Financial Services Client

Excel to system migration

Problem

As with most of the financial services industry, Excel played a large part in the client's risk calculation process. As a firm, they had a desire to migrate from their existing spreadsheet architecture to a more formalised system to reduce operational risk etc. However, individual traders were very keen to retain their spreadsheets as they were perceived to offer superior value.

The business's requirements from the sheets / system were flexible and changed in response to market conditions[1] and as such, it was important to be able to highlight any differences, not only to the technical team but also the trading team. This visibility was essential in gaining the confidence of the traders that the system would perform as required in the real world.

It's worth noting that the organisation had, to a large extent, managed to standardise a lot of the spreadsheets which were used so that the majority of the sheets which were used for different trading books were the same template sheet with different parameters.

The system that was being migrated to had support for running custom batch jobs.

---

[1] Additional variations of trades, altered model configurations etc.

Solution

It's worth noting that the batch processes that the system authors had already set up stored their results in a repository which was programmatically accessible.

The solution here was a multi stage process:

1. Extend the existing spreadsheet template's EOD macro so that all of the sheet's result types[2] could be written to a network share as part of said process[3].
2. Create an additional batch job[4] to be run after the system's EOD runs had been completed, which could process all of the books found in the aforementioned network share and perform comparisons. These comparisons took the form of
   a. Load up the CSV files exported by Excel
   b. Source the system's results from the results store
   c. Transform the results to a common format[5]
   d. Perform the comparisons[6]

The generated results were then uploaded to Conical in the form

Business Area\Region\Trading Desk\Book\
- PV,Risks
- P&L
- Middle Office feed #1
- Weekly VaR scenarios (for appropriate days)
- etc.

The output format of the results varied depending on the use-case, but where consumption by traders etc. was required, results were additionally presented in tabular form as that's what they were used to.

Once the nightly upload was completed, the batch job sent out an email with a link to the result set in Conical to both the tech team and the users.

For the technical team, this had the advantage that they could see the current status of the reconciliations first thing in the morning and could start work investigating any differences immediately without having to perform large numbers of manual checks. The ability to re-run

---

[2] This initially covered just PV and basic risks, but was subsequently extended to cover the downstream feeds for various control functions so more of the system's footprint could be automatically reconciled.

[3] This was performed by either serialising existing named ranges to disc as a CSV file or by creating additional named ranges and then subsequently serialising them. The aim was to minimise the changes to the spreadsheet as much as possible so that they [the changes] could be made to all relevant sheets, even those which weren't using the same template as the others.

[4] Note that this functionality was also made available to the system's UI interface for individual books so that the comparison functionality could be run interactively intra-day allowing for rapid investigations and turnarounds.

[5] Both Excel and the system had the results available at the trade level, but were represented using a different schema. The tool normalised the data for comparison purposes

[6] Zero tolerance was used as both the sheet and the system were supposed to be using the same underlying analytics / data and so no differences were expected.

the reconciliations in the risk system's GUI for a subset of the total books also meant that they could try to resolve the problem locally, whether it was code or configuration, and be confident of the impact that it would have when the candidate change was released.
For the traders and other consumers, it allowed them to be able to track progress themselves without needing to wait for status updates from project managers etc. This meant that any interested users, typically the downstream systems, could start their work first thing in the morning rather than waiting until mid morning when the previous manual reconciliation system would have started to show results.

For these downstream systems, the increased visibility of differences also allowed them to be proactive when it came to differences and perform their own investigations[7] as to which they thought was correct and which they thought was incorrect. It was frequently the case that the reference spreadsheet implementation turned out to be incorrect and the new system was correct.

---

[7] They were also given access to the risk management system so that they could run the same tools as the developers could. This usage also led to a large amount of very useful feedback into the development process of the RMS to make it more useful to the end users.

Summary

By adding a few extra exports to the existing EOD macros in the current spreadsheet, we were able to automate the process of generating full reconciliation data overnight so that:

- QA could perform analysis work all day rather than a lot of grunt work[8]
- Developers could rerun the reconciliation process locally to test their fixes
- End users could see a constant, up to date, full project status without much overhead

---

[8] Historically, the QA team performed a lot of manual reconciliations daily with the results starting to come through mid-morning and then they were able to start drilling into understanding why there were differences.This led to both them carrying out a relatively low percentage of 'value added' work as well as being bored from the manual testing required.