C#, MYSQL, .NET, Oracle, VB .NET, XML

Objective: Upgrade Software

The Brief

The client acquired an Aerospace insurance business from two other companies circa 2020.
The inherited software is a suite of connected systems that together support insurance broker front-end functionality, a content management system and support for underwriters with a connection to an IBA system (Insurance Broker Accounting), as well as other systems.

The core function of this software is to provide workflow and document management functionality for the Aerospace division.

OCS has been involved with a number of associated assignments since 2020 for the client regarding this particular ecosystem.

Multiple other vendors are also linked to the support of this Aerospace ecosystem. e.g. Microfocus for Continuous Integration and K2 for Business Process Management.

The Aerospace ecosystem encompasses systems created from using a number of different technologies:

- VB.Net
- C#.Net
- SQL Server
- Oracle
- XML messaging and
- Document management - principally using MS Word and Excel but also includes Visual Studio, (VSTO), customisation

functionality that allows the users access to bespoke functionality whilst within Word and Excel.

The client approached OCS to provide consultancy and resource to upgrade the core system from Office 2016 to Microsoft 365 to ensure compatibility.

In this case study we will discuss the issues found and resolution processes in terms of the technical approach and consideration of the application futureproofing using other tools and aspects.

Background
The client was established in 1927 in the USA and has become a prominent insurance brokerage, risk management, and HR & benefits consulting company on a global scale.
The organisation has a significant international presence, with over 39,000 employees worldwide.

Methodology
Initially there was an issue with running the application and logging in at all in a M365 environment.
This was resolved by a minor change in the email client application classes.

Once fixed, multiple document-related use cases were reviewed, and further application modifications became necessary.

There was also a failure in the application document assemblies, using VSTO customisation into specific document template instances and providing in-document functionality.

Extensive investigation established this was caused by issues in the document templates themselves which were incompatible with M365, as well as the application client and VSTO code solutions.

The issues were for both:

1. Creation of new assembled documents of a multitude of different template types, and

2. Historical documents in the application - some created even pre-Office 2016.

In addition, application stability was a concern that needed addressing.

Resolution of issues was complex, as they had multiple causes, as well as the nature of the application system architecture and operation, including concurrency and non-optimal legacy code – sometimes with many thousands of lines of code.

Solution and implementation
Visual Studio VS2022 in debug mode was utilised in the new M365 environment.
This was used for running the application client and server, as well as VSTO solutions.
These solutions use VB.Net, with C#.Net technologies.
The client application is a Windows application.

The approach then was to go through the application Use Cases using Microsoft documents.
Some of these returned errors in M365 which were not replicable in the Office 2016 test environment.
Again, errors found were addressed by minor changes in the application client classes.

The document assembly functionality required intensive investigation principally in both the application Client and VSTO components.

This highlighted issues with the original document templates themselves that were incompatible with M365.
The parts of the template that needed to be reconfigured in order to ensure they were compatible with M365 were established.
During the subsequent investigation stages, an automated process in the application was developed to manage this issue while the application opened the templates.
This process also resolved the problem with historical documents' incompatibility.

The solution involved a combination of using late binding, in conjunction with more defensive coding in dealing with the assembled documents.
This required multiple changes to both areas to resolve issues.

Informal testing commenced as soon as possible, and work continued to support this in the M365 updated application.

Once formal testing commenced, a number of other areas requiring development attention were discovered and each was addressed and resolved.

Daily standup meetings were held with OCS and the client application development leads. Progress and any potential showstoppers were discussed. Changes to priorities were agreed to mitigate any unexpected delay raised as a result of testing.

JIRA was used to track remaining tasks with an indication of the time expected to take to complete – where possible.

Occasionally, when resolved, certain tasks spawned sub tasks that also needed to be considered.

An Azure CI/CD (Continuous Integration and Continuous Delivery) pipeline was setup to push changes through environments.

Changes and updates were manually deployed into the test environment.

Consultant Contribution

Previous OCS knowledge (since 2020), of the overall application as well as experience in the mix of technologies used was beneficial.

OCS' key role was to assess and resolve the M365 incompatibilities.

This scope expanded slightly to include both current and historical documents operation, as well attention to the overall core application regarding current instabilities.

Both the client and the application document templates needed to be updated to ensure M365 compatibility.

VSTO required updating to both create and edit document assemblies in M365.

Lessons learned

This was a ‘first’ for a M365 software upgrade.

Experience was gained in options, and potential solutions to different parts of this type of activity.

The assignment has provided experience tackling specific issues regarding updating of legacy software to be compatible.

Additionally, in finding the solution to the problem with historical Microsoft Work documents, an understanding of the internal xml structure of Word documents was obtained.