

DITCH THE SPREADSHEET

Microsoft Excel has flooded the world in data. So how do you build an information ark to survive the deluge?

There is a complaint that I hear often in the corporate corridors and meeting rooms that I frequent; “We’re data rich but information poor!” The cry is inevitably from some beleaguered executive attempting to make sense of yet another voluminous spreadsheet comprising hundreds of rows replete with undecipherable formulae and references.

The concerns are always the same: have I got the latest version? How do I know if these formulae are correct? Where did this data come from? How reliable is it? Who else has been editing this document? Which tabs are the ‘live’ ones? Where do I make my changes? Who do I send this to next?

Excel is ubiquitous. It’s used in every organisation that I’ve ever worked for, as a calculator, as a modelling tool, as a notepad, a messaging system, a scheduling app, a planner, a business intelligence tool, and yes, as a *database* too.

Consultants like me are fond of saying ‘Microsoft Excel is the most popular database on the planet’. Which is an odd thing to admit to, especially as Excel isn’t a database at all, and more especially because Microsoft is also the author of two of the most popular *actual* database products on the planet: SQL Server and Access. One wonders how frustrated the product architects must be to know that 99 times out of a 100, users will choose to implement Excel rather than Access as their desktop database solution. Yet in my experience, wherever there is a business processes whose data handling requirements are supported by an Excel spreadsheet, there is a high probability of waste being generated, or that the process adds less value than it could. Excel – when used as a database - is almost always an indicator of underlying or latent inefficiency.

That’s not to say that Excel doesn’t have a place on the corporate desktop. It does, and I am actually a big fan. When deployed appropriately, Excel adds value and reduces cost; the problem is that it is rarely deployed appropriately. Instead, it’s usually the ‘go to’ solution for every project irrespective of its suitability for the task at hand.

WHY IS EXCEL SO POPULAR?

The reasons for Excel’s emergence as the de-facto solution for data handling are manifold. It is by any measure a superb application, well-engineered and more or less bug free. In the right hands, it provides more functionality per \$ than any other piece of software that I can think of. Building a case for its deployment is remarkably straightforward:

- **Simplicity**
It’s easy to get up and running. You can start entering data immediately. The learning curve is very gentle for most of the things you want to do.
- **Ubiquitous**
Everyone has a copy already. It’s on all of the company’s desktop PCs and laptops, and most tablets and smartphones either have a version installed or are equipped with an app that can display spreadsheets easily.
- **Low cost**
Other than the cost of an Office 365 license there is usually no additional investment required, so there is no need for a capex proposal or a new line in the budget – important considerations in the corporate world.
- **Self-sufficiency**
Deploying Excel usually requires no IT Department approval or input. In fact, Excel users generally work autonomously, supporting their own spreadsheets independently of the IT projects and support teams.

- **Collaborative**
Data stored in Excel can be shared using the simplest and most common technologies such as email or a shared drive on a server.
- **Rapid results**
Excel is an instant solution, you open a spreadsheet and you start typing and that’s it! You’re up and running straight away. You can have the data on everyone’s desk within seconds of collecting it.

There appears to be a compelling case in favour of the spreadsheet-as-database. But the benefits and advantages of Excel camouflage significant problems which inevitably present themselves once their parent projects and processes develop and mature.

THE CASE AGAINST

- **Simplicity**
Excel is *so* simple to get up and running that the temptation is to race ahead without planning anything. Often, not enough thought is given to the design of the data tables; there isn’t any *Information Architecture* taking place which leads to poorly structure data that results in confusing information.
- **Ubiquitous**
Because everyone has a copy of Excel, everyone thinks that they know how to use it. And while most people would admit that ‘they’re not experts’ it doesn’t stop users from editing worksheets that they don’t understand properly, or over-typing formulae by mistake or not adopting the optimum

techniques for securing and protecting their data in the first place.

- **Low cost**
There is rarely any cost/benefit analysis involved in initiating an Excel-based project. No one feels the need to assess the *true* cost of deploying a spreadsheet as the project's database, which may turn out to be considerable once the waste associated with re-work, transcription errors, version control, and bug fixes are taken into consideration. These costs are usually hidden and, with no oversight by the IT department to worry about, will often remain unknown for the duration of the project and beyond.
- **Self-sufficiency**
As users are able to work autonomously with Excel there is a tendency for 'islands of information' to form within the business. In the absence of technical management oversight users tend to decide for themselves how data is edited and presented and who should have access. This *freedom* often manifests itself as a lack of 'joined up thinking' leading to overlapping or duplicated processes and unnecessary data replication. Furthermore, opportunities to interface with existing systems and databases can be missed resulting in data being re-entered or manually transferred from one system to another. Another potentially costly side-effect of the lack of technical oversight is a tendency to ignore best practise, particularly with regard to security and business continuity. Spreadsheets are often sent unencrypted over unsecured email networks, and, in some organisations, may not even be backed up in a methodical and logical manner.
- **Collaborative**
The ease with which Excel documents can be shared gives rise to a number of problems. Many companies allow spreadsheets to be emailed between users on an ad-hoc basis which introduces the challenge of version control. As soon as a spreadsheet is emailed, two new copies of the document are spawned: one on the recipient's email message, and one that the user saves locally. It's too easy to lose track of which one is the latest, which one was edited most recently and which one has been superseded by another, more recent, email. The usual work-around for this confusing situation is to agree on a 'master copy' being held on a shared resource and for users to take turns to edit that single document, but this solution only works when the need to access the document for editing is occasional. Small businesses and departments might get away with this but it is an unscalable solution and usually unsuitable for corporate deployment.
- **Rapid results**
Excel is a rapid deployment tool, but rapid isn't always better. Too often spreadsheets are shared without any kind of quality assurance. Users without a technical or software background may

not understand the need to test formulae and pivot tables thoroughly before publication. Furthermore, the larger the data sets the more difficult and unwieldy the spreadsheet becomes, especially where multiple worksheets are used with their confusing cross-references and look-up tables. Yes, you *can* have the data on everyone's desk within seconds of collecting it – but is the information correct? And what are the consequences if it isn't?

It's no coincidence that the pitfalls of deploying Excel as a database mirror the advantages. This is because the problem isn't Excel per se. The problem is how and when it's used. The adage that '*when the only tool you have is a hammer, everything starts to look like a nail*' could have been composed for these business cases.

While the Excel application itself is reliable and robust, the spreadsheets that people create using it are too often fragile. There is no easy way to trace where the data has come from, and with no concept in Excel of a *transaction* or an *audit trail* it's too easy to overtyping the contents of a cell without realising it. You may never discover that a colleague was inadvertently typing dates into a value field. Furthermore, reconciling and testing a large spreadsheet is a daunting task that few people are prepared or qualified to undertake. Where you find spreadsheets you're invariably in the presence of errors – it's just that you have no way of discovering where they lie.

WHAT IS THE ALTERNATIVE?

Over the last two or three years a number of powerful, low-cost, cloud-based tools have emerged which offer businesses the opportunity to build *real* applications to meet the needs of their departments and workgroups. The solutions that can be created with this new generation of on-line tools are robust, secure and reliable, yet they also offer many of the same advantages that spreadsheets do, but without the corresponding disadvantages.

Like spreadsheets, these tools are simple to use, low cost and quick to deploy. They empower non-technical users to create sophisticated applications without learning to code. Being cloud-based they can be deployed seamlessly on any device from desktop to tablet to mobile phone. Most importantly, they require little or no input from the IT department because there is no software to install, no servers to configure and no local data to back up. The solutions are genuinely multi-user in that they can support many users editing, creating and viewing the data simultaneously and they are scalable: suitable for one user or a thousand users. The only application a user needs to have installed on their device is a web browser.

Two of the most successful players in this market sector are Knack* and QuickBase. Of the two Knack* is perhaps the most interesting success story: the company was formed just over three years ago in New York by a small team of frustrated web developers. Fed

up with constantly writing and re-writing the same application under a different guise they asked themselves a simple question:

“What if we could build a web application that could build web applications?”

This simple idea would result in the creation of a real killer app which within 18 months would be adopted by thousands of businesses around the world. Knack* is now used by blue chips and start-ups alike, including many household-name corporations such as:

Intel	Honda
Tesla	Leeds City Council
Hewlett Packard	General Electric
Cambridge University Press	VMWare
Harvard University	CapitalOne
Oxfam	Panasonic
Spotify	The Salvation Army

Knack* has enabled these businesses to empower ordinary staff members to build their own cloud-based apps in areas as diverse as inventory management, CRM, project management, scheduling, job costing, product planning, downtime recording and business intelligence. The key to its success has been ease of use; Knack* allows non-technical staff to quickly build beautifully simple applications which can be tested easily and deployed rapidly. Typically, these apps fill the gaps between enterprise software footprints, but do so much more elegantly and reliably than spreadsheets like Excel (or even Google Docs for that matter) ever could.

A TYPICAL KNACK* PROJECT

The gestation of a Knack* project will start in much the same way as an Excel database project. Someone needs to collect and manipulate data and then communicate the results. No one wants to create a software development request as that’s likely to sit in the IT Department’s pending folder for longer than the project will take to complete. Besides there probably isn’t time to scope out everything that is necessary at this stage, the team may not yet be ready or qualified to specify what they need in detail. What is required is something to ‘get started with’, something that can then be developed and fine-tuned as the project progresses. This is usually where Excel enters stage right, and this is where Knack* can provide a superior solution – and by superior, I mean more flexible, more cost effective, less prone to error, easier to deploy, secure and simpler to build and use.

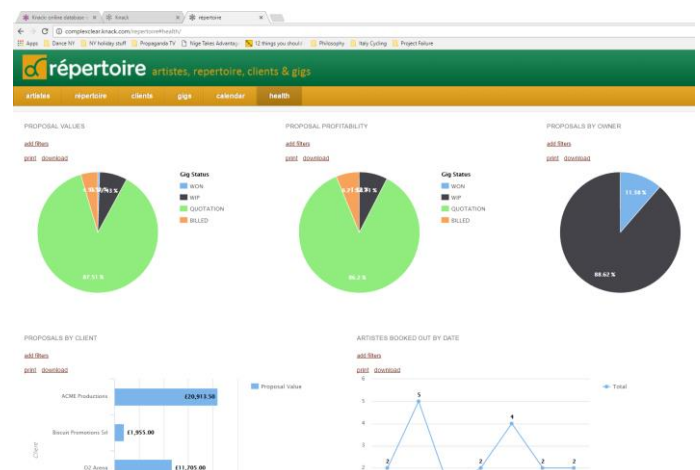
I should declare an interest at this point. I have been using Knack* for over a year now and I am a huge admirer of the product. I have spoken to the founders, Eric and Brandon, via Skype and have found them to be smart, dedicated, customer-focused and passionate about their application. They have been hugely successful and deserve every accolade that they are presented with. Knack* is a

game changer, so much so that it has already started to spawn copy-cat and ‘me too’ competitors. The press is referring to this as the emerging ‘age of the citizen developer’ and the beginning of a new era in ‘frictionless data’. It sounds grandiose, but there is real potential in Knack* and products like it to change the way businesses deal with data that doesn’t form part of the corporate data repository.

A typical Knack* project will be concerned with data collection, usually via input forms on a web-based application (see the example below) followed by data presentation and analysis. Like Excel, Knack* provides a suite of standard charts and pivot table views to visualise information, though the key differentiator is that all of the analysis and ‘drilling down’ is done via an intuitive user interface rather than a complex spreadsheet.

A typical Knack data capture form*

A key advantage of using a real database to store information is that it is transactional. This means that any change to the data is audited so that you know who added, deleted or edited a record and when it was done. This permits far greater data integrity than could be achieved with a shared spreadsheet.



*A typical data visualisation form built in Knack**

The authors of Knack* have also given a good deal of thought to the typical projects that are likely to be supported by a Knack* application and have created useful ‘wizards’ that can import a Spreadsheet and build a basic Knack* database complete with input

forms and views completely automatically. This allows new Knack* developers to get up and running quickly.

Mindful of the advantage of customers being able to integrate Knack* into their existing websites the developers have also made it very easy to embed a Knack* application into a standard web form. An embedded Knack* application also 'inherits' the style sheet of the host website so that it takes on the appearance of the client's corporate identity with regard to fonts and colours. There are also powerful integration features to enable Knack* applications to 'talk with' other on-line systems. While these capabilities are generally for the more advanced or technically experienced user, they do allow Knack* to operate seamlessly with other hosted systems such as Salesforce, XERO, Google Docs, MailChimp, DropBox, PayPal, Facebook etc. as well as leading commercial databases such as Oracle and Amazon Dynamo.

THE RISE OF 'FRICTIONLESS DATA' AND THE 'CITIZEN DEVELOPER'

The emergence of simple, cloud-hosted database applications such as Knack* has led to many businesses re-evaluating their approach to data collection and presentation. Excel is no longer the only option, and certainly shouldn't be considered the *default* option. In addition to the clear technical, security and productivity advantages there are other benefits to the adoption of cloud technology which shouldn't be overlooked:

Teams given the creative freedom to design their own solutions, or at least to influence the way in which their applications are built, experience a significant productivity-increase. Part of this comes from the empowerment that using 'the right tool for the job' brings (the hammer is no longer the only tool in the box) but much of the morale boosting effect is a result of being liberated from the strength-sapping delays that are part and parcel of having to deal with an over-worked, under-resourced IT department. Many companies allow their 'citizen developers' to build non-core solutions entirely independently of their IT function. As cloud-based applications require only a web browser and an internet connection there may be no compelling need to initiate an 'IT project' at all.

It seems that the revolution in cloud applications has started and that Excel may be the first casualty. This is a good thing for most businesses. Excel still has its place, but that place is for a niche *spreadsheet* application; a calculating, analysing, modelling 'what if' tool, not a database. Data collection and management is best handled by a real database, one that can enforce integrity, validate inputs effectively and secure corporate information reliably. In the past, systems like this were beyond the reach of most of the business, only the IT Department had the necessary skills and underlying technology to create and support such a solution. Fortunately, things have moved on and there are leaner, more creative and more productive options available.

- **Mike Moore, January 2017**
Managing Partner
Mike Moore & Associates

REFERENCES

The following solutions are part of the 'code-free, cloud-based revolution' empowering users to create their own low-cost applications and systems without the need to learn SQL, C++ or any other programming language:

Knack	www.knackhq.com
QuickBase	www.quickbase.com
Zapier	www.zapier.com
Kintone	www.kintone.com
Wufoo	www.wufoo.com
Track Via	www.trackvia.com
Caspio	www.caspio.com
Canvas	www.gocanvas.com
Device Magic	www.devicemagic.com