Why banks use spreadsheets

Dean Buckner
Financial Services Authority

July 2004

Outline

- Distinguish business critical from non-business critical applications
- How banks typically use spreadsheets
- Two case studies
- The cost of controls

Business Critical Applications

Non business critical applications

- **Spreadsheets used to solve some (typically numerical) problem**
  - Single user
  - Used once only
  - Typically thrown away after use
  - Typically a model of some kind
  - User knows the “right” answer

Example: The mother of all spreadsheets

- **BASEL II model**
  - QIS 3
- **Took 6 months to develop**
  - Team of people
- **2 Gb**
  - Needed dedicated server!
- **Still "disposable"**

Business Critical Applications

- **Any system whose function is critical to running of business**
  - More than one user
    - Probably doesn't know "right answer"
  - Performing cyclical task
  - Sometimes model
  - But usually data processing of some kind
    - Aggregation
    - Enrichment
    - Checking
    - MIS

Business Critical Applications

- **Financial regulators are worried about business critical applications**
  - In banking businesses, they are critical to the stability of financial markets, and the fight against financial crime
- **Financial regulators are not so worried about non-business critical applications**
  - These are not, by definition, critical systems

Examples

- Aggregation and MIS
  - **"Currently, regulatory reporting … is heavily reliant on spreadsheets. Whilst amendments to systems necessary for compliance with BASEL II … will significantly enhance our ability to automatically feed returns we cannot envisage that manual intervention won't be necessary."**

Data manipulation

- **Pivoting**
  - Column-row format changes
- **Mapping**
  - E.g. change “DEM” to “EUR”
- **Consolidating**
  - E.g. “short 100,000” to “-100000.00”
- **Enrichment**
  - User types or copies

Calculation

- **Creation of secondary or “derived” data set from primary set**
  - Very often involves matrix multiplication on large data sets
  - 50Mb spreadsheets
  - Common use of replicated formulae

Spreadsheets as GUI

- **In my experience, the most common form of spreadsheet application**
- **Spreadsheet is used as interface to**
  - Standard, robust database back end
  - Applications library in C++ with standard change control
  - Spreadsheet calls functions, reads and writes data to
  - Allows user to interact with powerful system in way that conventional GUI's don't allow

Case study

- In reaction to the rising trend of spreadsheet trades, [firm] has made its primary goal for [project] in 2004 to implement the new [ ] systems for the principal volume trading areas. Even with these roll-outs taking place, they still forecast the number of trades on spreadsheets may be higher by the end of this year compared to 2003's year-end figure, given that overall business volumes are expected to continue to grow. However, [firm] assure us that the derivatives operating environment continues to be adequate for current and envisaged trading levels, given other mitigating controls that are place, such as a regular programme of structured trade reviews, undertaken by product control.

Regulatory concern

- FSA to receive quarterly sign-off from European CEO on the adequacy of the operating environment for [firm's business]. In addition, we will also ask [firm's] senior management to provide their assessment of the progress they expect [project name] to have achieved by the end of this year in reducing operational risks caused by the use of spreadsheets as sources for risk/valuation data.
- … we will ask for them to provide to us year-end forecasts for certain operational metrics to enable us to more accurately quantify the progress being made.
- We will take the decision on what form this review should take, [use of internal audit or a Section 166 report], following update Q3 2004.

Replacement strategy

- **Firm's primary goal for [project] in 2004 is implementing [Sys1] and [Sys2] for the principal volume trading areas**
- **VERY expensive project**
- **However, even with these roll-outs taking place, they still forecast the number of trades on spreadsheets to be higher by the end of this year compared to 2003's year-end figure.**

FSA metrics

- **number of high-risk spreadsheets;**
- **number of external trades valued on raw spreadsheets;**
- **number of spreadsheets over 40 Mb.**

Good news, bad news

- **Good news**
  - The total number of spreadsheets remained broadly flat over last year.
- **Bad news**
  - the number of trades valued and risk managed on spreadsheets rose throughout 2003 and now stands in the tens of thousands.

Case study 2

- **Firm uses interconnected spreadsheet system to integrate its Value at Risk system**
- **Fully developed end-user development policy**
- **No intention to replace**

Costs and benefits

- **Panko's "iron law"**
  - All systems and controls have a cost
  - No error-checking for free
- **But spreadsheet systems are *much* cheaper than single systems solution**
- **Question**
  - Is cost difference explained by the cost of systems and controls?

The cost of a system

- **System cost = cost of controls + cost of functionality + slippage cost**
- **Where**
  - cost of controls is cost associated with error trapping, code reading, change control,
  - Cost of functionality is cost of “doing the business”
  - Slippage cost is billable hours “wasted”
- **Question: do large systems have high slippage costs that explain the discrepancy with spreadsheet solutions?**

Single system projects

- **Can be wasteful because**
  - They require large infrastructure
  - Many chiefs, few Indians
  - Find it difficult to interact with business users
  - Find it difficult to understand business requirements

User-developed systems

- **Can be efficient because**
  - Users understand exactly their own requirements
  - Have incentive to keep costs low
  - Do not need project infrastructure, hierarchy &c

However

- **There is also strong evidence that user developed systems have few of the standard controls associated with "proper" systems**
- **FSA "fit for purpose" test**

Conclusion

- **Little research on code production efficiency**
  - Software theft example
- **At a guess, non-user developed systems are too expensive**
  - Because of slippage and waste
- **User developed systems are too cheap**
  - Because users no real incentive to impose controls

And finally …

- **=+'S:\DTM\Ps\Subjects\RISK\Retail\mortgage lending statistics\Mortgage stats\[Summary Q4 2002.xls]Summary'!$D$52+'S:\DTM\Ps\Subjects\RISK\Retail\mortgage lending statistics\Mortgage stats\[Summary Q4 2002.xls]Summary'!$D$51+'S:\DTM\Ps\Subjects\RISK\Retail\mortgage lending statistics\Mortgage stats\[Summary Q4 2002.xls]Summary'!$D$50+'S:\DTM\Ps\Subjects\RISK\Retail\mortgage lending statistics\Mortgage stats\[Summary Q4 2002.xls]Summary'!$E$50+'S:\DTM\Ps\Subjects\RISK\Retail\mortgage lending statistics\Mortgage stats\[Summary Q4 2002.xls]Summary'!$E$49+'S:\DTM\Ps\Subjects\RISK\Retail\mortgage lending statistics\Mortgage stats\[Summary Q4 2002.xls]Summary'!$F$49**


FSA
Questions & Comments