

Shedding Light on Shadow IT: Is Excel Running Your Business?



Hired Brains, Inc.

By Neil Raden

January 2005

ABOUT THE AUTHOR

Neil Raden is the founder of Hired Brains, Inc., <http://www.hiredbrains.com>. Hired Brains provides consulting, systems integration and implementation services in Business Intelligence, Data Warehousing and Performance Management for clients worldwide. Hired Brains Research provides consulting, market research and advisory services to the Business Intelligence, Data Warehousing and Information Integration industry. Based in Santa Barbara, CA, Raden is an active consultant and widely published author and speaker. He welcomes your comments at nraden@hiredbrains.com.

SITUATION

Shadow IT, those people performing IT functions but not part of the mainstream IT organization, have been found to be as much as 78% of the size of the total official IT staff¹. The existence of Shadow IT implies a failure on the part of IT to provide all of the services to meet their clients' needs, and the problem is universal. This is especially acute in Business Intelligence (BI), which is the largest segment of Shadow IT. The vast majority of knowledge workers perform their modeling, reporting, number crunching and sharing with spreadsheets and personal databases, often in spite of the existence of IT-sponsored BI efforts.

Shadow IT doesn't usually attempt to do the core IT processes, like networking or security, or even the core applications, like ERP or CRM. For the most part, Shadow IT groups fill in the blanks left by IT, such as reporting, specialized modeling, and data capture from external sources. Shadow IT also covers analytics, especially budgets, planning and other aspects of what has come to be called Business Performance Management (BPM) and BI. These are areas that have been historically neglected by IT and enterprise systems software vendors, who focus more on good architecture and scalability, often to the exclusion of functionality at the knowledge worker level. The evidence of this disconnect is the unabated proliferation of spreadsheets despite huge investments in enterprise systems. The same result can be seen with data warehousing, where BI tools have had little impact on spreadsheet use.

Often, without being able to articulate why, users will shun these enterprise solutions and good architecture in favor of the ability to get their work done. A coherent architecture is a desirable trait to technologists, but knowledge workers are interested in tools that work and can perform for them. Queries per day, terabytes, metadata and APIs - these are all peripheral issues to knowledge workers. What is important to them is saving time and doing so in an environment that they can easily navigate. The requirements of time and simplicity translate into this: does the application have an expressive interface that is simple and consistent across all of the functions and does it have the smarts and the horsepower to solve difficult problems, end-to-end?

Employing a spreadsheet, namely Microsoft® Excel, is the convenient answer to that need. Excel resides on most users' desktops and is the natural environment for filling the gap known as Shadow IT. It is clear that what is best for the organization, long-term, and what works for the people on a day-to-day basis, are very different.

¹Shining the Light on Shadow Staff: Booz Allen Hamilton," CIO, Jan. 2, 2004, <http://www2.cio.com/consultant/report2085.html>.

PROBLEM

There are over 150 million business users of Excel worldwide, and a large proportion of them are devoted, at least part of the time, to entering data by hand, extracting data manually from other systems and functioning as report servers. Despite intensive efforts to provide centralized BI solutions, the uptake of BI as a replacement for manual methods like spreadsheets has been insignificant. Most BI tools today are designed to operate as part of a data warehouse strategy and have inherited data warehousing's read-only approach. They are suitable for creating reports and analyses of historical data, but they are weak in modeling, data entry, robust calculations, iterative and incremental calculations. Furthermore, they cannot handle real-time or right-time data. Spreadsheets pick up the slack in all of these areas, but oftentimes the solutions are not very elegant.

Part of the problem is that BI is often performed in a perfunctory manner as part of data warehousing initiatives, focusing more on the data management and database aspects and giving only minimal thought to analytical work processes. From a user's point of view, there is little incentive to disassemble years of application development in personal tools only to migrate to a new environment that offers less than 100% of the replaced functionality. Translation: spreadsheets work for people even if they do not comply with standards, best practices or the mandated approach. In addition, IT organizations have been focused on data management from an enterprise perspective and have not been effective in understanding the workflow and informational needs of knowledge workers.

So why is this a problem? One reason is that collaboration, reuse and uniformity are difficult to implement because the tools are not designed to be used that way. There is a broad range of skill and proficiency possible with the use of spreadsheets, meaning power users can create structures that cannot be maintained by others and the relatively unskilled users are capable of creating devastating errors that are difficult to trace. There is no concept of "version control" or any other feature of a professional software engineering environment, even though the tools are being applied as development tools. This leads not only to potential error, but most often, long set up times as new versions or scenarios of the models are introduced, such as copying and renaming files, rolling time periods and modifying reference information, such as hierarchies.

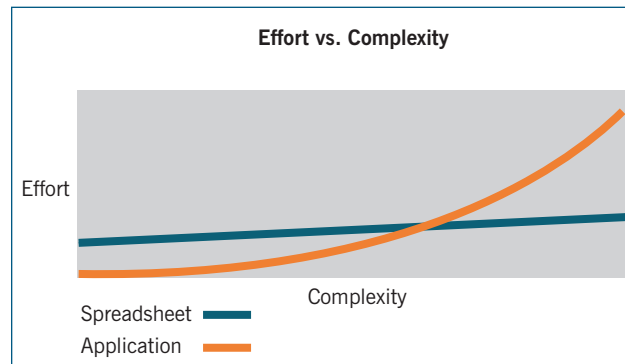
This problem is complicated by the fact that people tend to be more cognizant of the cost and risk of development than they are of ongoing maintenance. The ongoing effort to support a complicated system built with components that were never designed to operate in a collaborative manner is very significant. Maintaining information and especially reference information (look-ups, hierarchies, etc.) in tools that are, for example, positionally organized (spreadsheet notation AA221D\$21) rather than referentially organized (WHERE date is 11/04/2004 and CUSTOMER_NAME is "Yahoo") is devilishly difficult.

All of these factors affect the cost of Shadow IT in organizations, a problem measured in tens of millions of dollars, or more.

Benefits and Drawbacks of Spreadsheets

Based on some of our own current research², we identified a number of qualities of spreadsheets that make them indispensable: expressive, inexpensive, universal, autonomous, fast and portable. However, each of these qualities carries with it some serious drawbacks:

Expressive: Spreadsheets offer the broadest range of features and manipulations for defining calculations and models of any tool ever invented. In a single day, any literate person can learn how to create a spreadsheet of rows and columns, enter numbers, define calculations and follow wizards to format the output. The range of models possible in a spreadsheet is virtually limitless. **Drawback:** In fact, it is precisely because of their expressive qualities that spreadsheets are so often applied to problems beyond their scope, especially those that require systematic properties beyond the models themselves, such as security, integrity, version control and workflow, to name a few.



² Intelligent Enterprise, "Dashboarding Ourselves," September, 2003

Inexpensive Start-Up Cost: Depending on how you look at it, the start-up cost of a spreadsheet is zero, unless you include the developer's time, which may not seem significant until you evaluate it in total. **Drawback:** The problem is that the long-term maintenance and support costs of these spreadsheet applications are quite high. Forcing what is essentially a personal productivity tool into the role of departmental or even inter-departmental modeling, analysis and reporting tool causes the complexity of the application to rise rapidly. The cost of using spreadsheets is very low when the complexity is low but increases exponentially when the complexity increases.

Universally used: Spreadsheets have a very shallow learning curve, meaning it is easy to get started and use them for something of value. This is the reason they are so widespread. **Drawback:** Because the functionality of spreadsheets is so vast, there is a great variety in the level of skill and technique in the way they are used. This makes it extremely difficult to maintain standards.

Autonomy: The desire to operate apart from the organizational systems, free from restrictions and limited flexibility, is powerful. A spreadsheet provides a fantastic range of functionality for creating and distributing models, reports, data and commentary. Despite best intentions, there are always things that need to be done that either can't be done in the official workflow or are too time consuming. **Drawback:** Because of its autonomous nature, any spreadsheet application, report, data or extract must be viewed suspiciously, as it exists beyond the reach of the operational security, standards and controls. Adding these controls is a complicated, time-consuming and imperfect process. In addition, because of the autonomous state, data extractions into spreadsheets are often hard to maintain, slow and unreliable.

Fast: There is no faster way to build an analytical model and perform calculations than with a dedicated spreadsheet running on a powerful PC. **Drawback:** The speed of creating spreadsheet applications has to be evaluated against the entire amount of time it takes to prep and set up the models to run. Copying and renaming files, performing searches and global replaces on names in macros and manually re-keying data all take a considerable amount of time.

Portable: Because it is a PC-based tool, it can operate even when disconnected from the network. **Drawback:** The same issues as were listed above in the section on Autonomy apply here. In addition, it is even more likely to involve duplication of work, handling of files and directories, out-of-date information and manual data entry.

IMPLICATIONS

Spreadsheet software was designed to be easy to use and expressive, allowing anyone to simply enter text, numbers and formulas. A drawback to this all-in-one architecture, though, is that it is virtually impossible to expose, for example, just the model logic. Auditing a large, complex spreadsheet application is a manual, tedious process, and a powerful motivator for not performing it at all. Reports and analyses from a spreadsheet cannot be validated without a great deal of effort; the validation has to be repeated endlessly because there is no version control.

Some of the implications of Shadow IT, and spreadsheets in particular, are:

Wasted Time: Shadow IT adds a huge hidden cost to organizations, comprised largely of non-IT workers in finance, marketing, human resources or sales administration, who spend a significant amount of time wrangling data. With the effort of manual keying, building and maintaining spreadsheets and personal databases, and filing, searching and disseminating data, these professionals may spend only a fraction of their day doing their “real” job. The rest is consumed by data activities that should have been largely eliminated by the data warehouse and BI.

Inconsistent Business Logic: Because each spreadsheet application encapsulates its own definitions and calculations, it is very likely that inconsistencies will arise from the accumulation of small differences from one version to another and from one group to another, as spreadsheets are often copied and modified. In addition, errors large and small that occur from either lack of understanding of the concepts or incorrect use of the spreadsheet frequently go undetected due to a lack of rigorous testing and version control.

Inconsistent Approach: Even when the definitions and formulas are correct, the methodology for doing analysis can be distorted by the arrangement and flow of linked spreadsheets, or the process itself can be wrong. This is often an inadvertent error, but one of the worst kind because it appears correct when it is not. For example, one department may feed assumptions into one set of variables while holding others constant while another does the opposite. The results will differ even though the model was correct.

Wasted Investment: Shadow IT applications drain ROI from investments in systems that are designed to perform the functions assumed by Shadow IT. This is most prevalent in reporting and analysis applications, such as data warehousing and BI, where projects are initiated with good intentions, but the propagation of BI in the organization never gains critical mass.

Inefficiencies: Shadow IT can be a barrier to innovation by blocking the establishment of more efficient work processes. The reluctance on the part of spreadsheet users to quiesce their applications to make way for an IT-led initiative is well-documented, and in the long run, it serves to postpone or avoid productivity gains.

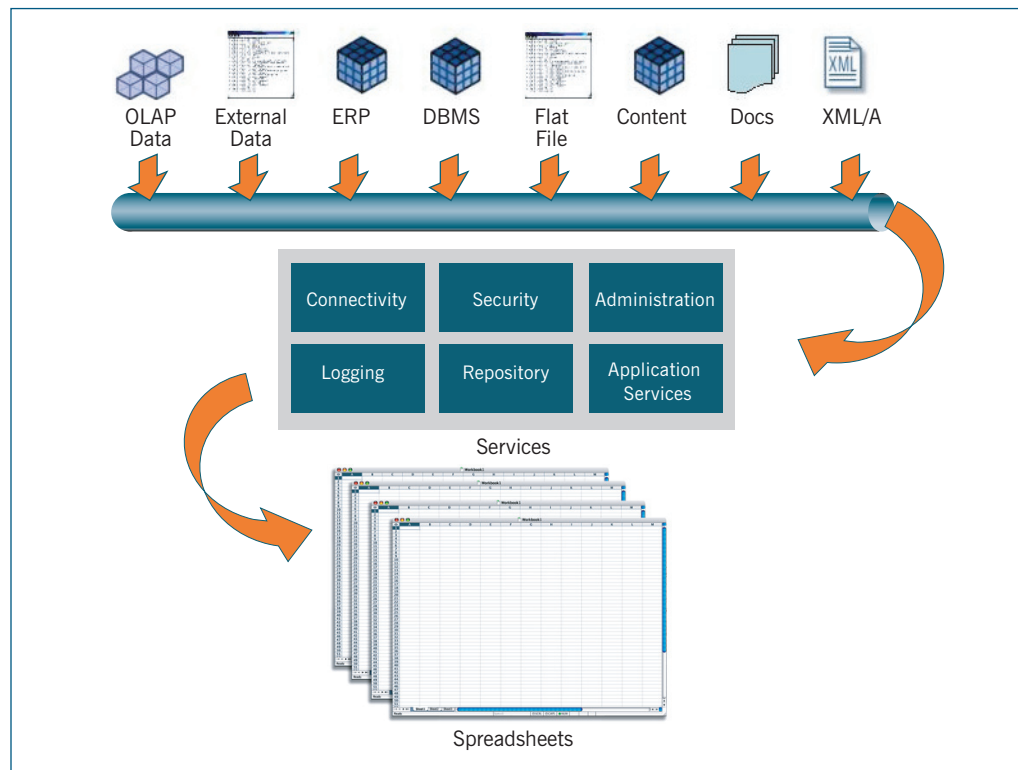
Barrier to Enhancement: Spreadsheet proliferation can act as a brake on the adoption of new technology. Because spreadsheets are deployed to fill critical needs, they must be replaced carefully. But lacking adequate documentation, controls and standards, that process is slow and error-prone. In many cases, initiatives with well-defined ROI are held up because downstream spreadsheet operations may be adversely affected. Only a painstaking manual process of sorting through the spreadsheets and cataloguing their flow and processing can insure that either their function can be assumed by the new technology or that their operation will not be affected.

SOLUTION

There is a solution to the problem, and it involves embracing spreadsheet use, but adding an element of control. Those aspects of spreadsheets that make them indispensable to both users and the organization can be applied to solving problems systematically and consistently. What is lacking is the application of a controlling environment that seamlessly provides all of the other services that spreadsheets require.

A promising solution is the application of rapidly emerging service-oriented architectures to provide capabilities such as persistent connections to data, security, logging and centralized management of query logic. In Figure 2 below, the middle tier depicts a series of services that can be invoked from a spreadsheet, eliminating the need for the spreadsheet developer to build these for each spreadsheet application. In addition, models, calculation rules, naming conventions, report templates and many other reusable objects can be managed at the server level, insuring adherence to standards while reducing the workload for both IT and business users.

Figure 2: Example of service-oriented architecture



Enhancing the Value of Spreadsheets

The benefits of spreadsheets are undeniable however, the drawbacks can be significant. Optimizing the situation requires retaining as many of the benefits as possible while reducing or eliminating the drawbacks. On the plus side, spreadsheets provide superior capabilities as a user interface, a distributed calculation engine, read/write data access, personal exploration and declarative model building. They are often applied in areas where they are weak, however, such as application development, collaboration, report engines, databases and information integration and transport.

By separating the useful qualities and replacing and upgrading the weak ones, a hybrid architecture can give rise to a whole new set of hybrid applications, such as the examples below:

Figure 3: Hybrid application matrix

	Spreadsheet Role	Services
Real-Time Alerting	UI, insertion of real-time data into local model, selection of further action	Event handler, rules engine, (light) data integration, pub/sub
Collaborative Planning	UI, incremental calculations, ad hoc modeling	Read/write access to source systems, security, version control, reporting services
Reporting from Internal and External Systems	UI, parameter selection, incremental results and statistics, charts	Federated data access, security, on-the-fly data integration, rules engine

CONCLUSION

Shadow IT is a barometer of the degree to which enterprise applications actually serve the needs of the organization. The broad and deep extent of Shadow IT clearly shows that enterprise computing does not fully meet all of the requirements. It is open to discussion whether or not these gaps should be addressed by more enterprise system functionality or by adapting the existing workarounds, which are devised primarily in spreadsheets and other personal productivity software. We firmly believe that logical partitioning of spreadsheet functionality, retaining its strengths on the desktop and migrating weaker services to a service-oriented architecture, is not only the best decision, but one that will succeed in the marketplace because of its perfect fit.

The rapidly evolving service-oriented architectures and standards, such as web services, SOAP, XML and even semantic modeling allow for new, creative solutions. It is now possible to separate the different aspects of an analytic application and locate the components anywhere in the network. To the extent that spreadsheets provide invaluable features, they can be preserved while their weaknesses, such as security, data integration, version control, reuse and collaboration can be relocated to centrally managed services.

Shadow IT is a hidden productivity killer. It creates the false illusion that IT expenses are low while penalizing the Shadow IT departments with higher budgets and masking the extent of IT work they are forced to do. Shadow IT is inherently inefficient and brittle and poses a real threat to an organization's agility. Until now, deploying services separately across a network was not feasible, but the technology and practices today make the decision not only practical, they make it imperative.