Risk Integrated

CE
Compliance Executive
> The Daily Information Source for Compliance Executives

Running the Company on Spreadsheets

Dr. Chris Marrison, Founder and CEO, Risk Integrated

Most financial organizations are heavily dependent on internally-developed spreadsheets. Spreadsheets are nimble tools for quickly developing complex financial models and immediately viewing the results. Their main strengths are their flexibility and their ease of use; demanding little or no formal training in programming. Upon these quick victories users eagerly add layers of sophistication and complication. Versions of the same core spreadsheet start to pop up everywhere and organizations quickly end up with spreadsheets escalating into mission-critical applications or databases.

We are all familiar with the day-to-day small frustrations of working with spreadsheets. A forgotten closing parenthesis or deleted cell yields results of "#N/A" and "#REF!". On a grander scale, there are headline-grabbing disasters such as Allied Irish Bank's loss of $700 M due to a trader manipulating a spreadsheet to hide his losses, and a few years back Fannie Mae misstating earnings by $1 Bn due to a spreadsheet mistake.

However, the biggest losses arguably come from accumulated day-to-day operational inefficiencies. Spreadsheets containing the organization's most vital data and most sophisticated analytics are often stranded on desktop PCs, inaccessible to the rest of the organization. This makes it difficult to get a single picture of what is going on in the organization. Also, by their nature, spreadsheets are easy to change. They quickly become non-standard, such that senior management are never quite sure that two results are comparable.

The problems associated with spreadsheets fall into four classes: problems in building them, problems in using them, trying to run them as applications, and, finally, data management. The problems in building spreadsheets accurately has traditionally been addressed by training and policies or procedures checking. Recently, new software tools have been developed which monitor and control any changes made to the spreadsheet. There are now several spreadsheet-aftermarket audit tools to assist construction.

The problems with using spreadsheets stem from people making unauthorized changes, using them incorrectly, or feeding inputs into them that are wrong. Several approaches have been introduced to mitigate these risks. The earliest were within the spreadsheet application itself whereby calculations could be locked and data inputs could be validated. Now, third-party vendors are offering solutions which centrally monitor the use of spreadsheets, and make standardized spreadsheets available to all users. Monitoring the use of spreadsheets is a good deterrent to users who know that any illicit actions will be recorded and traceable.

The underlying principle is that each programming tool should be used for what it does best.

Solving the problems of running the spreadsheet as an application and managing the data inputs, outputs and storage requires a radically different solution. For applications with many users, large amounts of data and heavy processing requirements, the traditional procedure has been: (i) first build a prototype spreadsheet; (ii) when the calculations have been proven in the spreadsheet, translate the calculations into a robust programming language such as C++; (iii) finally, link this application to the enterprise databases, reporting framework, and user interfaces. The problem with this approach is that it takes a very long time to get on the IT department's priority list, have a programmer understand the requirement, change the sequence of calculations to fit the new language, program it, and then have it checked and de-bugged thoroughly by the original analyst.

From the analyst's point of view, this is just implementation, but it can be a multi-million dollar nightmare that never quite works as intended. By the time it is implemented, the original users have lost faith and interest. This translation task is so daunting that many applications that really should be run as enterprise systems are left for years as spreadsheet applications on someone's desktop machine, held together by Visual Basic macros and plain manual labor.

Risk Integrated faced this problem when trying to implement its complex cashflow simulation models for many banking clients. The Risk Integrated solution was to create the Enterprise Spreadsheet Platform (ESP). ESP embeds the spreadsheet inside a C++ wrapper application, callable from within the enterprise computing framework. The C++ layer allows the spreadsheet application to be linked to the bank's main servers, centralizes all computations, and lets users access them via the web. End-users interact only with the application layer via their web browser and never have direct access to changing the core spreadsheet. Only the organization's designated expert superusers are able to view and modify the underlying spreadsheet analytics. These superusers view and manipulate the familiar spreadsheet format rather than lines of programming code.

The underlying principle is that each programming tool should be used for what it does best: Excel for financial modeling, C++ for computationally-intensive statistical calculations and data transport and enterprise databases for data storage, and a web-based framework for user interfaces. These new solutions bridge the gap between scattered spreadsheets and enterprise applications. They bring the benefits of transparency, flexibility, security and operational efficiency without the cost of creating traditional enterprise systems. Most importantly, with spreadsheets properly controlled, management can be confident that they know what is going on.