

**White
Paper**

The Managed Spreadsheet Environment:

How to preserve the best and most useful features of spreadsheets while mitigating the risks and expense caused by unbridled Shadow IT

ABOUT THE AUTHOR



Neil Raden is a BI pioneer and has designed and directed the implementation of dozens of analytical applications, Business Intelligence and data warehouse implementations across North America and Europe for over twenty years. He is the founder of Hired Brains, a consulting and implementation practice, and Hired Brains Research, providing market research, focus groups, collateral development, product marketing and strategic consulting services to the technology industry. Hired Brains is focused on using BI to improve organizations through a combination of architecture, execution and organizational change management. Neil is also an author, speaker and industry thought leader. His often-provocative articles (www.hiredbrains.com/knowout.html) appear in Intelligent Enterprise, DM Review, Business Intelligence Review, TDWI Flashpoint, Information Week, Teradata Magazine and many others. He is based in Santa Barbara, CA and can be reached at nraden@hiredbrains.com and he welcomes your thoughts and comments.

Neil wishes to thank Actuate for this opportunity to express his thoughts on this very important topic.

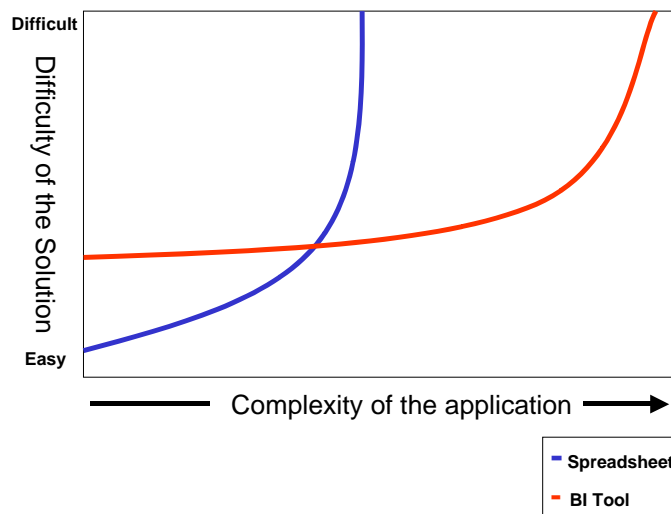
TABLE OF CONTENTS

- ABOUT THE AUTHOR 2**
- TABLE OF CONTENTS..... 3**
- EXECUTIVE SUMMARY 4**
- TWENTY-FIVE YEARS OF SPREADSHEETS 6**
 - Spreadsheets are expressive 7*
 - Spreadsheets are subversive 7*
- THE PROBLEM 8**
 - Shadow IT 8*
 - Other Spreadsheet Drawbacks 9*
 - Good Intentions, No Big Picture 10*
 - The Myth of Power Users 10*
 - The Silver Lining 11*
- THE SOLUTION 12**
 - Data Integration 13*
 - Abstraction of Data 14*
 - Standardization 14*
 - Security 14*
 - Integrity 15*
 - Scalability 15*
 - No Code 16*
 - Auditable/Repeatable 16*
 - TCO 16*
- Conclusion 17**

EXECUTIVE SUMMARY

The common assumption is that spreadsheets became widely deployed in organizations because they were intuitive and easy to use and filled a need that was not being met by other technologies. The truth is that spreadsheets exploded in popularity, not due to their original intended purpose, which was strictly for personal productivity, but because they were used as an alternative to enterprise-wide, IT-backed reporting, planning and analysis applications. Today, spreadsheets are used for complex multi-user application development, stretching to the breaking point their already razor-thin capacity for security, maintainability, scale, networking and concurrent use. Because of changes in the regulatory environment, such as Sarbanes-Oxley, these applications, no matter how well conceived and executed, can potentially expose organizations and their executives to fines and even criminal charges if they lack sufficient transparency.

As a prototyping tool, spreadsheets have no equal, but with constrained IT budgets the norm over the past few years, an attractive prototype can become a permanent stand-in. The drawback is that the feasibility curve for spreadsheets literally hits the wall as the complexity increases, as depicted in the figure below, and the applications either fail precipitously or gobble up greatly underestimated maintenance costs to keep them functioning. To make matters worse, these maintenance costs are largely the time of manager-level people, which not only increases costs in an obscured way, it diverts the attention of critical people from their primary roles. The principal contributors to complexity are scale (number of users, amount of data), serial models (linked spreadsheets, dependent applications), external data, such as customer lists or syndicated data, rapidly changing elements, assumptions and organizational structure. At the extreme, spreadsheet applications attempt to perform operations that are clearly best suited for enterprise tools, such as data cleansing and integration and often employ programming with scripting or Visual Basic, which places them squarely in the realm of software engineering.



Not all spreadsheet applications suffer from dangerous deficiencies in security, scalability, error trapping, etc. There are many instances of clever, even brilliant, workarounds that stretch spreadsheet software far beyond its intended purpose. The problem is that, in most cases, the workarounds are not brilliant and they generously substitute labor for good design.

Repetition, sluggish process, extensive setup, risk from error, maintenance burden and the diversion of the time and energy of highly paid people are the result.

Shadow IT, the direct and allocated costs of developing and maintaining IT applications and infrastructure beyond the control, and usually the visibility, of centralized IT, is a growing problem from a cost perspective, and IT needs to find a way to get on top of it in the interest of integration, standards, flexibility and agility. However, the use of spreadsheets is a crucial part of virtually every organization's information management process and they contribute a great deal of value. Banning them or trying to dictate their use has proven to be counterproductive.

How are spreadsheets used productively? What roles do they perform that are vital to an organization? Typically, spreadsheets are employed to fulfill roles in three primary activities:

1. Gathering, creating and managing data
2. Creating models and calculations
3. Disseminating information, principally as reports

Gathering, creating and managing data represent an entire discipline within IT, so the thought of domain experts with no formal training in data management devising applications is worrisome to most IT managers. Models and calculations, on the other hand, are more or less alien to mainstream IT, which is more focused on infrastructure and transaction processing. Nevertheless, a derived value shares equal weight with source data and modeling processes have to have a formal review and approval process, especially today. Only the last activity, disseminating information, is relatively risk free. The other two are the principal source of error and cost to organizations. A truly breakthrough solution would be one that preserves all of the features of current spreadsheets for personal use, but manages the use of spreadsheets for gathering data and creating models and calculations for shared applications.

There are alternatives to the Shadow IT conundrum. It is not necessary to throw away the baby with the bath water. All that is needed is to unbundle the best parts of spreadsheet programs from the practices that lead to dysfunction, excessive costs and risk in order to provide better managed solutions that suit everyone's goals. This paper proposes an alternative that preserves the best and most useful features of spreadsheets and mitigates the risk and expense caused by unbridled Shadow IT.

But first, having a better understanding of the evolution of spreadsheets from the perspective of work and how this software innovation changed the nature of work in organizations will be helpful to framing a solution.

TWENTY-FIVE YEARS OF SPREADSHEETS

Two Harvard Business School students invented the electronic spreadsheet concept in 1978, and they developed a product for the Apple Computer (and later the IBM PC) called VisiCalc¹. For some reason, they never patented the idea, and another Boston-based company, Lotus Development Corporation, converted the idea into a DOS-based (PC) product called Lotus 1, 2, 3. Lotus soared throughout the 80's until Microsoft challenged with a completely GUI-based spreadsheet for Windows called Excel². It took Excel a while to catch up to Lotus but with the help of Moore's Law and the company Moore founded, Intel, by the start of the 1990's, PC's had sufficient horsepower to run a GUI spreadsheet. It's been no contest ever since. There are over one hundred fifty million licensed copies of Excel in the US and around the world and by some estimates, twice that number of unlicensed copies - one half of a billion users of Excel. It is an anomaly to see any other spreadsheet tool used in most organizations. The term "Excel" and "spreadsheet" are virtually synonymous.

All of this begs the question, why was Lotus so successful in the 80's with their product? What was it about a spreadsheet product that made it the "killer app" for PC's, the tail that wagged the dog, the tool that drove the purchase of a computer where there was none before? The most commonly cited reason was that spreadsheets freed people from the dominion of IT. The computing landscape in organizations in those days consisted of a hodgepodge of proprietary mainframes and mini-computers, accessed only through programming languages or Byzantine request screens. A PC, spreadsheet and a printer turned an analyst into an instant report-writer. The delays, difficulties and costs of dealing with centralized IT resources acted as a powerful catalyst.

After some initial grumbling about it, IT saw merit in the idea, too, especially as it eased the burden on them for constant modifications to brittle, hand-made systems and endless requests for reports³. The only group threatened by this development was the mainframe reporting and decision support software vendors, who stood to lose the bulk of their customers (and ultimately did).

It wasn't long, though, before people grew weary of having both a PC and a mainframe terminal on their desk. Sensing a huge demand, a few companies developed hardware and software allowing the PC's to "emulate" a "dumb terminal," and PC-connectivity was born. The emulation software very quickly allowed for file transfer between the host and the PC, and the spreadsheets could now be fed data directly instead of needing a keypunch operator. Over time, with the rise of local-area networking, client/server computing, open network standards, such as TCP/IP and finally, today, data warehouses, data integration software and nearly unbridled bandwidth, the original spreadsheet concept has matured into a front-end modeling, reporting and application engine in a completely connected world.

In short, spreadsheets have become indispensable tools for getting the informational work done in organizations by empowering non-technical people to develop their own work environments, processes, standards and tools. People who rely on spreadsheets to get through the day, leverage two very important aspects of spreadsheet implementations:

¹ Dan Bricklin and Bob Frankston developed VisiCalc for the Apple and subsequently ported it to the IBM PC.

² In point of fact, Excel was originally developed for the Macintosh and later brought to the PC/Windows environment.

³ In fact, IT organizations now rely on spreadsheets to accomplish their tasks as much as any other group, in effect, a Shadow IT within IT

Spreadsheets are expressive

With little to no training, anyone can develop a model in a spreadsheet, format the results and print or publish it. It does not require a programming language, a scripting capability or knowledge about how the data is stored internally. There are no restrictions on commingling text and numbers, or adhering to rigid column definitions. There is no deceptively simple looking request language like SQL. A spreadsheet model can range from chaotic to tight, but it is entirely up to the author. There are no imposed standards.

Spreadsheets are subversive

Spreadsheets are subversive in the sense that they facilitate peer to peer and intra-departmental sharing through the back channel, the behind-the-scenes communication pathways. Spreadsheets provided the capability for non-technical people to use their computers to build analyses and to share them. This changed forever the way organizations worked. Now, a marketing analyst in Division A can craft a new pricing model and share it with a sales specialist in Division B without the need to move it through channels, a time-consuming and politically delicate operation. The back channel is where a great deal of work is accomplished today, in industry and in government. Spreadsheets reign in the back channel; their informality and individual ownership allow them to move freely without control or review.

THE PROBLEM

It's no mystery why there are 150 million licensed copies of Microsoft Excel on desktops and laptops around the world, and some estimate twice that many unregistered, nearly a half of a billion Excel users. Spreadsheets offer capabilities to create, modify, manipulate and report virtually any type of data without the need of supporting infrastructure. From their intuitive interface to their instantaneous response time, spreadsheets raise the level of expectations for all types of interactive software. Spreadsheets have expanded from the tool of choice for number crunchers to the de facto standard software tool for all "knowledge workers." In fact, spreadsheets are not just personal productivity software anymore; they are routinely deployed as critical, multi-user applications and relied upon as the repository and engine of some of the most important information in an organization.

The use of spreadsheets as data repositories, even in organizations with mature data warehouses, is derisively referred to as "spreadmarts" and "spreadsheet hell."⁴

Shadow IT

However, there is concern that a significant amount of the cost of developing and maintaining spreadsheet applications is out of line compared to competing alternatives. Spreadsheets comprise a major portion of a phenomenon known as Shadow IT, loosely defined as any of the direct and allocated costs of developing and maintaining IT applications and infrastructure beyond the control, and usually the visibility, of centralized IT. BoozAllen Hamilton first raised the issue of Shadow IT in 1993 in an article in *Technology and Business*. At that time, they estimated that in major commercial organizations, the allocated cost of business-area and departmental directed IT efforts, especially those involving PC's and PC networks, cost a staggering 140% of the entire enterprise IT budget. This had the effect of grossly overstating the efficiency of the IT organization (because their costs were being absorbed by the budgets of other groups) at the expense of every other cost center (who was shouldering the burden). This may distort measurement and metrics to a degree, but the implications are far greater than just the allocation of computing costs. Shadow IT suggests a clear failure on the part of IT to provide products and services that meet their clients' needs, and the problem is universal.

In the ensuing decade, Shadow IT has grown by 100% by some estimates, despite intensive efforts to provide Business Intelligence solutions centrally with data warehousing, metadata and reporting and analytical tools. Of the 150 million business users of Microsoft Excel worldwide, a large proportion are devoted to entering data by hand, extracting data manually from other systems, including data warehouses and even BI reports and performing as report servers. In fact, a great deal of the BI usage in organizations is driven by the need to populate spreadsheets, not replace them. The number one asked-for feature in BI tool evaluations is, "Can it export data to my spreadsheet?" This begs the question, has centrally managed BI been misguided, is there some essential characteristic about spreadsheets and personal databases that is too important to overlook, or are the efforts to convert people from disconnected personal tools a mistake? Enduring and expanding spreadsheet applications drain value from multi-year, multimillion-dollar efforts to develop a data warehouse and establish a standard BI environment. Or, is there a basic mismatch in the solution to the problem. Problem 1) IT is having a difficult time providing the data that the

⁴ "Spreadmart" is used by Wayne Eckerson of The Data Warehousing Institute in describing linked spreadsheets used to capture data persistently and operate like a Data Mart. "Spreadsheet Hell" is used commonly in the literature, but its origin is unknown to the author.

user wants; therefore, the dawn of ad-hoc tools. Problem 2) Excel doesn't have good data extraction capabilities, and in some cases has a limit to the amount of data it can hold. So, ad hoc web reporting is seen as the solution. The ensuing process change still keeps the manual data conversion process between the ad-hoc report results and the dump into Excel. In short, we never solve the problem.

Other Spreadsheet Drawbacks

And it is not just a cost issue. Research shows that as many as 90% of spreadsheets contain errors in data and logic and there are many cases of extremely costly mistakes being made. One Canadian company, TransAlta, was forced to take a \$24 million charge to earnings directly as a result of a spreadsheet cut and paste error that caused it to make an erroneous bid for the purchase of hedging contracts at a higher price than it wanted to pay⁵. Spreadsheet users engage in a combination of data acquisition/entry, model logic development and report formatting. Errors in report formatting are not often serious, but data and logic errors can be very serious and very costly. This is especially risky because spreadsheets commingle data and logic in a way that makes it very difficult to spot errors. Contrast this with an application that employs a database and a separate set of tools for building and executing rules. The abstraction between data and logic is absent in spreadsheets and is a primary source of errors.

Besides being susceptible to undetected errors, standalone spreadsheets are not very secure. Spreadsheets lack logs that capture and store changes. It is routine to send copies of spreadsheets by email (or even mail) and changes made in the process are not easily detected. Personal computers and laptops are frequently left unattended or stolen, allowing sensitive information to be leaked.

Usage modes of spreadsheets are not evenly distributed, either. The most advanced features of spreadsheets, such as pivot tables, require set up by the most skilled users. This creates a sort of hierarchy where many people depend on the skills of one person to develop these advanced features. The result is a faux IT environment where the work is divided between technical people and everyone else. This has HR implications, as well: if the person leaves, how can they be replaced? On the other hand, being the expert can act like an anchor, holding someone in a position long after it is time to move them on.

Today's regulatory environment demands more transparent and controlled processes, and spreadsheets are a weak link in the chain of accountability. People without solid training in software engineering and architecture, data integration and data quality, and overall project management are likely to make costly errors of design or omission. The risks are not limited to excessive cost, but potentially expose companies to loss of customers, disastrous decisions, fines and even potentially a loss of license to operate.

The problem is not limited to spreadsheets. Personal databases play a role too, but spreadsheets are the greatest part of the problem by a wide margin. When personal software is stretched beyond its intended purpose and deployed as a shared application, problems arise that were not anticipated in the design of the software. On the other hand, shared or enterprise software solutions rarely have the fit that a custom-made spreadsheet does. The user interfaces are rarely as intuitive and familiar and the care, custody and control of a centralized IT department can often be interpreted as arrogance and detachment. But only the centralized IT processes can deliver the security, scalability, reliability and consistency that are needed. It is nearly impossible to enforce these attributes on to Shadow IT applications.

⁵ Toronto Globe & Mail, June 4, 2003: <http://www.bpm.ca/TransAlta.htm>

Good Intentions, No Big Picture

When someone creates a spreadsheet for their personal use, or to share with others, they employ both subject matter expertise and elements of design. Whether they actually possess the needed subject matter knowledge to do the job effectively is highly variable. Possessing an understanding of good design for reuse, maintainability, security, endurance and a host of other desirable qualities in systems design, however, is very unlikely. There are those in “Shadow IT” organizations who are, in fact, quite skilled at building durable systems from personal tools, but these exceptional people are the exception to the rule. Eliel Saarinen, a groundbreaking architect of the early twentieth century, and also the father of Eero Saarinen, the designer of the St. Louis Arch, among other things, had this to say about good design:

Always design a thing by considering it in its next larger context – a chair in a room, a room in a house, a house in an environment, an environment in a city plan.

Saarinen never saw a spreadsheet, his domain was bricks and mortar, but he hit the nail on the head, so to speak. A perfectly functioning spreadsheet on a personal level exhibits undesirable tendencies, emergent failings, when it migrates across a few dimensions, notably time and activities. To paraphrase Saarinen, one might construct a spreadsheet to exist at the moment it is needed, but the “larger context” it exists in is time, and as things change in an organization, as they inevitably do, the rigid arrangement of things in a spreadsheet, a real advantage for capturing diverse relationships at a point in time, becomes a barrier that can only be crossed with additional development and/or remediation effort.

The problem is magnified by the proliferation of dependent spreadsheets that are modified and calculated in serial form. Lacking a professional system development methodology or even version control scheme, the whole process is manual and the quality of the effort is wholly dependent on the participants’ memory of which sheets are related and where they are, and which are the most current versions. It is possible to implement controls to avoid some or all of this confusion, but it is not realistic to expect accountants, actuaries, salesmen, marketing, executive or route salesmen to be able to both see the need for and have the skills and temperament to pull it off.

The Myth of Power Users

The term “power user” typically evokes images of individuals, or “go-to guys,” whose skills and proficiency with certain technologies are demonstrably better than those around them. Unfortunately, the power user is more myth than reality. In practice, there are many power users, of different types. There are different kinds of skills needed to develop applications, and the power user concept fails to take into account most of them. To understand the functionality of the software and how its various features operate and interoperate, to learn the “tricks” to make the software perform, and to keep current on enhancements requires experience and an intellectual curiosity that only a few people possess. But, spreadsheet applications that are shared and deployed, not just built for personal use, inherently require access to data. There is a pressing need to understand where the organization’s data is and what it means. The people in the operating or staff units who understand the information flows and semantics of the various source systems (source in the sense that they feed data to the spreadsheets) are rarely referred to as power users, but they certainly provide an indispensable component to the application building process.

There are also those who may not have the temperament to learn the intricacies of a software tool, how to write a macro or employ add-ins, but who know how the business operates. Writing a function is a useful technical skill, but knowing how the function should perform is just useful. From simple questions like, “How do we calculate operating cash flow,” to political/procedural ones such as, “What is the best way to align the budget process to our

data, manipulate and analyze data, but they also represent a statement about approach, goals, objectives and image and their content and presentation has to be thought through carefully.

Technical power users, information power users and domain experts all combine to build useful applications. In fact, it is a little hard to imagine how a “knowledge worker” could be useful to an organization without having skills in at least one of these three categories. In other words, everyone is a power user.

Why does it matter who is a power user or if everyone is? It matters a great deal, because when management, particularly IT management, attempts to “do something” about the Shadow IT problem, they typically target the “power user” of the first type and overlook the needs and skills of the other two. This is one of the problems with implementing Business Intelligence solutions – it is assumed that if you can entice the “power user” to give up his or her spreadsheet in favor of a BI solution, that the rest of the group they serve will follow suit, of necessity. If the “go-to guy” has gone, then they have to go, too. In practice, this strategy has not worked. Shadow IT costs have actually doubled from 2000 to 2003 from approximately 10% of IT costs to 20%. In organizations where IT spending is severely restricted, Shadow IT costs have risen to close to 100% of IT costs.

The Silver Lining

People who use spreadsheets are not irrational (generally) and are not intent on wasting the organization’s resources or making IT look bad. They are just trying to do their work. Spreadsheets work for them. Unless one has a background in software engineering, data management or systems architecture, the logic behind using a spreadsheet in the current mode is perfectly reasonable. The answer to the spreadsheet problem, and it is a problem, is to take a page from Dr. Strangelove’s playbook – learn to love them. The answer is not to ban spreadsheets, but to embrace them.

THE SOLUTION

With or without an official name like Shadow IT and research to back up its extent, many organizations recognized over time that spreadsheet proliferation was causing problems. Either acting alone or with the help of consultants and software vendors, quite a few solutions have been tried. The solutions more or less fall into one of three categories; the three primary approaches are:

1. Use an Excel front-end to applications, such as a plug-in, particularly with BI applications. The assumption is that the familiar interface will encourage people to abandon their spreadsheet processes in favor of new ones, fronted by the familiar interface.
2. Use a server-based generator of spreadsheet-like applications, with limited functionality, as a complete replacement for spreadsheets, allowing complete control of the usage and disposition of the product and the user experience.
3. An environment that centrally manages data availability and quality, security, version control and quality assurance of data, models and calculations, and generates actual Excel files for reporting, manipulation and any other function a user desires.

While the *first two solutions* can clearly eliminate spreadsheet risk and cost described above, neither has proven to be satisfactory with the user population. Both solutions assume that the spreadsheet user interface is the key to acceptance. Research has shown, however, that replicating the user interface is not sufficient. If the underlying (and usually hidden) conceptual model of an application is not obvious to the person using the application, the user interface is irrelevant. For example, if it were possible to port the control panel of a nuclear power plant to a spreadsheet, someone with no training, even if he/she were an Excel “power-user,” could not step up to the plant operator function. On the other hand, useful applications delivered in fully functional spreadsheets, that people understand, are a great improvement, which is the aim of Solution #3.

Solutions #1 and #2 are limited by deficiencies:

- Not leveraging skills of people used to working with spreadsheets. Excel as a front-end for a server-based system reduces the power of Excel to just a user interface and deprives skilled people of performing their own analysis as they see fit.
- Lack of speed: these solutions have proven to be arduous processes and as part of larger IT initiatives, they lack the fast turnaround time that people expect with spreadsheets.
- They are difficult to deploy, requiring multiple iterations of requirements, design and build. And, in the face of IT’s desire to reduce the number of specialized applications running on desktops, these systems add maintenance complexity.
- Spreadsheet applications are not necessarily durable. They are often short-term, temporary models for current issues. Investing time in server-based development, data modeling and metadata is not only too time consuming, but may not be needed.
- Tools designed to reach a wide audience with a general case, one-size-fits-all model are difficult and costly to maintain.
- Inability to support disconnected users, such as sales forces, field service personnel, frequent travelers, etc. One of the most desirable features of spreadsheets is the ability to use 100% of their features in standalone mode, when not connected to the network.

Solution #3 on the other hand, provides a number of very desirable features while preserving the use of the spreadsheet:

- Data integration/data quality
- Abstraction of data from the spreadsheet environment
- Standardization
- Security
- Integrity
- Scalability
- No code
- Auditable/repeatable Cell locking
- TCO (Total Cost of Ownership)
- Write back systems to improve data quality and support decision making processes

Data Integration

A major component of the cost of Shadow IT is the direct cost of reconciling conflicting, incorrect and out-of-date data. To the extent that data exists in original sources, orderly and accurate process is needed to direct and repurpose that data for other uses. This obviously requires some sort of metadata that exposes information as a business model and an engine that can move the data from the source to spreadsheet. Data warehouses, data marts and EII (Enterprise Information Integration)⁶ are just three common methods, and are routinely used now, but the connections are still largely point-to-point. A managed spreadsheet environment would simplify this process by mapping the most commonly used data through a metadata layer that could be incorporated, in whole or in part, to generate new spreadsheets as they are being built. Unlike a plug-in based solution, however, access through a metadata layer should not be the exclusive means by which data is available. A managed spreadsheet environment should be able to access multiple data sources directly and combine this data within calculations on the spreadsheet.

There should be some very basic rules for a managed spreadsheet environment. For example, no data should be created in a spreadsheet unless it is derived by a formula, or entered manually, with the following exceptions:

- No other application exists to capture the data. An example would be daily activity reporting of field people.
- Planning, budgeting and forecasting applications, where the synthesis of data as a result of calculations and models is reasonable.
- Operating a spreadsheet by entering assumptions, answers to prompts, etc.
- Other purely analytical work.

No data should be stored in a spreadsheet, except for optimization purposes or temporarily until it can be transferred to a more secure environment. A spreadsheet is not a database and lacks most of the features that make databases secure and reliable. If vital information is created or captured in the spreadsheet, then features should be available to deliver or write this new data back to permanent, central storage.

⁶ EII is commonly defined as the optimization and federation of queries and potentially caching as well (for frequently requested data)

Abstraction of Data

A facility that creates spreadsheet files may or may not populate the spreadsheet with data. In those cases where the spreadsheet operation calls for retrieving data from external sources, the process should be controlled by a facility that “abstracts” the data from the model. For example, suppose the analysis calls for all of the rejected credit card authorizations from a website in the last 24 hours, matched to master lists of merchants and products. Specifying this request should be no more complicated than the previous sentence and the managed spreadsheet environment should be able to resolve this request without any additional effort. This requires the features of an EII toolset, clearly features that are not available in a personal productivity product like a spreadsheet.

Providing sourced data to generated spreadsheets would have limited use, though, if it was not accompanied by dynamic access to enterprise reference data, such as hierarchies and master lists, so the reports can drill up and down according to the current arrangements.

Standardization

In the same way that a managed spreadsheet environment would build proto-spreadsheets for the standardization of data integration, presentation format and data quality management, standard calculations and models can be developed in the same vein. These standard spreadsheet designs are subject to much more rigorous testing, which serves to lower the risk of incorrect calculations getting into circulation.

It is far more effective to have people build models from snippets of standard models than to develop them from scratch and risk error. Also, as approved methods, models and calculations change over time, the need to find and modify dozens or even thousands of spreadsheets can be eliminated.

Building models with standard components is not a hindrance to independent thinking. On the contrary, it frees the modeler from tedious, repetitive work and allows him/her to concentrate on more important aspects the job. Using a standard set of models does not preclude anyone from being original, but it does insure that non-standard calculations and assumptions are obvious to everyone, while still encouraging evolution and refinement of the current models.

Security

1. Being assured that data, models and reports, once completed, vetted and saved, stay that way and can not be altered or distributed without the proper approval.
2. Being able to work independently, access the data. Models and reports needed and not have the work lost or altered, or be hindered, limited or preventing from being productive by arbitrary security schemes.

Because a managed spreadsheet environment operates from a central point, it is possible to create a single spreadsheet design, with one set of data connections, calculations and reports, and distribute the spreadsheet to hundreds or even thousands of users simultaneously, with only those parts of a spreadsheet workbook that each individual person is interested in and authorized to see. This not only increases the level of security and transparency, it widens the audience considerably. This sort of scale is impossible when customizing spreadsheets individually.

Spreadsheet security can come in the following flavors:

1. Personalized data delivered to the same template.
2. Personalized files delivered to the right people (distribution security, who, which file, when).
3. Point in time (standardization or integrity) so that everyone gets the same snapshot.
4. Fraud prevention through cell and worksheet locking. Preventing manipulation of cells or worksheets that may introduce fraud.
5. Transaction security, securing the round-trip through *http*'s, user credentials and centralized update rights.
6. Usage security, knowing who and when spreadsheets have been accessed if the email only provides a *url* to the central server.

Integrity

SOX (Sarbanes-Oxley) 302 requires that organizations attest to the fitness of their reporting processes 90 days prior to filing. SOX 404 requires that they disclose and provide proof of internal controls. A managed spreadsheet environment should be able to produce spreadsheets with locked fields. Defining what elements of a spreadsheet are updateable and which are not is a key feature in raising the level of confidence and compliance in spreadsheet-produced reports. Standard calculations should not be overwritten. Data sourced from other systems should not be modifiable. In some cases, this may be necessary, such as restatements or scenario-based analysis, but any field overwritten must be preserved in an audit trail and changes should be visibly highlighted. Most of all, eliminating the out-of-synch arguments that are often encountered from conflicting spreadsheets can be handled with point-in-time distribution to enforce a common view.

Scalability

Once a standard snapshot instance is established, the expected distribution of these spreadsheets to thousands of recipients simultaneously becomes matter of fact. A managed spreadsheet environment must meet this expectation. It is not uncommon for spreadsheet applications to collect and aggregate these hundreds, and in some cases, thousands of spreadsheets. Though there are good reasons collecting data this way, the process of aggregating the individual submissions using desktop software is not practical. In the same manner that a managed spreadsheet environment should broker data requests to various sources and provide reliable access and distribution of individual spreadsheets, it should also provide the mechanism to consolidate and store spreadsheet data and be able to pass this data back to a centralized location for permanent storage.

No Code

Development of the spreadsheet at the server level should include powerful features that are invoked without programming, ideally using a spreadsheet paradigm. These features should go beyond the supplied features of the spreadsheet program and allow the developer to easily expand the power of the spreadsheet without the need to write code—Visual Basic, for example. This requirement should not be exclusive, however. Managed spreadsheets should provide a coding capability as an escape hatch available for developers in order to make the impossible possible. Tasks such as the write-back capabilities mentioned above could be considered one of these impossible tasks.

Auditable/Repeatable

The origin of each spreadsheet can be traced, calculations and sourced data can be locked, so that it possible to meet the requirements of SOX 404, to deliver proof of process and SOX 302 which delivers CFO attestation to the validity of the numbers. Other features to support audit ability and repeatability are formula evaluation and validation, data integrity checking and security rights. The system should not only be able to provide these features, but also template design version control, scheduling, and automatic distribution rather than hand delivery, such as traversing a distribution hierarchy to include a new recipient for example.

Spreadsheets can be enhanced to preserve changes made, either inadvertently or deliberately, so that an audit trail can be constructed. This is not a feature of a standalone spreadsheet, nor should it be, as a personal tool.

TCO

Controlling Shadow IT costs is a multi-level effort, but saving the labor costs of professionals operating as programmers, database administrators and systems architects can go a long way toward solving the problem. A fair financial analysis of a managed spreadsheet environment has to go beyond a ROI calculation, because it would involve some upfront software costs that are already amortized with spreadsheets. More importantly, current modes of spreadsheet use are so thoroughly woven into the fabric of everyday work, and have been in place for so long, that working with spreadsheets, especially the repetitive, tedious, costly parts seems like the actual work. In other words, the time spent building spreadsheets from scratch, storing and finding them when needed, modifying them manually as needed, building redundant data extraction routines and a multitude of other tasks that could be eliminated, are not even viewed as counter-productive in most cases.

Conclusion

Correcting an inefficient and potentially risky spreadsheet environment requires examination of work practices with a realistic view about why the problem exists. Careful study will reveal the benefits that are possible. On the other hand, formulaic answers lead to formulaic solutions, and Shadow IT continues despite many efforts and initiatives to get it under control. A managed spreadsheet environment holds the promise of redirecting the nature of what we loosely call analytical work, but transforming work habits takes time. A measure of realism is a good idea, as people do not adopt a different mode of operating overnight. The alternatives, though, are not acceptable.

Schemes to limit or eliminate the use of spreadsheets in organizations by replacement with Business Intelligence software are ill conceived and bound to fail. More often than not, BI tools act as data sources for spreadsheets applications because the BI industry has focused more on read-only OLAP and reporting than on modeling. Spreadsheets offer a wealth of convenient and accessible functions to people of widely varied skills, backgrounds and experience. As personal productivity tools, spreadsheets have no equal. Application development with spreadsheets for multi-user purposes or as data or metadata repositories is, on the other hand, a major component of the serious and growing problem of Shadow IT.

A managed spreadsheet environment effectively retains all of the benefits of using spreadsheets are personal software, but eliminates substantial amounts of risk and cost. Implementing this environment will be challenging, though, as there are only a few technology providers able to provide the capability. An even greater challenge will be convincing organizations that Shadow IT is not only a problem, but a problem that can be solved. And the greatest problem of all will be convincing those who rely on spreadsheets to do their jobs that the managed spreadsheet environment not only will not make their job more difficult, but that it is a good idea. Changing work habits is far more difficult than implementing technology and a successful migration will require ample amounts of skill in change management. Data warehousing and BI was often presented as an alternative to spreadsheets by IT and users are naturally suspicious of the "next new thing."

The demand for transparency in reporting, like Sarbanes-Oxley, will make managed spreadsheets palatable to management, but ultimately, it will be acceptance by the rank and file that will make the difference.