



COMPLIANCE WITH TWENTY PRINCIPLES FOR GOOD SPREADSHEET PRACTICE

SELF-ASSESSMENT (REVISED)

September 2014

1. Introduction

1.1. Background and Objectives of this Paper

Recently, we approached Paul Booth, a member of IT Faculty, expressing an interest in certifying FinRobot's modelling services to be compliant with the Twenty Principles for Good Spreadsheet Practice (herein, the 'Principles'). In particular, we are keen to have our standard set of four core models - as described further bellow - assessed for compliance with the Principles. To complete the first stage of the assessment process we were asked to conduct a self-assessment for further review by a panel member expert. This paper summarises our key findings from the self-assessment process conducted by FinRobot development team.

It is worth noting that the self-assessment was conducted with a view that not all of the Principles directly apply to our product as we do not seek to certify a modelling standard. Nevertheless, our analysis highlights some features of our product that help end-user organizations to comply with the Principles. As our product is aimed primarily at the SMEs sector we believe it to be relevant as smaller sized businesses may find it too costly to independently launch and to maintain their own modelling standard.

Examples of compliance are provided with reference to our flagship Base Model. However, we believe that the common platform, choice of coding and formulae would make our observations equally valid for the other three models from our core library.

1.2. General information on FinRobot and our offering

FinRobot (www.finrobot.com) is an automatic assembler of financial models in Microsoft Excel. It allows users to -

- choose from one of the four core modelling templates depending on drivers / model structure relevant to their business environment
- set online parameters to customize chosen template to better align with their business configuration
- download a working financial model in Microsoft Excel

[Quick Guide](#) located on the front page of our site walks users through required steps.

Whilst inputting online data users have an option to skip input fields and focus on structural configuration only. In this case, the final model is delivered with dummy operating and financial data, which can be replaced with real inputs off-line.

We further invested time and effort to make sure our models can be modified off-line. The code and choice of formulae are intentionally simple and transparent. Please see further considerations on this in point-by-point self-assessment immediately below.

Currently, FinRobot provides four versions of the core model called 'Base', 'Case Builder', 'TopLine' and 'Manufacturing'. As the names may suggest each template aims to address specific business or modelling requirements, a company sector, level of capital intensity, leverage, case scenario building capabilities and so on. Our [Model Wizard](#) assists users in choosing a template which is best suited to their circumstances. Alternatively, users can study full description of our models accompanied by screenshots and detailed manuals located in our [Main Library](#).

Our models have been successfully tested for Microsoft Office Excel 2007-2010. We recently tested our models for Excel 2013 and found no immediate compatibility issues.

2. Point-by-point Assessment of Compliance with Twenty Principles

The spreadsheet's business environment

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

Does not directly apply to our offering. However, we invested in online tools such as [Model Wizard](#) or [Bespoke Tailoring](#) to help SMEs determine which model is best suited to their business environment. Users can also ask us questions via email or Skype.

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

All our core models are built to the same standard including colour coding for inputs, mechanics of timelines, choice of formulae and many more common features further detailed in section covering 'Designing and building your spreadsheet'.

In addition to the above, it is worth mentioning that if a business decides to use any of our models they can jump-start their standardization and documentation processes by utilising our model's manual. As can be seen from example of Base Model [manual](#), it should be huge savings on time and effort to start with a 24-page document detailing every aspect of how the model is built and should be maintained.

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

Does not directly apply to our offering. Nevertheless, we spend a lot of time experimenting and testing to make sure that any laborious and/or complicated task is handled by our online assembler. The resulting tailored model does not have complicated formulae or complex rules, thus vastly reducing end user organization's need to maintain professional modelling personnel.

4. Work collaboratively, share ownership, peer review

Does not directly apply to our offering. However, our models have very strict one-way flows (time line and other global settings – operations – supporting calc – financial reports – analysis tools), designed and marked in a transparent way. In our experience, this feature allows not only collaboration between members of model building team / owners but across functional departments of the organization.

For example, commercial and engineering people may become more familiarized with various aspects of operational part of the model whilst finance people can focus on funding and liquidity aspects of the business.

Figure 1 immediately below illustrates data flows in our Base Model.

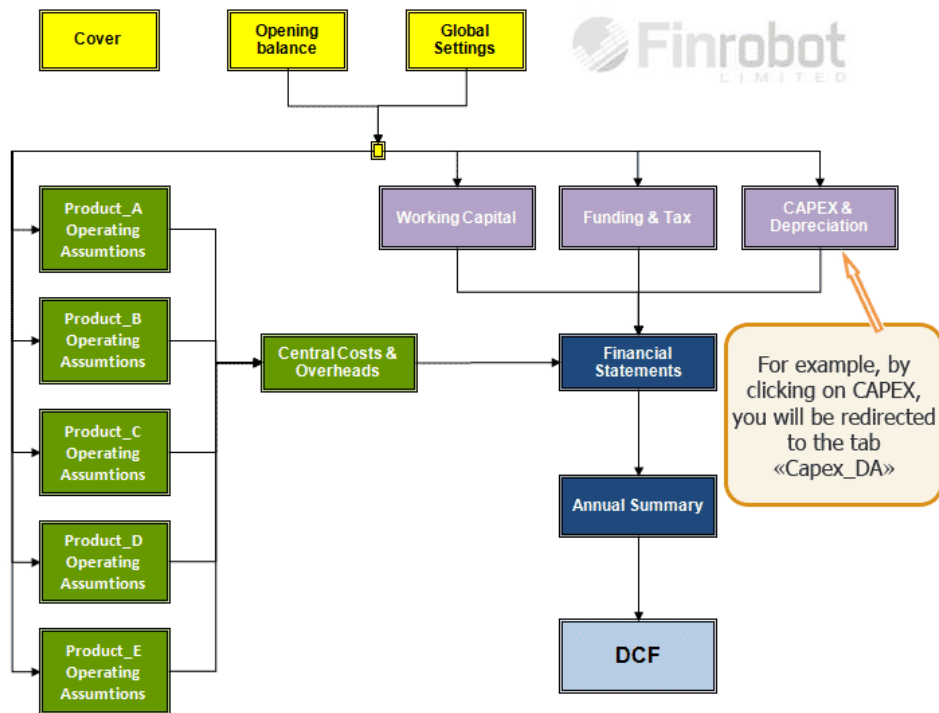


Figure 1. Example of hyperlinked map of Base Model.

With respect to peer review requirements, our offering benefits from the fact that no matter what the requested end model is the process always starts from the same template. Thus, any early unavoidable errors are corrected through testing and early adopters’ feedback. In an unfortunate event of a rogue cell being discovered, it is immediately corrected centrally on our servers for the benefit of all users. This should give our users great comfort that by assembling a unique model they still get something previously tested and proven to work properly.

Designing and building your spreadsheet

- 5. Before starting, satisfy yourself that a spreadsheet is the appropriate tool for the job**

See our response to principle number # 1.

- 6. Identify the audience. If a spreadsheet is intended to be understood and used by others, the design should facilitate this**

As mentioned above our Models are aimed at SMEs. The design, limited choice of formulae, easy one-click navigation, and absence of any hidden areas or computations ensures that anyone with basic understanding of commercial logic and accounting principles should be able to take full ownership of the model.

Figure 2 below illustrates our universal approach to clearly highlighting input fields. Any value in a cell with yellow background can be changed without any risk to structural integrity of a model. These can be easily spotted by least experienced users.

COGS Items	
Name or Name Range	Assign Value
COGS1	Raw Materials
COGS2	Personnel
COGS3	Fuel & Utilities
COGS4	Rent & Leases
COGS5	Other COGS

Any labels present in the fields with yellow background can be changed

Figure 2. Example of input cells format and colour coding.

Some areas of the model such as DCF requires more specialized knowledge, but this per se is not a modelling hurdle. Overall, anything that can ‘break’ or cause confusion is covered in detail in the model’s [manual](#) and our FAQ section of the website.

7. Include an ‘About’ or ‘Welcome’ sheet to document

Every model is equipped with a cover page as well as the clickable navigation page as shown in figure 1. Figure 3 shows example of a cover page.

Model for Financial Robot
Version: Base Model
Date: 25 december 2013

Upon the first view the FinRobot’s preset attributes of the Model are shown by default: Project and Model name and the current date

Figure 3. Example of Model’s cover or title page.

Note that we purposefully populate our standard cover with generic material. The advice given to users in the model’s manual that they should make this page relevant to their modelling objectives and processes.

8. Design for longevity

We pay a lot of attention to longevity of our models in a number of ways:

- Firstly, if our client organizations do not have means or resources to recut a model, they can always order a new version online. Provided that required changes are within the scope of our online algorithm, clients can get a new version built to familiar standard in matter of minutes (see Appendix One for a full list of structural options available for online model assembly);
- Secondly, our models allow for many non-structural changes without any need for a redesign. As clearly explained during online assembly stage and further detailed in the [manual](#), users can change their model’s timeline as well as re-label units, currency and most of balance sheet and cost line items. The central repository of such centrally controlled elements is in the tab called ‘global’, example of which is shown in Figure 4 below.

Period Start Date	Date_BoP	01 Sep 13	01 Sep 14	01 Sep 15
Period End Date	Date_EoP	31 Aug 13	31 Aug 14	31 Aug 15
Month Counter	Month_Count		12.0	24.0
Quarter Counter	Quarter_Count		4.0	8.0
Year Counter	Year_Count		1.0	2.0
Number of Periods Flag	Nperiod	1.0		
Time step duration, Days	step_d	365.0		
Time step name	step_name	Years		
Income Tax	CT	20.0%		
Currency	FX	USD		
Currency Unit	X	000s		

Figure 4. Example of centrally controlled inputs for a model's start date, income tax, currency and currency's / unit's scale.

- Thirdly, where practical every standard calculation block -be it a cost element, bank debt account or calculation of depreciation for an asset class – has row and column anchoring in a way that allows easy replication if a model requires further expansion.
- Fourthly, the sequential data flow in the model from operations via calculations to outputs is designed in a way to minimise links between tabs. For example, depreciation tab has a collection box with total depreciation wired to financials. Hence, if a new class of assets is added it would only require one additional ingredient in the summary depreciation box. Overall, reworks in one tab should not cause any collapse or major rewiring to other tabs of the model.
- Lastly, more advanced users can reset periodicity and length of the model's run. To be fair, unlike an instant re-plugging a model's start date, changes to forecast horizon or model's step would require a bit of working knowledge of the model. However, existing code is purposefully built and stress-tested to allow users' manipulation of the model's clock without any major rewiring or causing a model to collapse.

9. Focus on the required outputs

The suit of our core models are designed to cover generic financial modelling needs such as assessment of a new business project or to produce management forecasts and KPIs. Hence, the main built-in output is concise and readable financial statements. Additional overlay consists of discounted cash flow and IRR analysis. Nevertheless, as all intermediate items are clearly displayed users can easily build additional reports around existing structure of the model.

It is also worth noting that model's outputs are pre-formatted for easy cut-and-paste moving of relevant data to Word or PowerPoint documents to save time consumed by

preparing analytical reports such as management discussion and analysis, etc. Centrally controlled key variables and labels insure that changes are instantly and universally applied to all outputs.

10. Separate and clearly identify inputs, workings and outputs

Save for formatting of outputs (double lines, boxes around calculation blocks, etc.) our colour and formatting coding standard is limited to four options:

- Yellow shaded input cell with blue font
- Grey shaded link cell to highlight i) non-conforming cell in a row which should not be overwritten by accident (such as for opening balance sheet) or ii) a direct link to an input cell elsewhere in the spreadsheet
- Standard black font on regular background denotes regular formulae or a link to preceding block of calculations
- Green shaded cells denote error checks and warnings

As green and grey shaded fields are few the chosen colour mapping makes yellow input cells extremely visible and self-evident to any user.

Attention

It is worth noting upfront that we do not subscribe to certain popular school of modelling where all inputs are placed in one input tab. By experience, we find models designed in such away very cumbersome to work with. Instead, we place clearly defined input cells next to where they logically belong.

For example, operating assumptions for a product A would sit on the namesake tab whilst working capital drivers are positioned in the working capital tab. We believe users find it helpful to observe how the model instantly reacts to change in input values. The following Figure 5 illustrates our approach.

Product_A Operating Assumptions		Projected Fiscal Period Ending		
		aug.14	aug.15	aug.16
(USD in 000s, except otherwise stated)				
Volume	000s	800.0	832.0	865.3
Annual Growth Rate	% per annum		4.0%	4.0%
Average Price	USD	55.0	56.7	58.3
Annual Growth Rate	% per annum		3.0%	3.0%
Revenue	USD 000s	44 000.0	47 132.8	50 488.7
Annual Growth Rate	% per annum		7.1%	7.1%

Figure 5. Example of revenue inputs and revenue calculations for Product A.

The only exception to this approach is the layout of the tab called ‘Global’, which – as was mentioned above – is a central repository for variables and labels used repeatedly throughout the model.

11. Be consistent in structure

Consistency in structure is very strongly enforced in our templates. Save for cells shaded in grey signifying structural links (such as an opening balance sheet, etc.) any row in any block of calculations or outputs has identical formulae across the timeline (horizontally). Every time we release a new version of a model it is checked for formulae consistency across the board.

Likewise, every tab has identical row count and matching timeline. This insures that named range for dates and counters are guaranteed to work correctly in any tab. Likewise, this approach makes it is easy to audit/read date related functions or to set up a new tab with a timeline header at the top.

Consistent column structure also allows for applying changes to multiple tabs. Consistent labelling with help of ‘text’ & named inputs cells insures consistent presentation throughout the model.

The only exception to timeline consistency is the ‘Annual Financial Summary’ tab. The tab is designed to automatically pull annual data for monthly or quarterly model. Positioning of year-end columns of the ‘Annual Financial Summary’ tab would not match column count in the rest of the model. However, this time shift is clearly marked and explained in the [manual](#). In fact, users find it very helpful that tedious work of annual data aggregation is eliminated. Hence, we consider that benefits of this approach outweigh consistency argument.

Attention

12. Be consistent in the use of formulae

We limit our choice of formulae to simple ‘sum’, ‘if’, ‘min’, ‘max’ and a very selected use of ‘sumif’ and ‘offset’. ‘DCF’ tab uses NPV and XIRR functions for obvious advantages to arrive to a correct returns calculation.

13. Keep formulae short and simple

We generally subscribe and try to follow - where practical - a simple rule that any formula should not be longer than half the length of the single line of the formula editor window of Microsoft Excel.

Any database type of functions such as LOOKUP is strictly ruled out. Likewise, there are no pivot tables used in any model.

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

This rule is strongly enforced to a point that there are no embedded values in any cell save for +1 or -1 in obvious date or event related counters. Nothing else is permitted to be embedded. All formula ingredients are clearly identified as inputs or linked to preceding block of calculations.

That said, Base Model would still react properly to a change to its periodicity (e.g. change from a monthly model to a quarterly model). This is achieved by using one central input called ‘NPeriod’ and making sure that any calculation with a time component captures ‘NPeriod’ as an ingredient.

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

Our sequential model design insures that there is one single flow of data from inputs to intermediate calculations to outputs. We are strongly against any parallel calculations.

! Attention

With respect to always referring back to the original calculation we find that this principle – when enforced to the letter – conflicts with overarching objective to make every tab as transparent and as standalone as possible. Our approach tries to minimise chaotic links between tabs. For example, if EBITDA is picked up by DCF tab from Financials, then EBITDA based Terminal Value is computed based on EBITDA row present in DCF tab and is not linked back again to Financials.

This approach insures that i) maximum independent work is possible within any given tab, ii) usefulness of F2 function which highlight inputs within the same tab is maximised, and iii) commercial and accounting logic of entries is immediately obvious to a model user.

16. Avoid using advanced features where simpler features could achieve the same result

As mentioned above our models do not use pivots or database functions. These are or anything equally non-transparent is not allowed.

Attention

Since the commentary in the Principles specifically mentions circularity it is worth noting that we prefer to stand above the debate whether interest expense can be iterated or not. All our models have a switch, which allows users to go into iterative (circular) mode to perform interest charges calculations on average debt balances. However, default setting for the switch is to be off. Further, warning is given to users in the commentary box and in the [manual](#) that special consideration should be given, should they use iterative setting of Microsoft Excel.

Spreadsheet risks and controls

17. Have a system of backup and version control, which should be applied consistently within an organisation

Not directly applicable to our product. However, users can find it helpful that our website maintains their archived models. If anything happens to their downloaded copy they can always go back to their online user account and download original master.

18. Rigorously test the workbook

Not directly applicable to our product. Nevertheless, the concept where there is always one central template on our servers helps to weed out any rogue formula for the benefit of all users.

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

There are two types of checks built in. One flags if a balance sheet goes off in any given forecast period. The other - present at the top of every tab – instantly alerts users when one or more forecast balance sheet goes off anywhere in the model.

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

Every model is protected by a public key to avoid accidental overwrites and wipe-outs. Users are further advised to change password and keep tabs locked when not editing any calculation fields.

Appendix One

Structural Options available for Online Assembly – Base Model

Structural Element	Comment
Step Interval	Step interval can be set to year, quarter or month
Number of Forecast Periods	Can be set to any integer value between 3 and 60
Number of Product Lines	Can range from 1 to 5. Each product line has independent drivers for revenues and COGS, e.g. is modelled down to gross profit line
Number of Fixed Assets Items	Can be any integer value between 1 and 5. Each Fixed Asset Item has its own CapEx and DA profile
Number of Debt Instruments	Allowed range is between 1 and 5. Please note that the first debt element is always present and is an automatic cash sweep, or an overdraft facility. All additional debt elements have manual repayment schedules
Cost of Goods Sold Assumptions Fixed / Variable Driver	The fixed/variable driver input requires user to identify the type of driver for each COGS line item: fixed cost driver will forecast out at growth rates set by user; variable cost driver is modelled as a percentage of revenue (e.g. margin driven)
Overhead Assumptions Fixed / Variable Driver	The fixed/variable driver input requires user to identify the type of driver for each Overheads line item: fixed cost element will forecast out at growth rates set by user; variable cost element is modelled as a percentage of revenue (e.g. margin driven). This choice is structural and cannot be reversed once the Model is purchased.