

Identifying Enterprise Alternatives to Spreadsheets

Rachel Beadie, Technical Consultant

Project Performance Corporation

August 2009

Abstract

Spreadsheets offer a highly functional data solution with low barriers to entry. They are quick to implement, inexpensive, and can be put together by end users with no software developer intervention. A spreadsheet requires no additional hardware investment to implement, generally requires no additional software purchases (as most users will already have the software installed), and can easily be shared among colleagues for collaboration.

However, there are several features of enterprise data solutions that spreadsheets cannot easily provide. Once a spreadsheet has become mission critical, problems related to multiple versions, uncontrolled access, and complex references and formulas can render a spreadsheet solution unmanageable. This white paper discusses the basic functions of spreadsheets and explores four enterprise data solutions to replace spreadsheets.

The First Step: Identifying Requirements

Because a spreadsheet-based solution can be implemented quickly and without software developers, it has the appearance of not being a software development project at all, which commonly leads to skipping the time-consuming but critical requirements analysis and development process. After all, in the event that a new requirement is identified, users can simply add a new row, column, or formula. For this reason, when a project to upgrade or replace a spreadsheet-based solution is started, the first step must be to perform a detailed requirements analysis. No matter how mature the existing spreadsheet solution may be, it probably cannot sufficiently describe the requirements for the enterprise solution, as there are likely steps that occur outside of the spreadsheet. Additionally, the underlying business processes should be reviewed, as some business processes may have been defined or modified based on what the users were able to perform in a spreadsheet. Business rules or processes defined by the software may not always be bad, but they can become entrenched, so caution is recommended.

Exploring Basic Functions of Spreadsheets: Strengths, Weaknesses, and Alternatives

The first step in discussing the requirements of a spreadsheet-based solution is to identify the functions it supports. Because a spreadsheet is so familiar and conceptually simple, most users are unaware of all of the different functions that it performs, such as data storage, data collection, analysis, and reporting. However, to develop complete requirements and identify an alternate solution, it is important to look at how the existing spreadsheet solution is performing in each of these areas in order to accurately and completely represent the requirements of the system.

Data Storage

Strengths – At the most basic level, a spreadsheet stores data. This is the first function that any spreadsheet ever performs. A user identifies a group of facts that needs to be kept together, preferably in some organized fashion, so the spreadsheet is born. Spreadsheets do a fairly good job of storing data, though the files are not very efficient at storage, consuming more space than the data strictly requires.

Weaknesses – In addition to potential issues with the size of uncompressed spreadsheet files, spreadsheets are also limited in their ability to represent relationships. Spreadsheets can represent simple relationships (between the data in the rows and columns), but cannot really model data that has multi-dimensional relationships. Often, the approach used to store multi-tiered relationships can permanently impact the ability to perform crosscutting analysis.

Alternatives – The most traditional alternative to storing data in a spreadsheet is to use a relational database management system (RDBMS). A RDBMS is specifically designed to store data efficiently and reliably, to have built-in backup and recovery capabilities, and to model any type of relationship easily.

A second alternative for storage is a centralized document repository. In this model, the data is still stored in the spreadsheet, but the spreadsheet file is stored in another type of database called a document repository. A document repository, like a RDBMS, provides a centralized storage location, reliable storage, and backup and recovery capabilities. However, as the data is still organized in the spreadsheet, the limitations on modeling complex relationships remain.

Data Collection

Strengths – Another primary function of spreadsheets, related to data storage, is data collection or data entry. Spreadsheet applications are designed to streamline the data entry process, especially for users familiar with the application. The built-in cut-and-paste, drag-and-drop, sorting, filtering, and auto-fill features of most spreadsheet applications offer very flexible and sophisticated data collection features with no development investment.

Weaknesses – A spreadsheet solution begins to become complicated when there is a need to gather input from multiple users concurrently (rather than serially). A spreadsheet can be kept on a shared drive or e-mailed around and changes can be consolidated using the “Track Changes” and “Merge Changes” features, but this method is time-consuming and prone to errors. It is also difficult to manage access to data in a spreadsheet at any level more granular than the entire document.

Alternatives – Data that is naturally partitioned can be divided into discrete data sets, each with one owner or group of owners, and can be partitioned by storing the data sets on separate worksheets in a workbook or in separate spreadsheets. In this scenario, multiple users can edit the data concurrently without the fear of merge conflicts because they can each be working on a different worksheet, greatly simplifying the merge process.

To take this solution a step further, partitioned data sets can be stored in different documents and managed in a document repository. Document repositories, in addition to providing centralized storage, can easily facilitate collaboration, allowing users to lock data for editing by checking out a document and preventing merge conflicts. Document repository systems also allow an administrator to define and manage permissions on documents.

A RDBMS facilitates data collaboration at an even more granular level. A RDBMS is designed to allow a user to check out, or lock, individual pieces of data during a transaction. Access permissions on data in a RDBMS can also be managed at any level of granularity. However, a RDBMS does not have inherent data collection capabilities, so it is necessary to build or buy a data collection tool to provide this functionality.

Analysis

Strengths – The analysis features of spreadsheet applications are most likely what endears the product to its super-users. Experienced spreadsheet users can use built-in analysis functions to answer almost any question about their data, and even inexperienced users can probably get answers to basic questions about their data.

Weaknesses – Though spreadsheet applications have powerful analysis features, combining the analysis capabilities with the data storage capabilities can lead to the proliferation of versions, as each user may feel the need to save a copy of the spreadsheet when the analysis parameters are modified. Having multiple versions of an analysis spreadsheet is not inherently problematic, but when that same spreadsheet represents the data storage model, multiple versions can lead to data inconsistencies that impair an organization’s ability to conclusively answer questions about their data and that can be a challenge to resolve.

Alternatives – When a spreadsheet will be used for sophisticated analysis, the core data should be moved to a separate worksheet. If multiple people are using the spreadsheet for analysis, then locking the data worksheet should be considered. Although this can result in complex formulas and references, performing this partitioning can prevent the proliferation of data inconsistencies.

To control the number of versions of a spreadsheet, even one used primarily for analysis, a document repository can be used to store the spreadsheets.

A more sophisticated solution is to use a business intelligence tool suite. Business intelligence suites usually use data stored in a RDBMS. While there may be a high learning curve and a high initial investment, business intelligence suites offer a wider range of analysis tools and features than spreadsheet applications and they can ultimately decrease the amount of time required to perform analyses that were formerly managed in a spreadsheet.

Reporting

Strengths – Though rarely thought of as reporting tools, spreadsheet applications are fairly adept at producing reports. Whether the need is simply for a printout of all elements in the spreadsheet, or a report includes analysis and filters, the spreadsheet application provides formatting and layout features that can be manipulated easily to produce a report for output. The strength of the analysis features, combined with the ease of reporting, makes spreadsheets a common and widely used management tool.

Weaknesses – When a spreadsheet document serves multiple reporting functions, the potential for the proliferation of versions is high. Again, it is not inherently bad to have multiple versions of a reporting spreadsheet, unless the data being reported on is also stored in the same document. When the data is stored in the reporting spreadsheet, even the proliferation of worksheets within a document can lead to data inconsistencies. Additionally, a common feature desired in a reporting solution is the ability to view previous versions of a report. This feature is not an inherent capability of a spreadsheet.

Alternatives – To prevent data inconsistencies, the data used for reporting should be stored in a separate worksheet or document from the one used for data presentation. Although references can be time-consuming to establish, the value gained from consistent data is well worth the effort. To provide archiving capability, a business process can be established to “publish” reports to PDF format at appropriate intervals.

To provide further control and archiving capabilities, the spreadsheets (those used for data storage as well as those for data presentation) can be stored in a document repository.

The business intelligence tool suite is, again, a more sophisticated and versatile alternative. Business intelligence suites offer more flexible and more advanced tools for report design and layout and include features such as scheduling, publication to the Web or by e-mail, ad hoc capabilities, and dynamic drill-down in reports.

Four Alternatives to Spreadsheets

For any software development project, analysis and development of requirements should be the first step. Sound requirements are necessary to produce an effective and timely solution. In some situations, the most effective technical approach may not be clearly evident, and detailed requirements

will help to identify the solution. After the requirements have been analyzed and documented, the following potential alternatives to spreadsheets can be considered:

- Spreadsheets
- Document repository system
- RDBMS with Web data collection and business intelligence suite
- A hybrid approach: RDBMS with spreadsheet data collection

Spreadsheets

Although a spreadsheet solution has some limitations, it is not an inherently bad approach. For a relatively small user group (around 20 users) and a relatively small quantity of data (around 1,000 data points) that does not have extremely complex relationships, the spreadsheet solution can be a quick and cost-effective approach. However, to ensure data quality and consistency, the data storage function should be separated from reporting and analysis functions by using separate worksheets or separate documents.

Document Repository System

If version control of the spreadsheet is a significant issue, including a centralized document repository can provide that functionality. Because the storage, collection, analysis, and reporting functions are still essentially provided by the spreadsheet application, it is important to design spreadsheets to ensure that the storage function is distinct from the reporting and analysis functions; although the data collection function can be combined with the storage function or kept separate, depending on the system requirements. Because most of the functionality of the system is provided by the spreadsheets, this approach is still not well suited for large user groups, complex data relationships, or very large quantities of data.

RDBMS with Web Data Collection and Business Intelligence Suite

For large user groups or for data with complex relationships, it will likely be necessary to move away from spreadsheets toward a relational database. Complex relationships can be easily modeled in a relational database, and the ability to grant numerous users concurrent and consistent access to data can eliminate the need to manage multiple “versions” of a spreadsheet. For a RDBMS solution, data collection is usually provided by a web-based application designed specifically for data collection. A COTS (commercial, off-the-shelf) product can be used for the data collection and the database, although a more costly, custom-designed and -built solution may be required for unique or highly specialized applications.

For the RDBMS solution, the most effective approach to providing reporting and analysis functions is the use of a business intelligence suite. There is a wide array of options for business intelligence tools on the market, ranging from relatively simple and inexpensive to sophisticated and costly. Even for a specific tool, the reporting and analysis solution can vary in cost and complexity depending on the requirements that must be met. As reporting systems can be expensive to develop and deploy, requirements for reporting and analysis must be detailed and specific in order to control costs.

A Hybrid Approach: RDBMS with Spreadsheet Data Collection

An interesting variation on a RDBMS solution has been developed and deployed by Project Performance Corporation for two different projects. In this hybrid approach, the data collection is performed in spreadsheets while the data is stored in a relational database. The spreadsheets are constructed by a utility called the SSCD (Spreadsheet Constructor/Deconstructor) from data and metadata stored in the database. After the user edits the data in the spreadsheet, the SSCD then parses the changes and modifies the data in the relational database. The current deployments use business intelligence suites to provide reporting and analysis functionality, although reporting and analysis spreadsheets could also conceivably be generated by the SSCD.

About Project Performance Corporation

Project Performance Corporation, part of the AEA group, is a management consulting firm offering world leading expertise in environmental and IT and management solutions for top government, non-profit, and private sector decision makers worldwide. At Project Performance Corporation, we solve a wide variety of problems for clients by helping them improve the way their organizations function. From leading-edge solutions for optimizing the use of information and institutional knowledge, to highly effective solutions to address energy usage and climate change, we specialize in delivering effective, reliable answers to the most complex challenges. At the heart of it all are our people — innovative thinkers well versed in business processes and drivers, and an unequalled focus on delivering quality products and services.

About the Author

Ms. Rachel Beadie is a Knowledge Leader in Project Performance Corporation's Master Data Management and Data Quality practice. With over fifteen years of experience in database design and development, Ms. Beadie specializes in developing data solutions tailored to the business needs and data quality concerns of companies and government agencies. She has been certified by Oracle, by Microsoft as an MCDBA, and is certified as a Project Management Professional (PMP) by the Project Management Institute (PMI).

For more information, contact:

Rachel Beadie, Technical Consultant
Project Performance Corporation
1760 Old Meadow Rd., McLean, Virginia 22102
p: 703.748.7000
e: rbeadie@ppc.com
www.ppc.com