# A New Platform for Building and Managing Business Owned Applications (BOAs) for Insurance Organizations

# SpreadsheetWEB

*Whitepaper*

May 2009

**Pagos, Inc.**
**1031 Farmington Avenue**
**Farmington, CT 06032**
**www.pagos.com**

# Table of Contents

# Introduction

Business-Owned Applications (BOAs) is the general term for all applications used by the business units that are developed and managed completely outside the control of IT departments.

Why do BOAs exist? IT departments have neither the budget nor the time to develop and maintain applications for all business needs. With limited resources, they have to focus on bigger projects first and prioritize application requests coming from business units accordingly. When their needs cannot be met by IT departments in a timely manner, business units have no choice, but to create their own applications.

BOAs can vary from Excel based applications to complex externally hosted web applications. Most medium to large businesses typically have hundreds, if not thousands, of BOAs dispersed throughout the organization mostly on desktop computers.

Despite their widespread use and tremendous benefits to business units, BOAs have significant shortcomings. These applications are designed for single user desktop environments and cannot be deployed in a multi user web environment. Also, they cannot be used in a collaborative fashion in which multiple users can access and work on a particular spreadsheet model or process. While having limited integration with scalable database platforms, they lack security and version control features most enterprise environments require.

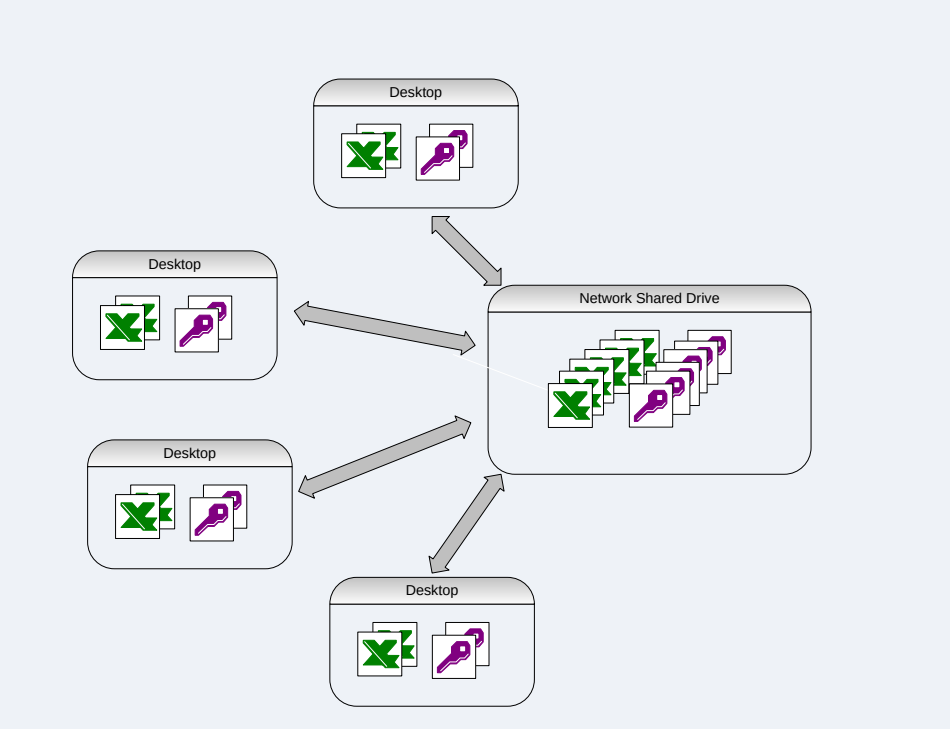
This whitepaper describes a new platform, called SpreadsheetWEB, for building and managing BOAs. SpreadsheetWEB software is designed for business users to effectively transform any business process built around spreadsheets and convert those legacy proprietary processes into ~~a~~ web based solutions without any programming or IT involvement. SpreadsheetWEB empowers business units to deliver without having to depend on scarce IT resources, while enabling IT departments to devote their valuable resources on more business critical needs and still ensure that BOAs are developed on an approved platform.

## What is a Business Owned Application (BOA)?

A BOA can be defined as any application that is developed and maintained outside the rules and regulations of an IT department. BOAs are fully funded by business units. In most cases, BOAs are developed and maintained by business units themselves by forming internal IT teams. Typically members of those teams are business people trained in developing Excel, Access, and VBA/Macro based applications. Their training is often informal such as learning from books, online sources or colleagues. If a more complex system is needed, business units rely on outside vendors to develop and maintain externally hosted web applications. Those BOAs are still operated independent from internal IT.

Most common platform for building a BOA is Microsoft Excel. Sometimes Microsoft Access has also been used in conjunction with Excel mainly as data storage and reporting facility. However, due to database knowledge requirements, Access usage has been limited in building BOAs. Excel is mostly used to handle business logic and the analysis portion of a BOA. If the data is not too large, it is typically embedded directly in the Excel file. Otherwise, it can be connected to a flat data file, i.e. txt, csv, or an Access database via macros. Excel macros are frequently used to build user interfaces and other operations that cannot be handled with built-in features.

Figure A below shows a typical BOA. It incorporates number of Excel and Access files. The application is copied on number of user desktops. Typically there is a network folder where users can download latest files and data. However, this does not imply that there is any integration. Each user still has their own local data and it cannot be shared or synchronized with other users. There is not a centralized data backup for the application other than general purpose file based back-up.



If a BOA is developed for reporting or analysis purposes, it relies on data generated from an enterprise databases system managed by IT. Typically, business units employ their internal staff familiar with those databases to generate queries to export large datasets. The data snapshots are often captured periodically. Often business units use a procedure to retrieve the data in a flat file format, process, and convert it into a format that can be used by the BOA. Typically, data fields and definitions are changed during that transformation making it difficult to track the original data source. As a result, different versions of the same data exist throughout the organization. Naturally, there will be inconsistencies from the original data, due to this process. Consequently, the Reporting of different results based on the same data becomes a significant pain point to the organization as their squander human resources in effort to match or recapture the original data.

## Issues with Traditional Approaches for Developing BOAs

While the BOAs discussed prove to solve a particular business problem and benefit business unit tremendously, they create a set of other problems. Mainly, those problems are caused by the limitations of Excel and the changing expectations from BOAs in the current business environment that requires superior integration and web-enablement.

Issues and limitations with traditional BOAs can be grouped in four categories:

**Security Related Issues**: Often, BOAs incorporate either sensitive data or business logic that should be protected. While Microsoft Excel provides several security measures, including password protection for workbooks/worksheets, none prevent viable protection. Distributing those files, even with password protection, doesn't always provide the level of security that most organizations need.

J. Walk & Associates [(1)] provides a detailed treatment of security measures in Microsoft Excel and its limitations. The page also includes a macro to derive password to unprotect a password protected worksheet.

**Distribution and Version Control Issue:** Most BOAs are desktop applications built using Excel, VBA, Access or combination of all. They are typically distributed by the duplication of files onto each user's desktop. As multiple people begin working on the same file, version control becomes a major challenge. Managing multiple spreadsheet versions becomes a time-consuming and costly task for any company. Accessing spreadsheets on shared network folders is a popular way to reduce the version control problem. However, this allows only one person to work on the same file at a time. Hence, this approach fails with frequently accessed spreadsheets.

**Lack of Collaboration, Workflow and Integration with other systems:** Because they are often designed for desktop usage with local data storage, BOAs cannot be used in a collaborative way. With hundreds of BOAs dispersed throughout the organization and developed and maintained by different units, it is clear that integration is necessary between these systems. Often, multiple BOAs either use the same data or business algorithms. Due to the limitations of the underlying technology, it becomes an eventuality for business units to develop duplicate algorithms and using duplicate data. Even more confusing is when reports produced by different BOAs are compared and results fail to match. End users will claim their application is correct and finger pointing begins. Management becomes frustrated as to which numbers to believe. Ultimately, the cost to the organization is the needless disposal of effort and resources to figure this out.

**Data Related Issues:** One of the major issues associated with BOAs is creation of data sources throughout the organizations on individual desktops that cannot be controlled and managed centrally. Most BOAs do not adhere to data standards and models commonly used within the IT organization. Because the data is copied to local desktops as standalone applications, there is no control over quality and versioning. Also, since the databases are not maintained by IT, they lack proper backup mechanism. This often leads to loss or overwriting of critical data. Business rules that reside in Excel applications can often be connected to incorrect version of data residing in Access databases resulting from the absence of version control. Unavoidably, this leads to erroneous results that could sometimes be done unnoticed.

## New Platform for Building and Managing BOAs - SpreadsheetWEB

Why do businesses continue developing BOAs despite all issues described above? The answer is simple. Without them, their operations could come to a complete stop. BOAs are inevitable and they are here to stay. The correct approach to deal with BOAs is not to shut them down but to manage and control them. The key to the success of this approach is to supply business units with a platform that they are familiar with to continue building and operating BOAs quickly and independently.

SpreadsheetWEB is a revolutionary alternative to developing BOAs. Based on the idea that business users are familiar with Microsoft Excel platform in developing business owned applications, SpreadsheetWEB offers a cost effective and flexible solution. Once those applications can be built in Excel, SpreadsheetWEB convert them into fully functional, database driven web applications or web enabled processes without requiring IT resources. Empowering business units to build and maintain business-owned web applications, SpreadsheetWEB reduces the long and expensive process of building web applications to a shorter, cost effective and business-owned and controlled one.

## Advantages of SpreadsheetWEB as a BOA Platform

One of the major advantages of SpreadsheetWEB platform is that it brings BOAs under a single platform. In doing so, it enables business units with the flexibility to build and manage their applications in familiar Excel platform. Simultaneously, giving IT departments piece of mind since SpreadsheetWEB is based on enterprise level security and data standards. Built on Microsoft .NET framework and Microsoft SQL server, it allows IT departments control the security and data models, and provide enterprise level disaster recovery.

SpreadsheetWEB platform also provides collaboration and workflow capabilities. Data and processes can be shared across business units and departments.

## How SpreadsheetWEB Works?

With SpreadsheetWEB, application development and maintenance is divided into three stages as shown in the Figure below. First stage is the design. It is performed only once and doesn't have to be done until either the business logic or user interface is needed to be updated. Second phase is the deployment where the application is setup in terms of security and accessibility. Third phase is the on going management, maintenance, and workflow.

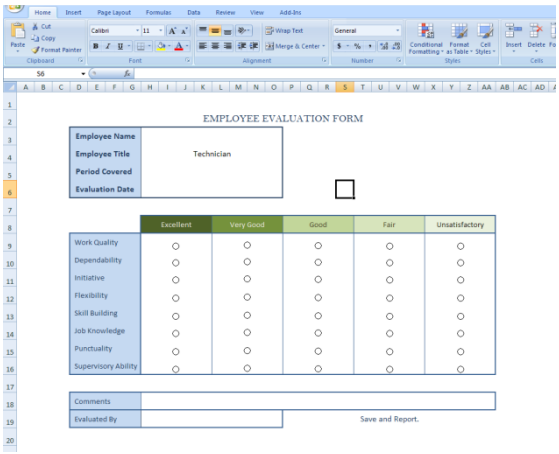
## Design

This step involves processing the spreadsheet file with a design tool called SpreadsheetWEB Wizard. This step is performed by the business user who is familiar with the spreadsheet model or process. In most cases it takes a matter of minutes, not hours, to process the file. SpreadsheetWEB Wizard then creates a custom code, in XML format, and embeds it in a hidden worksheet in the file. Once the code is embedded in the file, the business user can rerun the process without having to re-enter those selections.

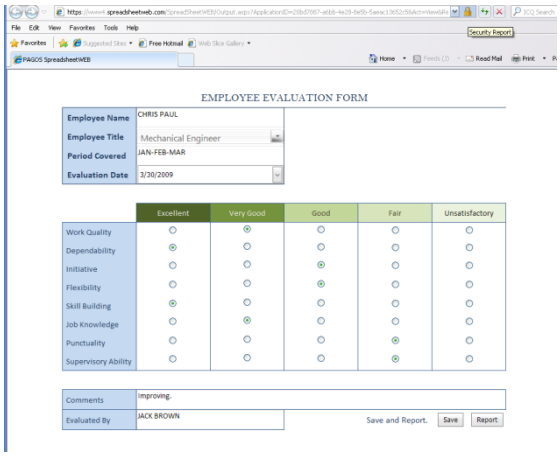
In design process, the business user has to make following choices:

- Number of pages in the application.
- Navigation method. Tab based navigation and wizard based navigation are the options
- Display areas on each page. There is typically unnecessary information like formulas, data, on each worksheet that the end users have no need to access. The business user will needs to identify the range of cells on each page that the web application should display
- Input controls on each page. Editable cells can be assigned one of the many controls supported: textbox, combo box, checkbox, radio button, list box, calendar, scrollbar, etc.
- Buttons. Business user can insert button controls on each page including Calculate, Reset, Next, Back, Save, Email, and Export.

Figure below captures how an Excel based "Employee Evaluation Form" on the left can be turned into a web based form with SpreadsheetWEB on the right. Note that web form includes browser base input controls like textbox, combo box, calendar, and radio buttons as well as buttons to save the entries in a database.




Excel




SpreadsheetWEB

## Deployment

The next step is to deploy the spreadsheet file on the server. SpreadsheetWEB Control Panel is essentially a web interface to deploy spreadsheets on the server. The user will then upload the spreadsheet file to the server triggering the SpreadsheetWEB server software to load the spreadsheet file, extract the custom code from the hidden sheet, and create a web

application. The web application has all the functionality of the spreadsheet file but delivered over the web. Note that the application will not include Microsoft Excel specific features like grid, menu, toolbars, etc. The application has now been converted to look and work as a professionally developed web application.

SpreadsheetWEB also creates a database driven web application that requires end–user data to be saved such as web forms. For those applications, SpreadsheetWEB automatically creates database tables with respect to data fields to be stored.

Depending on an organization's size and internal processes, either IT departments or business units are responsible for the deployment process. In smaller organizations or with smaller size applications i.e. data collection, web, forms, surveys etc., business units can be charged with the responsibility of deploying applications. Conversely in mid-size or larger organizations, or if the web applications is more complex, the deployment responsibility is given to IT departments as they are more familiar with the processes and risks involved in testing, deploying and managing web applications.

## Management

SpreadsheetWEB Control Panel is created to provide a web interface to manage SpreadsheetWEB applications, users, and database.

### Application Management

SpreadsheetWEB Control Panel allows application owners to control access to their applications. Applications can be setup as password protected applications. Those applications can also be setup with self registration where users would be prompted with a login page that includes a signup feature.

SpreadsheetWEB also supports encryption of data transmitted between the end user browser and the server via SSL. SSL enabled applications are recommended where sensitive data such as personal information is being transmitted.

It is also possible to make an application available during a period of time. At the end of this period, the application link will no longer be available for access

Database Enabled Applications is another important feature supported by SpreadsheetWEB. Here an application can be created with a Save button which stores the user data in a database table. Two types of database applications are supported: Single Record and Multiple Record applications. By default every database applications is "multi-record" meaning that each time a user presses the Save button, data is saved as a new record. This is a typical application type when building online forms, questionnaires, surveys etc. However, there are situations when a single record application is needed. For example, a monthly time sheet where users are expected to login and put their initials into preferred times of the month. In this scenario, there is only one data record that is modified by all users.

Another feature of database enabled applications is concurrent editing. By default each data record can be edited and if there is concurrent access to a particular record, the last save prevails. However, with SpreadsheetWEB's flexibility, it is possible to disable concurrent editing, and prevent a user from accessing a particular data record while it is being edited by another user.

Additionally, each web application can be assigned to a Group. For example, a company can define groups like Finance, Marketing, Sales, etc. Each application must be associated with a group. Applications being used by the Sales team can be put under "Sales" folder. Subfolders may also be created. Access to these specific applications can be controlled by associating Users with Groups as will be explained in the next section.

**User Management**

SpreadsheetWEB Control Panel includes a User Management Module that is accessible by only Admin level users. Admin can create users and associate them with access to specific applications via Group concept as explained above. For example, a particular user can be associated with Marketing and Sales groups. This user can access all applications under those two groups.

Each user is also given a user role. Depending on the role, the user's access to applications and data can be controlled by the Admin. There are 4 major user roles:

- Admin. This type of user has full access to all features and functionalities.
- Standard User. Standard user has full access to applications and data within own group. Standard User can also see data submitted by other users within the same group
- Limited User. This user is limited to viewing and editing own data only within a group. For example, in a CRM application built with SpreadsheetWEB, sales people would be created as Limited User hence can view/edit own customer data. While Sales Managers would de created as Standard User and see customer data for all Sales People within their group.
- Guest User. This is the most restrictive user type. This type of user has no access to Control Panel. Guest User can only access applications within their group provided that they know the application URLs. Guest user can also access own data. Since they cannot access Control Panel, they can only access their latest data. An example of this type of Guest User function might be an event registration system where staff members would be required to enter data for multiple seating. Once registered, a Guest User can visit the applications link at different times and always see own data.

**Data Management and Workflow**

With SpreadsheetWEB, it is possible to create data collection applications. SpreadsheetWEB software is fully integrated with Microsoft SQL Database. It automatically creates a database table for each application that includes a Save button. The process of creating and managing database tables is completely transparent to business user.

"Employee Evaluation Form" described in the previous section is a good example. Each instance a user accesses this application to enter data and presses the "Submit" button, that data is stored in the database. These records can also be viewed through the Control Panel. Control Panel users can acquire view or edit access to these records depending on their access credentials.

SpreadsheetWEB also includes features to build a workflow around the data. It is possible to define a specific set of statuses for each application independently. For example, a status list of "New", "Pending", "Sold", "Lost" can be assigned to a Sales Quoting tool. It is also possible to transfer the ownership of a record to another user in the system. One can also control the Status and Transfer features with worksheet formulas. Consider the example of

the Sales Quoting tool and assume that it will include a Lead Generation page to collect prospect information. A simple worksheet formula can be inserted in the Excel file to match a prospect to a sales person with respect to zip code. Hence, the incoming leads would be forwarded automatically to appropriate regional sales people.

SpreadsheetWEB also includes email integration. Information collected in a web application can be forwarded to a list of email addresses automatically. Email notification can also be made part of a workflow and can be controlled using logic built with worksheet formulas.

**Batch Process**

Batch processing is another important feature of SpreadsheetWEB. It is designed to run simulations and what-if analyses with applications developed with SpreadsheetWEB. A web-based process wizard allows users to upload a flat data file with fields matching the inputs of SpreadsheetWEB application. Upon the execution of the process, each record, i.e. row, in the data file is passed to the application as inputs and selected results are saved in an output file.

# Typical Insurance BOAs

## Rating

Rating is one of the most critical business processes in any insurance organization. It directly determines a carrier's revenue and competitiveness. Rating algorithms are typically developed by actuaries and incorporate complex calculations based on various data. In essence, rating algorithms are one of the most prized intellectual properties of a carrier.

In technical terms, a rating engine is a software program that returns results based on programmed logic for a given set of inputs. In some cases, it requires database connectivity; in others, stand-alone.

Most actuaries develop their rating algorithms in spreadsheets. The modeling capabilities of spreadsheet software used in conjunction with the many built-in formulas enables the development of rating algorithms for even the most complex insurance lines, providing a single source for all rating regardless of complexity.

SpreadsheetWEB provides an ideal environment for actuarial departments to build and test their rating engines. Batch processing capabilities allows actuaries to run simulations over a book of business and test the effect of various parameters and versions of their rating algorithms. SpreadsheetWEB's database integration features allow extraction of data used in rating calculations from databases in real-time. This allows actuaries to build their rating engines without having to place large amounts of data increasing the size and reducing the performance of their spreadsheet applications.

SpreadsheetWEB also replaces existing Excel/Access based rating applications that are used by underwriters and brokers. With its web-based front-end, built-in database and workflow capabilities, it allows business units to build, deploy and maintain their rating applications without having to depend on IT resources.

Integration capabilities of SpreadsheetWEB allows rating engines developed and maintained by actuaries to be used by other systems within the organization such as policy administration, renewal, and underwriting applications. By centralizing rating to a single rating engine, it eliminates duplicate development efforts and reduces possible rate mismatches between those systems.

## Claims Analysis

Actuarial and underwriting departments need to analyze large amounts of claim data to identify trends and understand the success of their products by various factors such as geography, demographics, seasonality, etc. They typically use business intelligence and data visualization software to better understand data. However, most software packages lack the flexibility to build complex analyses capabilities. Spreadsheets offer the highest level of flexibility and shortest learning curve to business users for building rules engines of practically any level of complexity.

SpreadsheetWEB combines data analysis and data visualizing providing business users all-in-one system that they can analyze claims data using complex rule engines.

## Data Scrubbing

Wikipedia describes Data Scrubbing as "the process of fixing or eliminating individual pieces of data that are incorrect, incomplete or duplicated before the data is passed to a data warehouse or another application".

Many BOAs rely on data extracts from data warehouses. In most cases, data has imperfections. Tommy Peterson [(2)] identifies following as the cause of data issues:

- Poor data entry, which includes misspellings, typos and transpositions, and variations in spelling or naming.
- Data missing from database fields.
- Lack of companywide or industry-wide data coding standards.
- Multiple databases scattered throughout different departments or organizations, with the data in each structured according to the idiosyncratic rules of that particular database.
- Older systems that contain poorly documented or obsolete data.

Automated data scrubbing requires incorporating various business rules such as if-then-else logic. Such logic is built by business users and requires many trials to optimize data scrubbing algorithms. Excel provides the best environment for business users to build and modify such algorithms.

SpreadsheetWEB allows business users to build and execute data scrubbing processes. Business users first build a data scrubbing engine in Excel using any built-in worksheet formulas and IF statements and upload them to the server. Data extracts from the data warehouse can then be run against the scrubbing engine. Business users can make adjustments on the scrubbing engine in a familiar spreadsheet environment upon reviewing results of the batch process.

## Reporting

Most BOAs are developed for reporting purposes. This is mainly because reporting needs of business units evolve rapidly and department needs are so immediate creating impossible scenarios for IT processes to keep up with. As a result, business units build their own Excel/Access based reporting tools based on periodical data extracts from data warehouses. Advanced reporting and charting features makes Excel an ideal platform for building those reports. Business users can adjust those reports quickly depending on changing needs and requirements.

Since reporting BOAs reside on individual desktops, they typically modify reports depending on their specific needs. This leads to an unmanageable number of different versions of such reports. Any discrepancies in final results would be difficult to trace back to original source. This can cause regulatory and compliance issues depending on the type of data used.

SpreadsheetWEB provides the ideal platform for building and maintaining reporting BOAs. SpreadsheetWEB can turn reports, and dashboards created in Excel into browser based reports. Built-in database features of SpreadsheetWEB allow for the creation of reports in Excel that query databases in real-time.

While still providing the level of flexibility to adjust reports in a timely manner, SpreadsheetWEB provides the necessary audit trail for regulatory compliance. Business users continue building and maintaining their reports in familiar Excel environment.

SpreadsheetWEB also provides role based security features enabling access to sensitive data and reports which can be controlled easily through user management module of SpreadsheetWEB.

## Data Collection and Collaboration Applications

Data collection and collaboration is a very common task in insurance organizations. Various tasks are shared among multiple people across business units and departments. These tasks are usually handled manually. Often spreadsheets are used to collect data and are shared by multiple people by either emailing or placing them on shared drives. This often leads to multiple versions, overwritten and lost data. Due to lack of collaboration, these simple tasks often consume valuable time and resources needlessly causing frustration both internally and externally from customers who are expecting receive the outcome of processes such as quoting and underwriting operations in a timely manner.

SpreadsheetWEB automates this process by turning those spreadsheet based data forms into web forms that can be shared by multiple users across the organization.

SpreadsheetWEB also incorporates features that allow hiding or disabling certain data fields depending on user credentials.

# References

[1)] J-Walk & Associates. "Spreadsheet Protection: FAQ"
http://spreadsheetpage.com/index.php/tip/spreadsheet_protection_faq1/

[2)] Tommy Peterson, "Data Scrubbing". Computer World, Feb 10, 2003