

Introduction

Many spreadsheets are created with little thought given to design, with inadequate or non-existent documentation, with no integrity checks, by people who have had little or no training in their use. A few high-profile, costly spreadsheet errors hit the headlines from time to time, but also a vast number of less spectacular problems occur throughout businesses of all shapes and sizes.

The purpose of these Twenty principles for good spreadsheet practice is to help reduce the amount of time wasted, and the number of errors caused, by businesses (including accountancy practices) as a consequence of the way they and their employees use spreadsheets.

- See more at: <http://www.icaew.com/en/technical/information-technology/excel/twenty-principles#sthash.kMQrHaXH.dpuf>

The spreadsheet's business environment

1. Determine what role spreadsheets play in your business, and plan your spreadsheet standards and processes accordingly

If you have spreadsheets that play a key or critical role in your organization, ensure that they are developed and tested, managed and monitored to an appropriate level. Spreadsheets that form part of an organization's key business processes will need to be managed differently from ad hoc spreadsheets for short-term use by an individual.

2. Adopt a standard for your organization and stick to it.

This might be one that is developed in-house, or adopted from outside and shared with other organizations. A common standard within an organisation facilitates collaboration, aids understanding and saves development time. The standard should include, among other things, consistent conventions on use of cell formatting.

3. Ensure that everyone involved in the creation or use of spreadsheets has an appropriate level of knowledge and competence

For anyone designing, developing or maintaining (as distinct from just using) a spreadsheet, this will include: awareness of the range of functions available, clear understanding of such basic concepts as relative and absolute cell references, and an appreciation of the importance of carefully checking the results of functions.

4. Work collaboratively, share ownership, peer review.

The extent of collaboration and review needed will depend on the size and complexity of your organization and of each project.

Spreadsheet risks and controls

17. Have a system of backup and version control, which should be applied consistently within an organization. The appropriate levels of backup and version control will depend on the organization and the nature of the work, but there should always be, at the very least, a reliable means of preserving, identifying and restoring earlier versions of a workbook.

18. Rigorously test the workbook

The level of testing required will depend on the size, complexity and criticality of the workbook, with riskier workbooks needing a greater degree of independent testing.

19. Build in checks, controls and alerts from the outset and during the course of spreadsheet design.

These checks might include, for example, tests to ensure that a balance sheet balances, assets do not depreciate below zero, and so on. One approach would be to build in a set of audit tests to check validity and use flags to signal compliance or non-compliance. Use a master flag to summarize all the individual flags and place it prominently (on the output sheet, or even throughout the workbook eg, on sheet headers) so that users are bound to see it.

20. Protect parts of the workbook that are not supposed to be changed by users.

The level of protection will vary according to the nature of the spreadsheet and the kind of use/users it will have. It might include locking whole worksheets, all cells containing formulae, or everything except designated input cells.

Designing and building your spreadsheet

5. Before starting, satisfy yourself that a spreadsheet is appropriate tool to use.

Spreadsheets are not best solution for every problem. Lot's of time will be wasted when another application would be more appropriate. The more appropriate tool might be a word processor (if it's a table of text), a database (if processing large quantities of similar data items) or an existing software package (if it's a well-established processes, such as bookkeeping). Even if a spreadsheet is still the right answer it's worth looking for existing templates before starting a new one from scratch.. **6. Identify the Audience.**

If a spreadsheet is intended to be understood and used by others, the design should facilitate this. If the only 'audience' envisaged is yourself, you might perhaps justify less explanation and help. Even so, good documentation is helpful if you come back to a spreadsheet a while after you created it; and many spreadsheets come to have a much wider audience than originally intended.

Designing and building your spreadsheet (cont)

Ensure that adequate instructions, validation and help are included to promote ease of use and avoid input errors. Even if parts of a spreadsheet are 'locked', keep calculations visible..

7. Include an 'About'/'Welcome' sheet to document the spreadsheet

Include basic information as author, purpose, version number, and description of general approach. Also include explanations of color codes and other formatting conventions, any sources of input data (with, where appropriate, hyperlinks to the original data), and any macros and what they do. The more complex the workbook, or the more it needs to be shared, the greater the requirement for good documentation. Conversely, a simple

8. Design for Longevity.

Design spreadsheets to adapt to any reasonably foreseeable future changes in values (tax rates, etc) or volume (eg, items in a dataset) of data used in calculations. However, the need for adaptability should be balanced against following the Agile principle of 'The simplest thing that could possibly work'.

9. Focus on the required outputs

Work backwards: be clear about the purpose of the spreadsheet, what outputs achieve that purpose and therefore what inputs and logic are required to derive the outputs.

10. Separate and clearly identify inputs, workings and outputs.

A properly structured spreadsheet will be easier to understand and to maintain. If pivot tables are used, it may be possible to relax this principle, but clarity remains crucial. Design to ensure that any input should be entered only once.

11. Be consistent in structure

Use the same columns for the same things in each workbook, especially when working with time series. A consistent convention within a workbook reduces the risk of error where one sheet refers to another. For example, a common convention is that time flows horizontally from left to right (and a specific column is always 'Year 1') and calculations flow vertically from top to bottom. Such a structure will help to avoid circular references.

12. Be consistent in the use of formulae

On any worksheet use the smallest practicable number of different formulae. Where it is necessary to use different formulae, ensure that groups of cells using the different formulae are clearly separated.

13. Keep formulae as short and simple as practicable.*

Shorter formulae are easier to build (less likely to contain errors) and easier to understand/reviewed. Stage a calculation through multiple cells rather than building complex/long formula.

14. Never embed in a formula anything that might change or need to be changed.

Designing and building your spreadsheet (cont)

Instead, put such values into separate cells and reference them. This ensures that values enter the spreadsheet only once, and if change is needed would happen in just one place. It also allows for all formulae cells to be locked without denying access to input values.

15. Perform a calculation once and then refer back to that calculation.

Do not calculate the same value in multiple places (except perhaps for cross checking purposes). This reduces risk of error, is more efficient, as fewer calculations are being performed.

****16.** Avoid using advanced features where simpler features could achieve the same result. In particular, avoid using programming code unless necessary – in which case ensure that it is clearly documented within the code itself, as well as in a documentation worksheet. Similarly, avoid circular references, and control and document any exceptions. Do not change the software's key default settings (for example, do not turn off automatic recalculation) unless essential, in which case include a prominent message to warn users..