ICAEW CORPORATE FINANCE FACULTY

RSM

Best practice guideline

Financial modelling

SECOND EDITION

Author: Alistair Hynd, RSM

July 2016 Issue 63

ICAEW

SPREADSHEET PRINCIPLES COMPLIANT

Acknowledgements
Thanks are due to my team, in particular Ben Ducker and Richard Hill, for their technical and practical input into producing this updated second edition. Katerina Joannou and David Petrie provided valuable comments on the first draft and I would also like to thank Rob Bayliss, Darrin Shipp and John Tennent from the various faculties' technical committees for their insightful peer review feedback. Finally, thanks are due to David Lyford-Smith and the Twenty Principles Recognition Committee from the IT Faculty.

ISBN 978-1-78363-668-6

Contents

Foreword

Welcome to the second edition of the ICAEW *Financial Modelling Best Practice Guideline*.

First published four years ago, in 2012, this document received a very positive reception on its publication from the corporate finance community, from our clients and from our competitors. We believe now is an appropriate time to revisit the guideline to ensure it remains relevant and helpful to its target audience of transactors and modellers.

Things move quickly in the world of corporate finance, and while the pace of change in financial modelling methods and techniques is much slower, enough has changed over that period to merit updating and reissuing the guideline.

- **Sharpened focus on risk**: since the first edition, some very high profile cases of flawed models have hit the headlines. Most notably the derailment of the West Coast Mainline franchise process and the hurried reissuance of guidance on gross domestic product (GDP) growth rates from some noted economists. We have certainly noticed that central government has become much more attuned to modelling risk in its procurements, reinforced by the 2013 Sir Nicholas Macpherson review of quality assurance of government analytical models.
- **Enhanced focus on standards**: a growing number of organisations have been promoting spreadsheet methodologies and good practice frameworks. The ICAEW IT Faculty published its own ***Twenty Principles for Good Spreadsheet Practice*** (the twenty principles) in 2014 on which we comment further in chapter two. We wholeheartedly endorse their adoption by any organisation that is serious about managing spreadsheet risk.
- **Continuing development of Microsoft Excel**: arguably the one thing that has not changed is that spreadsheets generally and Excel in particular, remain the tool of choice for financial modelling, especially on transactions. Microsoft continues to invest in Excel with a host of new features and tools in Excel 2016, the latest release.

Against this backdrop, in many respects modelling approaches have not changed materially, and there is increasing convergence among practitioners on what constitutes best practice. Witness the twenty principles publication referenced above, which was the result of a cross-firm, cross-industry collaboration.

This rather begs the question, so what's new in this second edition? Sticking to the old adage, "if it ain't broke, don't fix it", we have focused on:

- extending specific existing subject matter of most use to transactors, for example, dashboarding and model review techniques;
- adding new material, particularly looking at useful functions for corporate finance modellers, looking at how to enhance model speed, and comparing the features of the latest versions of Excel; and
- reviewing, clarifying and updating existing content generally, including referencing, linking, and highlighting the ICAEW twenty principles throughout the document.

On top of all this, we have rebranded globally as RSM. We hope you find this guideline useful and if you have feedback, questions, or would like help with a knotty modelling problem, we'd be delighted to hear from you – please contact me or one of the team.

Alistair Hynd
Partner
Head of Project Finance, Modelling and Valuations
RSM Corporate Finance LLP

1 Executive summary

1.1 Introduction

The purpose of this best-practice guideline is to provide a practical guide to financial modelling in a corporate finance transaction context. That rider is important. The guide is aimed at corporate financiers and modellers who need to prepare, use or review financial models for deals; it is not aimed at financial modellers who need to develop their understanding of corporate finance.

In a corporate finance setting, financial models are ubiquitous. Indeed, it is difficult to imagine a deal closed in recent years without one. In our experience most transaction models are developed as spreadsheets using Excel and that is the focus of this guideline and the context of much of the work that has gone into it.

There is a growing and improving body of literature on basic financial modelling practice and application – typically crafted by modellers for modellers. However, in that body of work, little has been said about the transaction-driven financial model or the particular demands of the corporate finance context.

This guideline therefore draws on the wider literature as well as our experience of developing and reviewing deal models, to provide guidance on the framework and context within which modelling for a corporate finance transaction takes place. It also provides practical advice on modelling techniques relevant to the deal setting. In particular, in this second edition, we have added new material on some of the more useful and commonly used functions, as well as giving additional focus to presentation and review techniques.

In this guideline the term financial modelling is used to refer to both business modelling and to financial modelling. Most transactional models will address both aspects: what does the expected performance of a business look like, and how does a given deal impact the cash flow generated by that business, and thus the sources of finance available to that business.

This guideline does not explicitly address economic, banking, statistical or risk modelling, though many of the best-practice principles discussed herein are portable to those contexts.

The audience for this guideline ranges from management teams embarking on a deal and for whom an understanding of the modelling process and risks is vital, through to transaction teams charged with developing financial models with which to assess a deal.

1.2 Financial models

Our working definition of financial modelling is the use of software to create decision support tools for businesses. Put simply, a financial model is a mathematical tool, typically a spreadsheet, used to calculate the financial impact of a business decision under a range of scenarios and to communicate the potential outcomes of that decision to interested stakeholders. The latter point is important: the clarity and user-friendliness of presentation is second only to the accuracy and integrity of the business logic.

Note also that a model is just that: a model. No model can hope to replicate reality precisely and nor should they. Model users need to recognise that a model is only a simulation of a deal or business, and further, will only be as good as the assumptions and data underpinning it.

In practice, models can be one of the most important components in a deal. They represent the meeting point where information on the commercial, operational, financial, legal, fiscal and economic factors relating to a deal, and their interrelationships, are synthesised and summarised to explain the consequences of a transaction (see Figure 1). Models are widely used and relied upon across the corporate finance world, in mergers, acquisitions, restructures, valuations, leveraged buy-outs, refinancing, securitisation, and project and infrastructure finance.

Figure 1: The central role of financial models

Financial modelling has advanced massively during the last 20 years, a function of improved software and processing power, emerging best-practice consensus and techniques, and the gradual professionalisation of the activity.

Financial models are a vitally important enabler to corporate finance transactions and we consider financial modelling a core skill for corporate financiers.

1.3 Modelling in a deal context

This guideline advocates strongly a structured and disciplined approach to modelling. If circumstances and timings sometimes militate against a controlled approach to developing a financial model, then corporate finance transactions only create additional pressures on the process:

- timetable – time periods tend to be compressed and deadlines less flexible;
- audience – deal models are frequently used, shared and relied on by multiple stakeholders;
- optionality – stakeholders typically require the ability to assess numerous options and 'what if?' variations;
- evolution – deals evolve and change shape as negotiations unfold and new information emerges which has to be incorporated; and
- uncertainty – business circumstances are often subject to a high degree of uncertainty, with analysis of the potential impact of uncertain or volatile assumptions a pre-requisite.

A minimum application of modelling best practice at an early stage can help alleviate the risks that arise when developing models in a fluid deal environment; thus saving time and cost, as well as avoiding financial loss.

1.4 Overview of this guideline

Part A covers the core elements of best-practice financial modelling, focusing in particular on the framework within which modelling takes place.

Part B is a 'How to' guide, providing practical guidance on applied and proven approaches to modelling, and to reviewing other people's financial models, that are especially relevant in a deal context.

In particular, Part B focuses on techniques that help modellers to:

- avoid introducing errors during development;
- make best use of Excel's functionality to solve certain common modelling challenges in a deal context;
- create user-friendly dashboards and present clear and understandable financial results to support the management decision-making process;
- control data and model versions;
- facilitate comparison and analysis of scenarios;
- improve Excel's run speed and calculation efficiency; and
- assist reviewers to identify errors.

This guideline is not an exhaustive treatise on financial modelling or on Excel functions and techniques and some of the material presented assumes a reasonable level of competency. There are some very good third-party reference sources and books available and the author of this guideline's contact details are listed at the end.

Financial modelling of deals generally takes place on open software platforms, with Excel the most common, and as such the variety and heterogeneity of models is practically limitless. As a result, while a number of broad assertions are made throughout this guideline, there will always be exceptions. Some of the exceptions will be valid. However, that does not mean to say there is not a core set of principles which can be rightly considered to represent general best practice.

PART A:
Best practice framework

2 Best practice financial modelling

2.1 Overview

This chapter sets out the context and need for a financial modelling best-practice framework:

- looking at what we mean by financial modelling;
- considering why financial models are used so widely;
- reviewing the issues involved in selecting a modelling platform and why spreadsheets have emerged as the preferred platform;
- looking at the benefits and risks attaching to models; and
- introducing the three elements that underpin best-practice financial modelling: people, process and principles.

2.2 What is financial modelling?

Financial modelling is the use of software to create decision support tools for businesses. Few models are created other than ultimately in support of investment decisions of one kind or another (whether this is explicitly understood to be the case or not). Those decisions range from allocation of scarce resources through the budgeting cycle, capital investment appraisals, fund raising, and project and infrastructure finance transactions, through to the evaluation of mergers and acquisitions, valuations, disposals, and restructures.

We use the term financial modelling interchangeably to refer to both:

- business modelling (ie, modelling the operational outcome of business decisions in financial terms); and
- financial modelling in the more narrowly understood use of the term (modelling the impact of business decisions on funding, financial instruments and finances).

Dictionary definition

Financial (adj):
Of revenue or money matters.

Model:
A simplified description, especially a mathematical one, of a system or process, to assist calculations and predictions.

Most transactional models will be concerned with both aspects: the baseline forecast performance, and the direct impact of a transaction on the cash flow generated by a business, as well as the consequential impact on its need for finance and its ability to service said finance. This is true both for the explicit servicing of, for example, tranches of debt finance, as well as for the implicit accretion of equity value as measured through discounted cash flow and/or multiples-based analysis.

The definition therefore encompasses simple cash flow forecasts, cost models, valuations, and pricing models, all of which inform a management or investment decision. However, in the context of this deal-oriented guideline, we focus on those financial models that are used to help businesses evaluate ‘macro’ level strategic corporate decisions.

In effect, we exclude from our working definition those models that support ‘micro’ level analysis, and which might inform tactical decisions or day-to-day management of the business. For example, management accounting tools, databases, and simple pricing templates. Having said that, most of the concepts and principles set out in this guideline remain valid in those other modelling contexts.

Financial models are the output or product of a financial modelling process. Typically, though not exclusively, built in Excel, financial models used in a deal context have two core objectives:

- to **calculate** the expected impact of a potential decision on those financial measures that matter to a given business and its stakeholders under a range of alternative views of the world; and

- to **communicate** those potential financial outcomes clearly and relevantly to the business and its stakeholders.

In achieving these two objectives, a good model should:

- be accurate and robust;
- present relevant decision information only, and with simple clarity; and
- instil confidence in the financial bases underpinning the decision process.

It is also the case that a good financial model should be a dynamic tool – modelling relationships and business drivers, rather than a static picture, reporting a stale snapshot of the world. The key point here is that the user can quickly explore the ripple effect of a change to one input or set of variables without having to make extensive changes. For example, changing a single cell to explore the impact of a percentage change in growth rates, rather than manually adjusting absolute sales, cash receipts and debtor figures in each period. This is a core principle that is explored in section 5.5.

While this guideline focuses on financial modelling in a spreadsheet context, it is important to note that financial modelling as a skill is quite distinct from Excel literacy or expertise.

2.3 Where, when and why are models used?

In a corporate finance context, the financial model is ubiquitous. We challenge the reader to identify a significant deal they have closed in recent years that has not had one. Even when there has not been a model in existence at the inception of a deal, you can virtually guarantee that the funders and financiers will require one during the deal process.

This applies whatever the sector and whatever the nature of the transaction.

In practice the model is typically one of the first meaningful deal documents produced, often as a kind of 'electronic napkin' on which the principals sketch out the shape of a deal or transaction. As deals evolve and gather momentum, models typically also change and expand to:

- accommodate new information made available;
- reflect the passage of time;
- adapt to changing deal structures; and
- enable the provision of ever-evolving 'what if?' analysis and emerging stakeholder information requirements.

Without a financial model it would be extremely difficult to track these aspects; it often remains difficult even with a model. While a financial model therefore needs to be constructed with due skill and care, this does not obviate the need for a robust and disciplined process (see chapter 4) surrounding its development, population and use.

The processes you implement to develop, test, control and monitor your models, as time passes and contexts change, largely depend on the roles played by your models. If the model is core to your organisation you may wish to apply higher standards than if it were a quick, point-in-time, analysis.

The pace and nature of change in a deal context brings with it the risks of misunderstandings as well as simple model-related errors. But the need to distil the essence of a business decision, all its variable outcomes, interdependent and complex business drivers, and, critically, the decision's likely financial impact, into a coherent and reliable picture, is why models continue to be used so extensively.

Indeed the role of the model is pivotal in most deals and with this in mind it is important to get the modelling right. Of course, the concept of 'right' in financial modelling can be a source of debate: can a model ever be truly right? After all, a model is just that, a simulation or approximation of reality. It is really important to bear this in mind and use and approach models in this light so as not to invite spurious accuracy.

2.4 What platforms and tools are used?

Financial models can be prepared using a variety of software tools and platforms. These range from spreadsheets (in some cases with specialist add-ins) through extensions, to management information systems, or enterprise resource planning packages (eg, Cognos, Oracle), to more bespoke solutions and forecasting software (eg, Sage 50 Forecasting, Cashflow Wizard, Brixx). In selecting a platform on which to develop a financial model, there are a number of considerations for the modeller (or organisation commissioning the model):

- **cost:** how much does the software cost, and is it an incremental and/or ongoing cost to the organisation?
- **compatibility/user base:** is the output compatible with other systems to upload, download, and review data, and does it need to be? In the context of a transaction where a model may be shared with multiple stakeholders, or a bid situation where multiple tenders may be received, it is crucial that the model can be reviewed by all relevant parties, and in many cases used and manipulated by them.
- **platform maturity:** related to the preceding points, is the software platform properly tried and tested, does it have wide market acceptance, and ready availability?
- **speed and flexibility:** will the software platform allow a usable, complete solution to be delivered within the deal timetable, and is it sufficiently flexible to cope with fast-evolving deal requirements, and to do so on a timely basis?
- **training/skills:** does the modeller or organisation have skills in the relevant platform, or is there a training requirement or cost implication? Is the platform accessible and intuitive to the audience?
- **specialist requirements:** once the need for a model has been identified and the specification drawn up (see chapter 4), consider whether a spreadsheet is the most appropriate platform or whether specialist or bespoke software is required. The specification should be scrutinised and challenged closely and it may be worth seeking external advice. The instances where a specialist package or bespoke software solution is needed to support financial modelling for a transaction or deal tend to be the exception.

In our experience most deal models are built using spreadsheets and we focus on the Excel spreadsheet platform in this guideline.

The reason for this is that Excel ticks most of the boxes above (speed, accessibility, compatibility, and cost), and is well suited for ad-hoc, flexible analysis. The two main reasons for using platforms other than Excel are, typically: compatibility with existing systems; or, for certain financial modelling software packages, a perception that they require less skill to program and use, and that they are lower risk solutions. Accordingly, they are often used where the subject matter is simpler, or for enterprise-wide budgeting and forecasting models. In practice, it remains very rare for anything other than Excel to be used in the development of deal models or high level strategic planning tools.

While the applied best-practice guidance in Part B relates to spreadsheets, many of the general principles around approach and process remain relevant whatever the software used.

2.5 Why does it matter?

Why does financial modelling matter so much? It matters for three closely linked reasons: the sheer ubiquity of financial models, the benefits that sound modelling and analysis can bring to the decision process, and the risks and threats to a given process from flawed financial models.

2.5.1 Benefits

A good, well developed financial model will augment the decision process. The underlying decision process has not fundamentally changed, what is different is that financial models:

- can aid clarity – presenting ordered and transparent results and an audit trail;
- can speed up the process – avoiding transaction delays and allowing time for further analysis;
- bring accuracy – enabling granularity of calculation and incorporation of detailed real-time data; and
- bring breadth – facilitating a broader range of 'what if?' analyses.

2.5.2 Risks

If spreadsheet models are used to support important decisions because of their inherent benefits and relatively ready availability, this must be balanced with an appreciation of the risks run in relying on models as widely and heavily as the corporate finance profession does.

The investment decision process can be visualised according to the steps shown in Figure 2. These steps have been augmented by the rise of spreadsheets, and the ability to create ever more sophisticated deal models.

The risks of using financial models are little different from those intrinsic to any investment decision process or indeed any human activity. The issue is that computers, spreadsheets and, by extension, financial models, allow us to make more errors more quickly.

Figure 2: The investment decision process

STAGE	DESCRIPTION	RISKS
Issue	The inception of the process; an initial view on the decision to be made/problem to be solved	Poor or ambiguous communication and/or direction
Identification of options	Review of the alternative courses of action and associated planning assumptions	Anchoring on first attractive solution Garbage in, garbage out
Evaluation and design	Comparative analysis of each of the options, comparing both financial and non-financial aspects	Optimism bias Weak or insufficient consideration of non-financial issues
Decision	Selection of the preferred option	Halo-effect: evidence distorted to favour preferred choice Fitness for purpose: the model is flawed or simply unable to answer the question
Implementation	Implementing the preferred option, measuring progress against the initial decision evaluation	Transaction focus means model unsuitable for post-deal monitoring
Monitoring/ feedback	Feedback of results to inform follow-up and future decisions and to enhance the decision process	Inflexibility, model not easily adapted for alternative uses/analysis Poor quality feedback

Modelling risks are real and persistent. But how do these risks manifest themselves? The impact of poor or misconceived financial modelling can be felt through:

- **Bad decisions and financial loss** – models can lead businesses into making poor decisions if, for example, they're flawed, overly simplistic (or complex!), become corrupted or are just based on incorrect data. In some cases actual losses may be made when a decision is based on a model that does not accurately reflect the interrelationships between business drivers.
- **Misrepresentation and reputational damage** – models are frequently used to present the actual and/or expected performance of a business to its external stakeholders; models that are wrong (or just poorly presented) can quickly erode confidence in the business, management team and board members.
- **Delay** – poor or inaccurate modelling can lead to delay and additional cost, either through reworking models or additional review effort. Delays can potentially also exacerbate exposure to market risk.
- **Litigation** – legal redress may result from any or all of the preceding risks crystallising. This risk is sharpened by the depth of audit trail that usually results from any modelling exercise: mistakes, however subtle, are there for stakeholders to pore over.

These risks are real and persistent. Research into financial modelling has repeatedly shown that pre-review, most models contain multiple, often significant coding errors. It is important therefore to understand the risks and underlying causes in order best to anticipate and avoid them.

2.6 Best practice framework

We see financial modelling as a core competency that corporate financiers need or need access to. As complexity increases, we increasingly see modelling as the preserve of dedicated specialist modellers or modelling teams, but even for simpler transactions, a basic level of competency is a key enabler to any transaction. A good quality model can greatly assist the smooth running of a transaction process, saving time and money and providing clarity to the decision process. A poor model can cause delay, uncertainty, and undermine confidence; at worst it can lead to financial loss.

Most models are built in Excel, an open software platform which requires the modeller to act as both programmer and data analyst. Many models are built as one-offs for a transaction. Modelling carries with it a high degree of risk, and the blank sheet provided by Excel, coupled with the extent of bespoke programming, only serves to exaggerate this risk.

It is vital that these risks are recognised and that financial modelling takes place within a structured and disciplined best practice framework. Best practice is not just about the modelling techniques and principles used; it is also about how the financial models are developed (the process followed) and who does the work (the people deployed and their skills).

We refer to these elements as the 3 Ps (see Figure 3). Each of them underpins the modelling workstream and each is equally capable of undermining the workstream if not carefully managed:

- **people**: the sponsor and staff involved;
- **processes**: the methodology or approach used; and
- **principles**: the practices and techniques applied.

For any deal, the financial modelling work stream itself is a product of these three elements and the principals of a transaction should pay due attention to each of them.

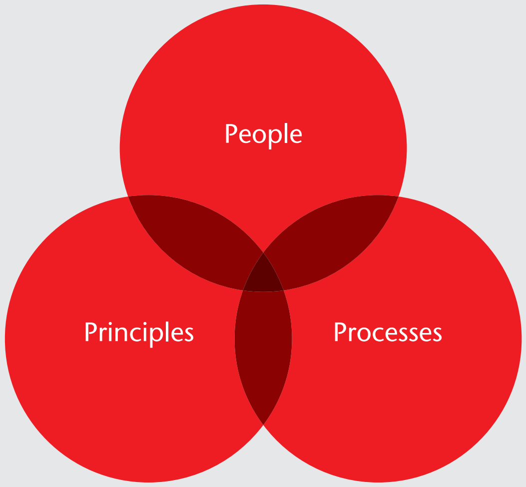
A war story

In 2012 the bidding process for the West Coast Main Line rail franchise was struck by problems as a result of four of the bidders being given erroneous information by civil servants when preparing their bids. Incorrect assumptions in the department's own model about what would happen to inflation and passenger numbers led to the TOC's offering far too little protection to taxpayers against the risks of collapse by a franchise holder. If the mistake had not been picked up, taxpayers would have been excessively exposed to potential losses at some point during the 15-year life of the franchise.

The bidding process was cancelled and several interim extensions to the franchise granted to the incumbent operator Virgin Trains, prior to a new invitation to tender expected in November 2016 for a contract to be awarded in November 2017. Bidders in the 2011/2012 process were reimbursed costs totalling £39.7m, with the original winner First Group being awarded an additional £4.8m for mobilisation costs. As these figures attest, the importance of a rigorous risk management process to prevent and mitigate the impact of potential errors cannot be underestimated.

Source: BBC / DfT

Figure 3: The three Ps



Subsequent chapters explore each of these aspects to explain their particular relevance in the context of commissioning, developing and reviewing financial models. The fundamental point of this best-practice framework is to reduce risk, and to aid efficiency. Happily, in a modelling context, these goals are mutually self-reinforcing.

2.7 ICAEW Twenty principles for good spreadsheet practice

ICAEW's IT Faculty launched the twenty principles in 2014 with the aim of helping businesses that use spreadsheets to work more efficiently with them, and to reduce the risk of errors arising from their use.

The principles are intended to promote an environment of control and managerial oversight, as well as practical principles of good spreadsheet design. In a similar way to this guideline, the twenty principles are not standards or specific prescriptive instructions, but rather guidance on best-practice techniques. The twenty principles are summarised at Appendix A.

The guidance given in this document is compatible with and consistent with the twenty principles. Wherever there is clear overlap and consistency with the twenty principles, we have included call-out boxes in the margin of this document (such as the adjacent) to provide a cross reference to the relevant principle.

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

Transactional financial modelling, as a primarily spreadsheet-based endeavour, naturally falls under the aegis of the twenty principles. We wholeheartedly support the twenty principles and highly recommend and encourage their adoption, particularly for corporate financiers and businesses that are serial or regular users of spreadsheet-based financial models.

2.8 Key points

- Financial modelling is the use of software to create decision support tools for businesses.
- Such models are used to calculate dynamically the financial impact of a potential decision (typically under a range of assumptions) and to communicate those results to relevant stakeholders.
- A model is just that: a simulation or approximation, and its limitations should be recognised.
- The corporate finance context brings particular challenges to the modelling process: time pressure, deal evolution, and multiple stakeholders.
- Most modelling is carried out in Excel, for reasons of speed, accessibility, compatibility and cost. Consider these dimensions carefully before adopting an alternative solution.
- Good financial modelling is taken for granted, but it can greatly enhance the decision-making process, bringing clarity, accuracy, speed, and breadth of analysis.
- Modelling is not without risks, which can manifest themselves at each point in the development cycle, and research reveals a high propensity for error.
- Risk can be mitigated by observing best practice in relation to people, processes, and principles.
- The ICAEW IT Faculty's ***Twenty Principles for Good Spreadsheet Practice*** aims to help organisations that use spreadsheets to work more efficiently and reduce the risk of errors.

3 People

3.1 Overview

This chapter reviews the different roles involved in the modelling process considering:

- who should be involved;
- what roles they should play;
- their responsibilities; and
- what skills are required.

3.2 Modelling roles

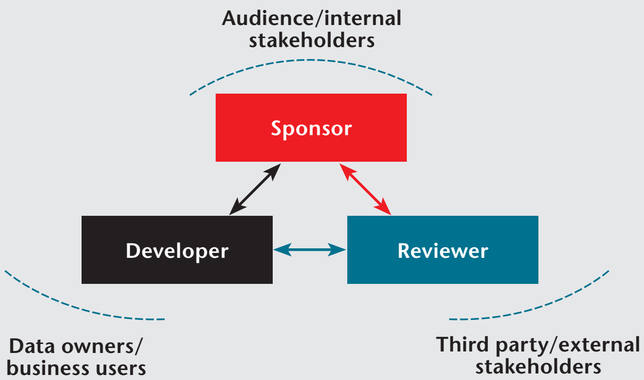
Financial modelling is a human endeavour because of the degree of judgement and interpretation required in order to evaluate the available information and data, and to assess how best to model the complex interrelationships that exist within a given business' data set.

In an ideal world, there would be three core roles in the model development process (Figure 4):

- **sponsor**: the party leading or commissioning the modelling work;
- **developer**: the model author; and
- **reviewer**: the party checking that the model is fit for purpose.

Note that we have intentionally ignored the concept of user as a discrete role in this guideline, since the concept is more elastic in a transactional context. Various parties to a deal may well use or operate the model at different points and we acknowledge these uses within the role descriptions below. A discrete definition of user is not needed; in practice, in most transactional contexts the developer will operate the model on behalf of the sponsor and is the user in the traditional sense.

Figure 4: Modelling roles



These roles are central to best practice modelling and those assigned responsibility for these roles will need an appropriate level of knowledge of the software, model structure, subject matter, and development process to perform their duties. However, their activity/ interrelationships do not take place in a vacuum. In commissioning and developing a model, consideration needs to be given to the wider stakeholder environment: the model's audience, third party users, and those providing source data and information for the model.

It is important that these roles share ownership of the deliverable, to maintain engagement and to promote best practice.

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

ICAEW Principle 4
Work collaboratively, share ownership, peer review.

Models can be commissioned for projects of varying scale, from simple businesses to extremely complex, large scale operations. Accordingly, it is sometimes necessary to deploy a team to build or review a model and the level of cooperation required to successfully deliver the model will depend on its scale and complexity. In the following sections these roles are described from an individual perspective.

However, it is possible to deploy a team to build or review a model and our comments apply equally whether an individual or a team is carrying out the work. In practice, using a team can bring additional risks and challenges. Managing the interfaces and overlaps between multiple developers or reviewers requires care and sound project management and may negate some of the time savings from parallel working. Clearly in some circumstances, the deployment of a large team is an absolute necessity due to the sheer volume or complexity of the subject matter transaction (eg, modelling rail franchise tenders in the UK).

3.3 Sponsor

3.3.1 Role and responsibilities

The sponsor[1] is the party responsible for initiating the modelling work, and is usually a member of senior management or a lead transactor. The sponsor's interest and responsibilities in relation to the model typically comprise the following:

- **providing the specification or brief.** The sponsor determines what it is that is to be modelled, providing guidance on the form of outputs required, the key metrics, and the core planning assumptions. These elements form the basis of the 'specification' or scope of work given to the developer.
- **review and sense checking.** While the sponsor may delegate certain review and testing tasks, ultimate responsibility for the model's 'fitness for purpose' remains with them. Through assessing review requirements, directing timelines and resource availability, sponsors have a major influence over the review process. They will usually be close to the given transaction subject matter, and can typically also play an important role in providing quality assurance and analytical review of the model's outputs. A sponsor's intuitive understanding of a transaction, properly applied in analytically reviewing the outputs from a model, is one of the most effective tools for identifying errors or input discrepancies.
- **use of the model, or the results of the model, in decision making.** The sponsor will have commissioned the model for a purpose and will use the model to inform their decision making in that given context. The sponsor will usually direct the model's continuing development as a deal or transaction evolves, and typically will drive the analysis process (for example, through iterative requirements for the development of further scenarios and sensitivities).

3.3.2 Key considerations for sponsors

A sponsor should focus on specification, skills, timing, and review.

Specification

The sponsor is typically the driving force behind the initiation and commissioning of a financial model. As such they should be closely involved in the development of the model specification; in an ideal world they should review and sign it off before the model is built. In particular, they should consider the 'big decisions' (eg, timeline, the entity level at which a business is modelled, required level of accuracy, and the shape of the output required). As far as is possible, they should anticipate the likely direction of development for the deal and advise the modeller accordingly so that the model can be designed so as to most readily accommodate such changes.

1 The term 'sponsor' in this context is distinct from the use of the term in relation to, for example, financial sponsors (ie, private equity owners) and market sponsors (eg, nominated advisers).

Where a specification has limited detail, which can be common in a deal setting, then the sponsor will typically need to be more hands on in the project management and review process during the development phase. This recognises the iterative nature of model development generally, and in particular where development and presentation of (potentially a series) of 'straw-man' model versions is used as a surrogate to test and drive out the required outcome.

Skills deployed

As the sponsor will be relying on the model to support a transaction decision process, so they need to be confident that the model is fit for purpose. This is a function of two things: having it built by people with the right skill set; and having it properly reviewed by an experienced model reviewer.

It is important for the sponsor to take stock honestly of the quality and competence of the skill sets available to a transaction team before commencing the modelling, and to consider whether external help/advisers are required. For example, if the business has a small finance team with only occasional need for financial models, it is unlikely to have invested in the skills required or to be up to date with current best practice. In such circumstances, using external experts can be much more efficient – not least because it mitigates the potential drain on day-to-day management of the business that is attendant in many transactions.

Also, in a small team situation or where the transaction is particularly commercially sensitive, it may be that there are insufficient staff to allow an appropriate division of duties for the modelling process. External help can reduce the risks of 'flying solo'.

Timing

The sponsor will typically have overall responsibility for the transaction timeline and they need to allow a realistic and achievable timeframe for the development of the model. More importantly, the timeline should provide adequate time to review and test the model. This is important both in respect of the sponsor's role in sense-checking the model and getting comfortable that it reflects the design brief, and in relation to the wider process, allowing sufficient time for a thorough independent review.

Review

There is a high risk that any financial model on which a sponsor is relying heavily to support a decision may contain flaws or latent errors. The sponsor has a responsibility to satisfy themselves that the model is fundamentally accurate, whether through personal inspection, reliance on an independent review or more usually both. Done properly, the review process can be time-consuming, as models typically have thousands of formulae. Therefore the sponsor should consider their assurance needs early in the process and plan and resource appropriately.

3.4 Developer

3.4.1 Role and responsibilities

The developer is the author of the model, building the model to the direction and specification given by the sponsor. It is also usual for the modeller to lead on drafting the specification and other documentation (see sections 4.3 and 4.8), to interface with sponsors and data owners to obtain the information with which to populate the model, to lead on implementing version control protocols, and to liaise with model reviewers to assess and, where needed, implement any review findings. The distinction between a model's development and its population with data is an important one, they are discrete tasks. Frequently the latter is best done by others within the organisation, who are closer to the business and familiar with the available management information. We discuss this below under section 3.6.

Generally the developer should be an experienced financial modeller with the requisite skills and training in the platform being used (whether Excel, Winforecast, or another tool). This is a pivotal role and we have often seen clients push it down to junior team members without adequate training or support.

3.4.2 Key considerations for developers

The developer (or modeller) should ensure that the brief is comprehensive, understood, and unambiguous, and that the technical demands of a particular assignment are within his or her competency.

Capturing the brief

The modeller will be responsible for constructing the financial model for a deal in accordance with the needs and intentions of the sponsor and other users. It is therefore critical that they properly understand the brief. The modeller needs to ensure adequate access to the sponsor and other data owners, to ensure they have a clear overview of the project background, sponsor requirements, and available data.

The modeller will typically be responsible for preparing the specification, and needs to do so in a way that is clear and unambiguous. Skimping on this stage can result in misunderstandings or misinterpretation, which can be damaging, waste precious time, and/or be costly to fix.

It is important to emphasise that in capturing the specification, the developer role is not a passive one. The modeller has a responsibility to ask the questions necessary to clarify the brief, to anticipate and explore likely development needs and, where necessary, to challenge the sponsor's assumptions, expectations and preconceptions.

Skill set

It is important that model developers have the right level of skills for a particular project. As highlighted earlier: financial modelling is not the same as Excel literacy. Delegating the modelling to a junior analyst who happens to be good at pivot tables is unlikely to result in a coherent or user-friendly model. Development by those with weak or under-developed skill sets tends to result in models that are inflexible to adapt to evolving transaction requirements. Conversely, the mark of a really good modeller is the ability to anticipate the likely direction of development and user requirements, and to address those needs pre-emptively.

Training in financial modelling is readily available and should be viewed as a pre-requisite; it also needs to be underpinned with:

- Subject matter knowledge (commercial, financial, funding, accounting and/or taxation): ie, understanding the problem to be coded. In our experience, most models are concerned with reflecting the financial consequences of business decisions and the results are typically presented in formats consistent with statutory or management reporting information. They also have to accord with accounting requirements and tax legislation so a formal qualification in accountancy is often an excellent foundation for financial modelling. From a transactional perspective, an understanding of basic financial instruments, investment accounting rules, and financial performance metrics is also key.
- Excel skills: important simply because Excel is generally the programming environment (or language) used by modellers to code problems; and
- General IT literacy/programming skills: these bring the mindset, discipline and skill to analyse a desired outcome, and to develop and code the logical steps (or building blocks) required to generate that outcome.

The three strands are complementary and interrelated. They are not sufficient in themselves – experience and training in applying these together in a modelling context is critical.

3.5 Reviewer

3.5.1 Role and responsibilities

The reviewer is typically independent of the model development process and engaged to rigorously scrutinise, test and review the model. Model review can take various forms but at its most basic it assesses whether the model is 'fit for purpose'. The reviewer role can be undertaken by an individual or by a team, depending on the size and complexity of a model, time constraints, and separability of review tasks.

ICAEW Principle 18
Rigorously test the workbook.

The reviewer role can include checking some or all of the following points, according to the level of assurance required by the sponsor:

- that it reflects the specification or brief;
- that the results generated are consistent with the expected outcome for a given set of inputs (and continue to behave consistently under stress testing);
- whether it has been coded properly and robustly (from a technical perspective);
- that the model and any associated data book/documentation/inputs are consistent;
- that the model is free from common and/or material errors; and/or
- that the model is compliant with best practice.

Note that it is a generally accepted axiom of software development that it is not possible to attain 100% confidence that a piece of software is free from material error. Financial modelling is a form of software development (in an end user environment), and that principle applies equally. No amount of testing can guarantee that a model is free from error, but review and testing can reduce the risk, and the more that they are performed, the lower the risk. Ultimately, however, it is for the sponsor to determine the level of risk acceptable to them, and by extension the degree of assurance required.

In our experience, an appropriate independent review of a given model is the task (and therefore the role) which is most often missing from the model delivery process in deal environments.

We consider the different review options and approaches more fully in chapter 16.

A war story

In 2010 an academic paper by professors at Harvard University reached a controversial conclusion on the effect of public debt on economic growth. The paper was later found to be based on coding errors and debatable calculations.

The model referenced an incomplete cell range in calculating average growth rates, which had the result of reducing the result and formed part of the results they published.

This demonstrates the benefits of segregation of roles; risk of the problem could have been mitigated by asking a third party to review the model's integrity.

Source: EuSprIG

3.5.2 Key considerations for reviewers

When initiating a model review process, there are a number of factors to consider, in particular: independence, subject matter knowledge, timing, and access to information.

Independence

As a general rule, the reviewer should be independent of both the sponsor and the model developer, and this is even more important for larger, more complex, models. This is not to say that a third party should always be used[2], rather that the reviewer must come to the review with fresh eyes and free from potential 'contamination' through exposure to, or involvement in, the design or construction of the model. In practice, where resources are constrained, the reviewer role can be performed by the sponsor. At least this level of division of responsibilities (ie, a minimum of two people carrying out the three roles) should be maintained. Having a single person cover all three roles greatly increases the risk of errors, and of bottlenecks in the process.

Subject matter knowledge

Reflecting the sometimes technical nature of reviewing complex spreadsheet models, the reviewer ought generally to be an experienced modeller in their own right,

2 For example, where a model has been developed by a given organisation, that does not necessarily preclude a separate team from the same organisation from carrying out the review work, provided: the relationships are transparent to, and understood by, the sponsor; there is no involvement in the review from any of the development team; and all review comments and findings are channelled through the sponsor.

preferably with specific training and experience in reviewing financial models. The general approaches to review can be separated between analytical and technical. For the analytical approach the required skill is simply an understanding of the subject matter. Indeed, an intuitive understanding of a business and the financial and commercial indicators and results being modelled is often the most effective means of identifying major modelling errors. In turn a technical review (for example a formal model audit) requires knowledge of specialist tools and techniques and a high degree of modelling skill and experience. The specific skills required will naturally depend on the nature, scale and complexity of the project and models developed.

Timing

Model evolution right up to deal closure is common and so review work often has to start before the model has stopped moving. Deciding when to commence the review is a finely balanced judgement. The review process should generally only start once the model is structurally complete and stable. Unfortunately, real life does not always allow for that, with significant, late structural alterations not uncommon in a deal setting. The risk is that starting the review process too soon can result in repeated or duplicated efforts, start too late and the review process may put pressure on deal closure timings.

Modern spreadsheet review tools allow the review effort to be time-bounded reasonably accurately and can give an idea of the latest realistic start date. If the review work has to take place in parallel with ongoing development, it is important to ensure that a rigorous change log and version control protocol is in place, so that review findings can more readily be interpreted into subsequent iterations of the evolving model.

Access to information

For the reviewer to do their job properly they need adequate access to the model developer to discuss and resolve queries; they also need a full suite of project or model documentation. This helps put the model properly in context and provides the backdrop to any analytical sense-checking. A clear specification gives the reviewer an objective, static picture to test whether the model is meeting the sponsor's intended purpose.

3.6 Other roles/interested parties

There are other interested parties in a modelling process.

- **Internal audience/stakeholders.** A financial model will usually have a wider audience than just its sponsor. This may be indirect, with the analysis being used to support board investment papers and shareholder briefings, or direct, with other third parties (such as lenders or rating agencies) using or reviewing output from the model. The sponsor and modeller will need an awareness of this wider context in order to ensure that the model addresses their analytical needs.
- **Data owners/business users.** The modeller will often act as a conduit or filter between the underlying business data and the model itself. The role typically involves the developer liaising with the data owners to obtain information from the business that is relevant to the model and required to populate it. Typically there will be other people within the business better placed to source and provide the financial information and assumptions required for the model. The developer's task is then to interpret, manage and process that information within the model. The developer also has to ensure that the data is kept up to date, and is synchronised across the data owners, because developments in one area of a business will often have repercussions elsewhere. Acting as a go-between, the developer is ideally placed to intercept, and to help address inconsistencies in data sets and thinking.
- **Third-party stakeholders.** The review process may be undertaken or augmented by third parties. For example, it is increasingly common for lenders to seek an independent

review of spreadsheet cash flow forecasts against which they are lending – indeed, in the project finance arena it is the market norm. This can create interesting dynamics as interests may not be wholly aligned between equity investors and lenders, and the model may contain cases and/or analysis that the sponsor would not want shared with certain third parties or their advisers. This places a premium on the need for clear communication of confidentiality issues between the sponsor, the data owners, and the developer, as well as the need for tightly managed version control.

3.7 Key points

- Split out responsibility for the different modelling activities, with a clear and shared understanding of respective roles.
- Involve the model sponsor and end-users fully in the design and testing/review of the model.
- Ensure that the developer has the requisite skills, experience and capacity, remembering that modelling is not the same as Excel literacy.
- Ensure that the model is independently reviewed and do not rely on the modeller to also carry out quality assurance on their own work.
- Give due consideration to the wider audience when planning, delivering and using the model.

4 Process

4.1 Overview

This chapter provides an overview of the process (or development lifecycle) of a financial model, considering:

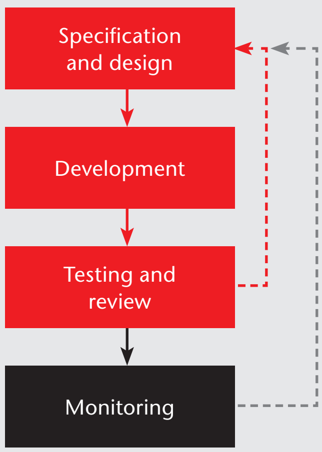
- the three core phases of development:
  - design and specification;
  - development; and
  - testing and review;
- the use of models as a post-transaction monitoring tool;
- project management and planning; and
- the types of documentation that should be prepared to accompany a model.

4.2 The financial modelling lifecycle

The process of developing a financial model is similar to any other development process: plan, execute, check and use. In a financial modelling context, the process can be described as set out in Figure 5. Each step in the process is considered below. Part B provides more detailed applied guidance.

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

Figure 5: The modelling process



4.3 Specification and design

The starting point should be the development of a model specification, a document that defines the problem and sets out the proposed approach. An example is given in Appendix D. The specification typically includes transactional background, available data and information (and constraints thereon), key calculations, and approaches to addressing specific modelling issues.

In particular, the model developer will seek to map the main interrelationships between business activities and their financial consequences, to ensure that these can be modelled dynamically. They may also include example schedules showing the design of the output format and details of the proposed software to be used (see also section 2.4). Time permitting, this phase may also allow for the development of a 'straw man' or the prototyping of certain complex areas of a model.

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

It is also important to note that the specification should be seen as a jumping off point not as a straitjacket. In all modelling processes, there is a significant element of iteration. While a model is in development, new constraints arise, unforeseen issues materialise, and sponsor requirements evolve in response both to changing deal parameters and to results and analytical possibilities emerging from the model.

In a deal context, this step is often curtailed owing to timing pressures. We recognise that it is not always practicable to invest significant time in the design and specification, but as a general rule we have found investment in getting the fundamentals right early is rarely wasted.

From a more general design perspective, different financial models will often contain common constituent parts, such as version control sheets, change logs, and model maps. It will therefore be beneficial to adopt a standardised approach to creating and maintaining such components, particularly if your organisation has more than one developer. Doing so will enhance transparency, improve model assurance, provide efficiencies and, aside from all else, create a consistent presentation style and improve stakeholder confidence.

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

4.4 Development

Once the desired shape of the model has been determined and designed in outline, the next task is to turn the specification into a live model. This phase is the coding of the specification into a dynamic spreadsheet model. During this phase the modeller will:

- collate and organise the source data/information;
- set up the workbook and worksheet template layouts;
- design and implement the model output formats;
- develop, programme and test the individual calculations, remaining mindful of the eventual outputs; and
- document the model (eg, user guide).

ICAEW Principle 9
Focus on the required outputs.

It is not uncommon for a finished deal model to have hundreds or even thousands of data input points and thousands of individual formulae. This brings an inherent level of complexity and the design should be carefully considered to enable use by third parties, by providing sufficient instructions for use, and by placing validation checks on inputs to mitigate the risk of error. In addition, as noted above, it is often the case that the implementation of the intended specification will need to be flexed to accommodate the actual (versus the expected or desired) information available, as well as adapting to changing deal circumstances.

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

For these reasons, the build or development phase needs to be approached in a disciplined and structured manner. (Part B explores some applied approaches and techniques and chapter 9 sets out a recommended sequence for building up the calculations.)

4.5 Testing and review

Testing and review are considered here as distinct but related activities:

- testing is aimed at checking that a working or calculation does what it is expected to do when assumptions and/or data change (tends to be carried out by the modeller or sponsor);
- review is aimed at providing holistic assurance that a model and its workings calculate logically, consistently, and accurately (tends to be done by the reviewer and/or sponsor).

4.5.1 Testing

Once model coding has been written it should be tested. In practice it is often better to ‘test as you go’: ie, write a section of code, then test it before moving on to the next section.

‘Test as you go’ is powerful because it tends to concentrate on bite-size elements of the model’s logic and reduces the risk of testing results becoming opaque due to ‘noise’ from other factors. In addition, by building in checks during the design process, with clear flags on each page, a number of bugs can be ironed out during the development phase.

While most good modellers adopt a ‘test as you go’ approach, formal user acceptance testing seems relatively rare. In an ideal world, all models would each have a detailed specification drawn up, which would be tested against and signed off by the sponsors, with the level of testing guided by the complexity of the model. In a deal context, this step is often missed out and can be impractical, with reliance instead placed on an ‘external’ review process.

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

Hint

Simple techniques include: testing each leg of any IF or cross-reference formulae (MATCH, CHOOSE) to ensure they work as expected, testing for error conditions/error traps, testing with live and dummy data and testing with zero values or extreme values

ICAEW Principle 18
Rigorously test the workbook.

4.5.2 Review

After the model has been constructed, it should be subject to a comprehensive, independent review. Comprehensive because you can’t risk missing an error by testing on a sample basis; independent because it is extremely difficult for a modeller to self-review 100% effectively. This is why we advocate the separation of roles.

Time should be built into the transaction due diligence timeline for this process, whether the review is carried out internally or by external advisers. As with testing, there can be some merit to starting the review process earlier, for example seeking third-party advice and input on the specification. However, in most cases a full technical review should be carried out as late as is realistically possible without compromising deal deadlines. The reason for this is that deals often evolve; and even with sophisticated review and comparison software it is quite likely that late modelling changes could result in significant duplicated review effort and costs.

As Figure 5 above illustrates, review findings should be incorporated into the development process, typically starting with revisiting the specification. The review feedback will typically specify the problems or issues found, these can then be recoded or rebuilt and then tested, before the amended model is reconsidered by the reviewer or review team. Note that this process can become cumbersome if version control is loose or the review process not properly project managed.

We recommend a discrete review stage commencing once the model is substantively structurally complete. Ideally, the base model should effectively be ‘frozen’ while it is being reviewed, with any other desired changes being held over and implemented only once any corrections arising from the review have been processed and checked. We recognise that deal timelines often militate against a clearly demarcated review exercise. Where review work is carried out in parallel with development, we recommend that the modeller maintains a detailed log of changes. In particular, and as a minimum, listing any insertions or deletions of rows or columns. Without this, the process of review, particularly a structured external review, becomes markedly more difficult, with greater risk that issues may be missed.

In a formal review or model audit process (see chapter 16), typically at least two to three review iterations are required. One initial review; one re-review to check that any review findings have been addressed; and in some situations, a final review at the decision gateway (eg, financial close or an investment approval committee), acknowledging that the model and its inputs may be subject to late changes. This applies equally to a less structured or formal review process, which should repeat or revisit aspects of the review as the deal and model evolve.

A word of caution: if review findings are presented and allowed to be addressed in a piecemeal manner, then the reviewer or review team need to keep very careful track of the model versions received for review and of the extent of work undertaken on each. The risk otherwise is that the model development and review processes can become dangerously out of step, with an increased risk that any errors could slip through the gaps. Where timelines permit, we would advocate avoiding staging the sharing of review findings, but there can be timetable benefits to a tranched approach.

Chapter 16 considers the issue of model review and assurance further.

4.6 Monitoring

Post deal, the financial model may continue to be used to evaluate the performance of the transaction or investment. This creates a further feedback loop: ongoing use may highlight issues or latent errors in the model that require correction. It may also highlight future development needs either because the model is to be used in a further deal or the business' needs have evolved.

One general note of caution in relation to using deal models to monitor transactions post deal, is that deal models are rarely well suited to address ongoing needs, because they are typically aimed at answering a specific question at a specific point in time. While a well-constructed model can anticipate or be re-engineered to be used in a monitoring role, it will then contain a lot of redundant functionality and likely be cumbersome in use. Accordingly, in many situations it is better to start over, with a simplified forecasting tool informed by the deal model and integrated with the business' management information systems.

4.7 Planning considerations

Transactional timelines frequently put pressure on one or all of the steps above. We recommend that the different elements are carefully considered when constructing the transaction project plans and timeline.

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

ICAEW Principle 8
Design for longevity.

In the first instance, spending significant time upfront to get the planning and model design right (or as right as possible) generally saves time later in the deal process through upfront investment in understanding, documenting, and anticipating the likely uses and developments of a model.

Secondly, a formal model review gateway (eg, a third-party review or sponsor sign-off) should be built into the process. Too often this step is neglected or an afterthought and in many cases real money has been lost as a result of incorrect models that have either not been reviewed or have not been adequately reviewed.

4.8 Documentation

In a deal context, time pressures mean that meaningful documentation for a model appears rarely. Documentation that is not done well or with consistency, or is simply missing or incomplete, is a significant risk area.

The advantage of documenting the model and associated design, specification, and development processes is that it can help to avoid misunderstanding, help to make sure that implicit assumptions are captured, provide an audit trail of decisions and data, help with continuity and user handovers, and assist the user or reviewer in using the model correctly.

There are four core pieces of documentation:

- **specification.** A statement of requirements for the model to be built (see example in the Appendix D).
- **databook.** A clear statement of all the assumptions and inputs for the model. In practice, a well-designed set of input worksheets can fulfil this function, with the added

benefit that being integrated into the model, they remain more readily up to date and in step with evolving assumptions.

- **change log.** A listing of version history, with development and input changes, with impact, dates, and attribution. This can assist in tracking and tracing when errors arose, and provides a ready audit trail and backup directory. Elements of this process can be automated, though Excel does not natively have robust track changes functionality in the way that, say, Microsoft Word does.
- **user guide.** A document that sets out how to use the model, explains its purpose, structure, layout and formats, describes the major calculations, and contains other information essential to an understanding of the model. Often they include, or overlap with the databook, which also sets out key assumptions. As with the databook, elements of the user guide may lend themselves to incorporation within the model itself, for example via an introduction sheet within the model.

The first two documents are the most important: a clear statement of requirements, and an ordered, transparent listing of inputs and assumptions.

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

ICAEW Principle 7
Include an 'About' or 'Welcome' sheet to document the spreadsheet.

4.9 Key points

- A modelling exercise benefits from a structured approach: design → build → test → review.
- Invest time upfront in the specification and model design to avoid modelling decisions that are difficult to reverse.
- Implement rigorous version control from the start of development, and build the model according to best-practice principles.
- Test as you go and build time into the modelling process for independent review after the development phase.
- Deal models are often cumbersome or unsuited to use post transaction. If a model is expected to have an ongoing role post deal, plan for this early in the development at the scope and specification stage.
- Make sure the modelling process and transaction timeline also provide for adequate time to plan the work and review the work.
- Document the model (in particular the specification and data sources) so that the sponsor can better assess whether the finished product meets the required needs.

5 Implementation

5.1 Overview

This chapter provides a brief overview of the key tenets (or principles) of best practice:

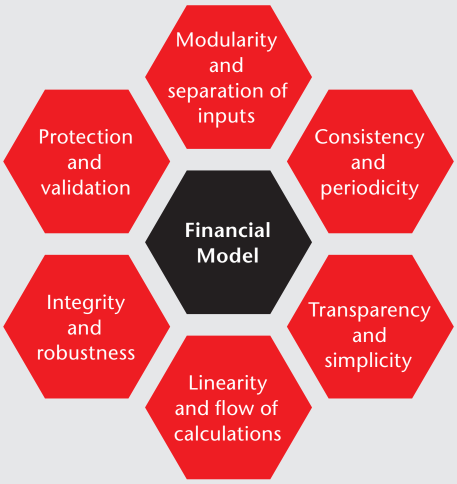
- modularity and separation of inputs;
- consistency and periodicity;
- transparency and simplicity;
- linearity and flow of calculations;
- integrity and robustness;
- protection and validation.

While descriptions and labelling may vary, we believe that most professional modellers can agree on these core principles.

5.2 Best practice principles

There is an emerging consensus on core principles to be observed when building a financial model.

Figure 6: Best practice principles



Taken to extremis, applying some of these techniques may appear counter-intuitive or time-consuming; however, any incremental investment of time incurred should reduce the risk of subsequent error and is likely to be recouped later in the transaction process through more efficient implementation of changes in the modelling requirement.

Observing the principles in Figure 6 will make the model review process more straightforward, and a review-friendly model is also typically a user-friendly model.

5.3 Modularity and separation of inputs

The first principle is to keep inputs separate from calculations and calculations separate from outputs, ideally on separate worksheets. The developer should also ensure each input is only input once, in a dedicated input cell, and therefore easily updated should it need to be changed.

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

Inputs should not be hardcoded within formulae, as this obscures them from view and hampers the process of updating the model should the input assumptions ever need to change.

These disciplines make incorporating new workings more straightforward, make the functional purpose of different parts of a model more obvious to a user, reduce the risks of duplicate inputs, and avoid the legacy issues and risks that arise when inputs are mixed with workings (such as occurs, for example, when a user overtypes a formula with a calculated figure).

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

5.4 Consistency and periodicity

The second principle to observe is that of consistency:

- consistency of timeline and worksheet layout across worksheets (ie, any given column on any given sheet should represent the same time period in the modelled timeline);
- a consistent period duration along the timeline (periodicity);
- consistency of formulae blocks in a row or column (from left to right or top to bottom);
- consistency of approach to like calculations, to formatting, and to descriptions; and
- an internally consistent data set.

This approach makes updates easier and reduces the review burden for anyone looking at a model's logic.

ICAEW Principle 11
Be consistent in structure.

ICAEW Principle 12
Be consistent in the use of formulae.

5.5 Transparency and simplicity

The third principle is to make the model transparent and simple to follow. The design should facilitate this by using clear descriptive labels for calculations, using bite-size formulae, and using formatting styles to assist the audience in understanding the purpose and function of individual cells. The hallmark of transparency should be that the user is able to work out the function of a cell or group of cells using only a print out of the model, and is also able to replicate individual cell results using little more than a pencil, ruler, and calculator.

ICAEW Principle 13
Keep formulae as short and simple as practicable.

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

In practice, this means breaking each problem or calculation task into individual and transparent steps. One process or task per formula is to be preferred over trying to solve a series of computational tasks within a single formula. One of the modelling boutiques coined the phrase 'rule of thumb', ie, that no formula in Excel's formula bar should be longer than your thumb. Tongue in cheek perhaps, but it accurately encapsulates the principle. We can't emphasise strongly enough how important this point is: it enhances readability, reviewability, and usability of models, not to mention helping to ensure that updates and changes can be made more easily.

Transparency extends to avoiding hiding model content (whether rows, columns, or tabs). Hidden content gives rise to two issues:

- Hiding material goes directly against the principle of transparency and begs the question why does it need to be hidden? If secrecy is the goal because given information is sensitive, spreadsheets are not a secure environment and the practice is ill advised. For sensitive information important to the flow of calculations (eg, salary data), it would be better to perform the relevant calculation outside the model and bring in the results as inputs.
- Hiding material introduces additional modelling risk: information in hidden rows and columns is easily accidentally deleted or corrupted.

As a result it is rarely, if ever, a good idea to hide components of a workbook, with the possible exception of dialogue sheets, though these are rarely needed in a deal context.

The other aspect of simplicity relates to ease of use. If the model is appropriately dynamic, with inputs separated for calculations, and interrelationships modelled and captured within the formulae, then it should be simple for the user to run sensitivities and adjust the model by changing only a few clearly marked inputs. Highly static models, with a large number of 'coded' inputs, and only limited modelling of interrelationships, will likely involve multiple detailed input changes and require an element of thought or calculation outside the model to implement. This brings risk, reduces transparency, and may not be straightforward in use.

At its simplest, the principle being that a model's outputs respond dynamically to simple input changes without the need to:

- adjust multiple related inputs; or
- calculate inputs 'off line'.

A transparent and simple model is easier to use, easier to update and, if the worst happened and the files were lost or corrupted, easier to reproduce.

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

Illustration

A good example of dynamism can be illustrated by working capital. Debtor and creditor balances and the cash impact of movements in working capital should be dynamically linked to sales and costs, and calculated live in the model by reference to, for example, debtor days inputs. They should not simply be hard-coded forecast figures. Similarly, VAT payments and receipts should be linked to a VAT rate input and calculated simply in the model rather than being input as nominal figures, allowing updates to VAT rates to be made in one designated input field, rather than in each calculation. Another example would be to calculate costs as a proportion of sales or other ratio such that changes in a given input, in this case sales, would still return a valid model result.

5.6 Linearity and flow of calculations

The fourth principle relates to the flow of logic and data through a model. Generally, a model should be laid out to be read like a book, ie, from left to right and top to bottom. This principle should be applied rigorously within worksheets, recognising that occasionally a calculation will have to reference subsequent working blocks. In these cases it can be helpful to label such contra-flow calculations. At a workbook level, the principle also stands, though like a book or report, it can be helpful to include an overview at the beginning. It is common for deal models to include the dashboard and summary outputs as the initial tabs that the user sees.

One type of contra-flow calculation is to be avoided in all circumstances: the circular reference. A circular reference occurs where a cell references a calculation which directly or indirectly references itself. They are avoidable in almost all cases through intelligent, thoughtful coding, and where they are unavoidable without degrading a model's accuracy, then they should be explicitly and manually controlled and managed. Circular references introduce instability and mask other (possibly unintentional) circularities as discussed in chapter 8.

Models should also be designed to avoid duplicated calculations. Duplicated calculations detract from a model's linearity and increase the potential for error. For example, a modeller may update the logic of the calculation in one point in the model but not the other, resulting in calculations becoming out of sync. Furthermore, this can also make investigating bugs more troublesome and lead to assumptions that errors have been fixed, when in fact only half of the issue has been resolved.

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

5.7 Integrity and robustness

The fifth principle is that all financial models should include a full set of integrated financial statements with appropriate error and balance checks. Even for pure cash flow models, the discipline of including the income statement account and balance sheet creates a self-checking mechanism that should trap or identify errors introduced during the development process. Depending on presentational requirements, proforma financial statements will often suffice – the key point being that they are fully integrated and independently calculated (ie, no 'plug' figures).

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

Building these checks and flags in from the start provides reassurance that the model has been built correctly throughout the development process and should therefore be easier to debug. It also minimises the time required to find and resolve modelling errors, because they are identified earlier, and it is more likely that a given error condition is the result of a single error in isolation, as opposed to becoming compounded with other errors, which becomes more difficult to unpick.

It is vital to include error checks throughout the model to re-perform sums, to check input validity, and to check at a global level that the relationship between inputs and outputs remains sound.

5.8 Protection and validation

The sixth principle is one of error prevention. Excel's security features can mitigate the risk of accidental amendment of code and its validation features can help to restrict the values that users can input into input cells. This reduces the risk of a user 'breaking' the model by using invalid or unplanned inputs, or by making amendments to existing formulae.

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

5.9 Rules are made to be broken...

It is important that the principles are understood and that any deliberate breach of them is done in a controlled manner and with a full appreciation of the risks.

There are certain applications that do not fit within the principles set out above, though in a corporate finance context these are rare. For example, securitisation models have traditionally been laid out with a vertical rather than horizontal orientation; though this was driven largely by Excel's original 'space' limitations, and the rationale for this layout has receded since the launch of Excel 2007. Similarly some problems are inherently circular, such as certain debt sculpting, interest, tax and dividend calculations (though not always).

Another justifiable departure from best practice would be the left to right and top to bottom layout of a workbook. For example, if a model incorporates a worksheet per division or company, with integrated financial statements developed at that level, then it may be helpful the show those financial statements above the workings on these pages. This approach will result in an upwards dataflow on these sheets, where the workings below feed to the financial statements above; the justification is that consistent placement of the financial statements will enable consolidation by 'adding through' the worksheets using bookends. By placing the workings below the financial statements any changes to the workings (eg, inserting/removing rows) will not impact the alignment of the financial statements versus the other sheets in the group.

Separation of inputs, calculations and outputs is another area where deviations are sometimes acceptable. For example, an input sheet may include a link back to an output cell, in order that workflow is made easier for a model user, ie, the user can immediately see the impact of changes they are making to input assumptions. Also, a dashboard output sheet may include areas for inputs, again to improve ease of use of the model. Note that in both of these circumstances the underlying inputs and outputs should still be separated and all cells calling up data from input sheets ('call-ups') clearly identified or highlighted.

5.10 Key points

- Separate the model into distinct sign-posted components, and keep all inputs separated from the model's workings.
- Make sure the model is internally consistent across and between worksheets, particularly the timeline.
- Keep formulae brief: better to have lots of short formulae than a few 'jumbo' formulae.
- Avoid circular logic.
- Build in checks, and an integrated cash flow statement, income statement, and balance sheet.
- Use protection and validation to prevent users 'breaking' the model.

PART B:
Applied best practice financial modelling

6 Introduction to applied best practice

6.1 Overview

This part of the guideline provides some practical guidance on applied best practice financial modelling relevant to a transaction or deal context. The focus of this section is on Excel, as the most common modelling platform, although the principles (consistency, clarity, etc,) apply equally when modelling on other platforms. We have focused on issues that in our view, are:

(a) most relevant/useful in a fast moving deal context (that is to say they ultimately save time, reduce risk, and aid clarity/review); or

(b) are so fundamental to good practice that to neglect them would be a dereliction of duty.

Note that both aspects are thoroughly intertwined.

6.2 Content and audience

The table below indicates which chapters are likely to be of most relevance to the different parties to the model process.

Topic	Description	Sponsor	Developer	Reviewer
7 Model specification and design	Commentary on major structural and design issues for the modeller to consider at the planning and specification stage, including some illustrative approaches and recommendations.	★ ★	★ ★ ★	★
8 Applying the best-practice principles	Advice and practical guidance on interpreting and applying the best-practice principles.	★	★ ★ ★	★
9 Model structure and flow	Practical tips on model structure and layout, in particular the classification and layout of worksheets and the sequence for layering up the main calculation blocks.	★	★ ★ ★	★
10 Useful functions and techniques	An exploration of the more common useful functions and techniques that may be used within Excel to create financial models.		★ ★ ★	
11 Communication and presentation	Tools for enhancing model communication and presentation, including data table and chart use, dashboard and formatting.	★ ★	★ ★ ★	
12 Dashboards	Considers the purpose, usage and construction of dashboards within financial models.	★ ★ ★	★ ★ ★	★ ★
13 Model management	Techniques to enhance risk management, use and control of a model, with particular regards to version and scenario control.	★ ★	★ ★ ★	★
14 Productivity and efficiency	Covers techniques to speed up model development and reduce model risk. Provides advice on 'dos and don'ts', keyboard shortcuts and the use of flags, masks and counters.		★ ★ ★	
15 Model speed	Practical tips on choice and structure of formulae and other ways to save time when building or using models.		★ ★ ★	
16 Model assurance	Overview of modelling risks and advice on model review, covering: role and timing, and the approaches and techniques that can be used to carry out quality assurance on financial models.	★ ★	★	★ ★ ★

7 Model specification and design

7.1 Overview

This chapter deals with the following practical matters that should be considered at the specification and design stage:

- timeline;
- entity level;
- sign convention;
- layout;
- accuracy; and
- range names.

7.2 Context

In constructing a model there are some aspects that once set are difficult to reverse because to do so is time-consuming, may be complex, and will certainly introduce execution risk, particularly in the context of a deal.

These aspects are best considered and addressed early in the overall model development lifecycle.

7.3 Timeline

Most financial models tend to forecast a business' performance over time, and many of the calculations in a typical financial model are time-dependent. For example, loan repayment schedules, transaction execution and exit timings, and the application of inflation and growth assumptions.

Accordingly, getting the timeline and time-based calculations right is of critical importance.

7.3.1 Timeline calculations

From a planning perspective, the unit of time used (weekly, monthly, quarterly, annually, etc,) depends on the quality and granularity of data available and the purpose of the model. For example a short-term cash flow forecast prepared for banks by a business in financial distress might forecast in weekly time units, whereas, a high-level strategic planning tool or valuation appraisal model based on filed accounts might focus on annual periods.

There are two particular challenges with regards to the timeline:

- Firstly, once set, it can be very onerous to change due to the need to convert data values, the reconciliation of data sets, and the adjustment of the calculation bases (for example, discounting interest rates).
- Secondly, sponsors and users often want to see different time period lengths for different project phases. For example, lenders might want monthly cash flows for the first 12–24 months after financial close, and be satisfied with annual or semi-annual figures thereafter. Furthermore, the stub period has long been a source of complexity in deal models for a management buyout (MBO) or leveraged buyout (LBO) transaction.

In respect of the second point, variable period lengths can be dealt with using convoluted calculations although they add complexity and can make the model workings harder to understand. Variable period lengths can also be accommodated by changing the formulae part way along the timeline, but that breaches the principle of consistency and increases the risk of future errors.

The solution to both problems lies in the planning stage. It is vital to consider and anticipate the deal structure and the likely users of the model, their requirements and how they might evolve. Once the developer has clarity on the users' needs, we recommend building the calculation engine using a timeline with the lowest likely common denominator, and with a consistent structure on each worksheet where a time series is used, such that future requirements to calculate or report at a detailed level can be easily accommodated. In our loan example above, we would suggest building a monthly model for the full forecast period and then aggregating the monthly cash flows for later periods into annual or semi-annual flows solely for output purposes. A similar principle applies when dealing with the stub period. A technique for summarising timeline calculations is illustrated in section 8.3.2.

7.3.2 Time-based calculations

Time-based calculations are calculations that test for a time-based condition. They can be split into two further sub-categories:

- primary time-based calculations – calculations which are pure logic tests against date parameters, which are then referred to elsewhere in the model, such as period counters, masks, and flags; or
- secondary time-based calculations, where a financial calculation includes reference to a date-based test or a primary time-based calculation.

As we show in chapter 8 there is real merit in splitting out frequently used time-based calculations in terms of clarity and efficiency, particularly where they are widely used, such as in project finance transactions. In other words, break out frequently used logic tests that check against dates or timing, and model them in an area set aside at the top of the relevant worksheet. These primary time-based calculations can then be referenced by other calculations as needed. Examples might include indicators showing whether the modelled period is in a historical or forecast time period, whether the current period contains the deal date, or whether a loan payment is due in the current period.

The advantages of breaking them out separately are transparency and efficiency, as otherwise the timeline and related calculations can become overly complex. To re-emphasise an earlier point, we advocate the modeller spending significant time at the planning and early development to get this aspect right.

Once the timeline and primary time-based calculations are programmed, the remaining financial calculations (ie, secondary time-based calculations) can usually be implemented relatively rapidly.

7.4 Entity level

Businesses do not neatly fit into a single set modelling template as the structure and shape of individual businesses varies enormously. When specifying the model, it is therefore important to agree the level of corporate entity or subdivision at which the model will be required to operate. We use the term 'entity' or Strategic Business Unit (SBU) to refer to the most meaningful/measurable building block for the model. Examples are:

- service line: product/service type, customer groupings;
- geography: office, region, country;
- functional: sales, production, finance etc; and
- legal entity; company, division, contract.

SBU classification within the model is typically used to organise, process and manipulate the financial data for operational performance of the relevant entity. It will usually reflect the sponsor's understanding and view of their business, often mirroring management information reporting.

Depending on the granularity and complexity of a business, it may be possible to categorise and analyse operational data under multiple headings (eg, product and geography); if so, the model specification needs to capture these analytical and presentational requirements. Furthermore, the modeller will need to think carefully about how to group and classify data and workings, and the taxonomy required to enable such information to be processed and reported in different dimensions.

As with setting up timelines, it is generally preferable to design with future flexibility in mind and model at the level of the lowest common denominator or the lowest realistic denominator. This will be the lowest level to which users or stakeholders will want to, or be able to drill down. Using the lowest realistic denominator, different views of the underlying entity level results (eg, product category or territory) can be grouped and summarised for presentational purposes as required by the model users. It is much easier to consolidate different entities' results than to split them out into more granularity at a later date.

The decisions on entity structure and level need to balance meaningful management analysis with introducing unmanageable or unnecessary complexity, data overload, or spurious accuracy. In a deal context there may be a natural compromise between available data and the level of detail required by the sponsor.

7.5 Sign convention

Sign convention is the presentation of numeric financial data. For example should costs be presented as positive decimals or negative? While this can be a contentious area, in practice it really should be straightforward: choose a sign convention and apply it consistently.

There are three general approaches:

- **Accounting.** Accounting convention expresses debits as positive and credits as negative. For presentation purposes, the sign attaching to specific values may then need to be adjusted (eg, sales shown as positive rather than negative credit values). This approach is sometimes seen where models use data feeds from accounting systems/software. It can be counter-intuitive for non-accountants.
- **Contextual.** Under this approach, the sign convention relates to the context. So, a loan balance is expressed as positive, with draw-downs also positive, and repayments negative; similarly, a fixed asset would be expressed as a positive balance, with additions positive, and disposals negative. Elsewhere, this has been described as a 'positive is normal approach'.
- **Hybrid.** This method treats inflows and assets as positive and outflows and liabilities as negative, combining elements of both the accounting and contextual approaches.

While the contextual approach is generally more intuitive for other users/reviewers, the key is to apply the chosen convention clearly and consistently. This principle applies equally within the use of the SUM function. A SUM should follow the 'what you see is what you get' principle, adding the figures in the relevant selected range. Use of minus signs within the SUM function to reverse the sign of particular cell values should be avoided as it is unnecessary, rubs against the transparency principle in section 5.5, and is confusing to the model user/reviewer.

Hint

When changing sign, you can use spaces in the formula to signal the change clearly. Eg:

= - (X+Y) or = -1 * (X+Y)

Rather than

=-(X+Y)

7.6 Layout

Layout refers to the orientation of the model timeline and calculations within a model's worksheets.

The developer has a choice to lay out the timeline horizontally (from left to right) or vertically (top to bottom). Historically most cash flow forecasting models have been laid out horizontally, this reflects the way in which analysts, financiers, and accountants tend intuitively to read financial information. In addition, it allows the calculation headings to be tabulated on the left – meaning more text information can typically be fitted clearly on the page.

The most notable exception to the horizontal approach arose from the securitisation arena. The need for the timeline to be modelled at a monthly level for periods typically in excess of 20 years was not tractable within the limitations of older versions of Excel (Excel 2003 and its antecedents only had 255 columns, whereas later versions, from Excel 2007 onwards, permit 1,048,576 rows by 16,384 columns). Accordingly, the data and calculations for long duration financial instruments were typically set out vertically. With the changes to Excel in 2007, the horizontal format is best suited for most corporate finance related applications. If a vertical format is needed for presentational purposes, this can simply be accomplished on a discrete output worksheet, transposing the horizontal workings using (for example) INDEX or SUMIF formulae. However your organisation chooses to lay out models the most important concept is consistency, which will help to reduce the risk of errors and circular references.

Tip

If you are working with a larger dataset, Excel 2016 provides the facility to import larger datasets using its 'get and transform' functionality.

7.7 Model accuracy

It is important that the modeller and sponsor agree and share an understanding of the level of accuracy to which a model will operate. A model can only ever be an approximation of reality and will never be accurate in the purest sense, although it is reasonable to expect a given range of input values and their specified modelled relationships to be calculated accurately.

In practice, the acceptable bounds of accuracy will depend upon:

- **Quality/granularity of available data:** if only broad brush information is available, then accuracy will be constrained commensurately. For example, if on contemplating an acquisition, you do not have detailed divisional data for the target, you may have to model at group level or at the level of legal entity, losing a degree of granularity in forecasting performance.
- **Complexity of interrelationships between business drivers:** it may not always be practicable to model complex interdependent business relationships – either due to time or data constraints. Sometimes such relationships have to be simplified – such decisions being a key output from the specification phase.
- **Extent of inherent estimation risk:** in some situations, there may be significant exogenous risk factors that cannot reliably be estimated (eg, sales take-up on a new product line, contract win rates). In practice, this uncertainty is typically addressed through scenario planning and sensitivity analysis.
- **Required accuracy/materiality threshold:** comments on data/complexity notwithstanding, a model's users may have specific requirements in relation to accuracy, and tolerance of approximations will vary according to the audience and the use of the model. For example, in a highly leveraged project finance structure, small differences in revenues and costs can have a material impact on lender covenants, such as debt service cover ratios – therefore accuracy is at a premium. In a venture capital-backed business start-up there will be greater uncertainty of outcome and significant variations in costs and revenues are to be expected, and will be factored into planning through scenario/ sensitivity analysis.

Accordingly, during the specification/design of the model, it is vital to establish the quality and availability of business data and the acceptable parameters of accuracy with the users.

7.8 Use of named ranges

In Excel, cells are referred to by a grid reference – column/row, eg, A1, T9, etc, – or by using the R1C1 notation. Ranges of cells are referred to in a similar style, eg, A1:H1, T9:V12. Excel also allows users to assign alphanumeric names to cells and ranges of cells. These names can then be used in formulae in substitution of grid cell and cell range references[3].

The use of range names is a hotly debated topic in modelling circles. Some modellers advocate near-exclusive use of named ranges in models and formulae, and others suggest that names should be avoided at all costs. The benefits of using names are that they can make formulae more intuitively readable and interpretable (eg, 'Sales = Volume * UnitPrice') and make widely applied assumptions (eg, VAT rates) readily and simply visible in formulae. They also make it easier to anchor ranges referenced by macros (see Hint, right).

The disadvantages are that names can become high maintenance, very long and they can obfuscate the review process.

Using names in large or complex models requires disciplined taxonomy and management of name use, which is not necessarily a bad thing. However, this can become onerous to maintain, and has the attendant risk that the names become cryptic and difficult to interpret, or that the discipline is not adequately maintained.

Our main reservation about the widespread use of named ranges relates to the review process. Named ranges divorce formulae from physical orientation in the spreadsheet, and can make it more difficult to visualise, understand, and review a formula's precedents. Ultimately the extent of use and application of names is a matter of modelling style. We tend to advocate a limited use of names for commonly used or static assumptions, and to support specific macro functionality, rather than promoting an all or nothing approach to name use. Once an approach has been adopted it should be applied consistently.

Hint

Unlike Excel worksheets, the VBA code in macros does not automatically update any cell references when blocks of spreadsheet code are moved or when rows/columns are inserted or deleted. Named ranges referred to by macros do not suffer from this limitation.

7.9 Key points

- Always model the timeline and time-based calculations using a consistent base period (eg, monthly, quarterly, etc,) and at the lowest common denominator that provides meaningful accuracy.
- If you need to present summarised groupings of data (eg, annual totals, or semi-annual forecasts for later years) build the model using consistent time periods at the lowest sensible time unit denominator and simply aggregate results for output/presentation purposes.
- Identify the most relevant entity level building block for the business, this will depend on data available and will not necessarily be individual legal entities.
- Choose an intuitive sign convention and apply it consistently.
- Always prefer a horizontal timeline presentation over vertical.
- The model should be accurate enough for its intended purpose and no more, avoid spurious over-accuracy.
- Named ranges can make formulae more intuitive to understand and are a must when automating procedures with macros. They can become cumbersome to maintain.

3 Named ranges can also be defined to refer to constants, however we would tend to discourage their use in this way due to the reduction in transparency which comes from entering values in places other than on a sheet.

8 Applying the best practice principles

8.1 Overview

In chapter 5 we introduced six golden rules, core principles that when followed reduce risk and facilitate quicker updating and amendment of models:

- modularity;
- consistency;
- transparency and simplicity;
- linearity;
- integrity; and
- protection.

In this section, we explore each aspect in more detail, providing guidance and advice on how to interpret and apply them in practice.

8.2 Modularity

A model should be constructed on a modular basis, by which we mean two things: inputs must be clearly separated from workings and from outputs; and individual blocks of calculations should be self-contained as far as is reasonable.

8.2.1 Separation

The separation point is particularly important. In practice:

- There should be a discrete area for inputs (ideally one or more separate worksheets, though a clearly identified and separated inputs area on a working sheet can suffice).
- Inputs should be clearly identifiable through colour coding/format (indeed other types of cell should also be colour coded – see section 11.6).
- Each piece of input data should be entered into the model once and only once, and dependent calculations should also not be carried out more than once.
- Input values should not be hardcoded directly into formulae.

The benefits of clear separation are that it:

- Signals to the user and/or reviewer where they will find the key data used in the model and what that data is.
- Brings transparency, order and organisation to the modelling process, avoiding orphan inputs embedded in calculations, and aiding change management in a fast moving transaction context.
- Reduces the risk of duplication of inputs (such duplication creating the risk of dual calculations and of such inputs becoming out of step).
- Aids the audit trail and management of version control.

Hint

To quickly format inputs, select the input tab and hit the F5 function key. Click on the 'special' button and then select 'constants' and 'numbers' before hitting 'OK'. This selects all the cells with input numbers, to which formatting can then be applied as a group.

8.2.2 Self-contained calculations

Calculations should be a mirror of the overall modular structure. Each calculation block or working on the calculations worksheet should have three components: an input section (links to inputs called-up from the input sheets), calculations (the processing of those inputs), and the summary outputs (the results of the calculations that will be referenced elsewhere).

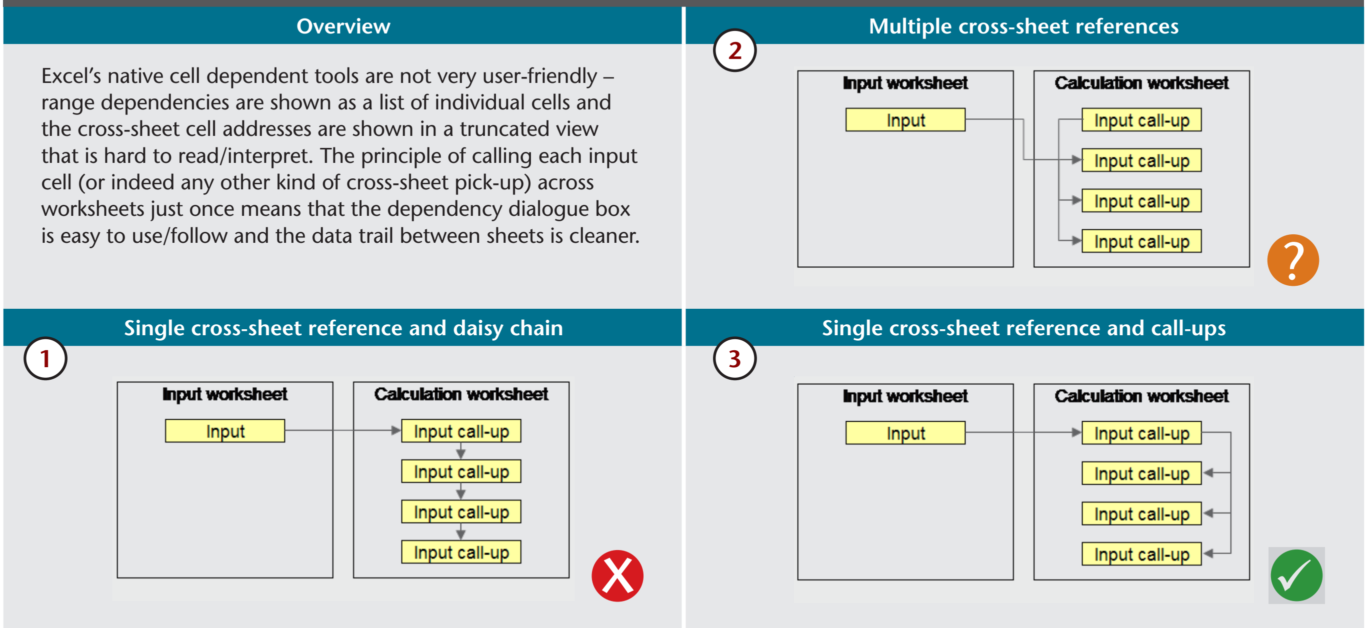
We use the term 'call-up' to refer to the practice of separately calling-up the relevant inputs and components needed for a calculation before they are used in that calculation.

This approach has a number of advantages:

- Each calculation is self-contained – you can clearly see what is going on (the variables used, the nature of calculations carried out, and the results), without having to scroll or page across or up and down worksheets unnecessarily.
- The plumbing becomes hardwired in, facilitating easier changes to calculations – ie, the workings can be amended without impacting the linkages to the rest of the model.
- Navigation and investigation of formula dependents and precedents across and between worksheets (ie, the audit trail) can become easier to follow (provided inputs are only called across worksheets once).

Figure 7 below shows conceptually what we mean by a call-up. In particular, it illustrates the final point and why it is important.

Figure 7: Calling up data and calculations



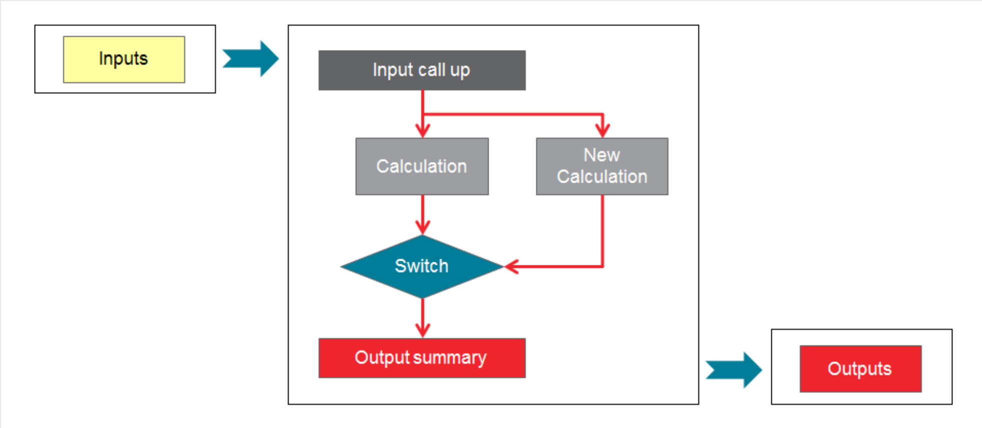
Commenting briefly on the three options shown:

- Option 1 'daisy chaining' leads to a confused and lengthy audit trail and increases the risk of error, since if any intermediate step between the input and its use becomes corrupted, it can impact multiple calculations and will be harder to trace.
- Option 2 'direct references' can work well in simple models, it should mean that any call-up of an input is only one jump away from the source input and easily traceable. Reviewing links across sheets from the source input to see where it is used can be more painful, particularly where the input is widely referenced.
- Option 3 'call-up and reference' sees a single cross sheet jump and then direct references within the sheet. This has the advantage that the link between sheets is most transparent, it slightly lengthens the chain between on-sheet call-ups and the source input but represents our recommended/preferred approach.

There are advocates of the call-up approach who take it to extremes and call-up every input for every individual calculation. In cases where there are repeated blocks of like calculations, this can result in a plethora of white noise. This occurs where the calculations and substance are vastly outweighed by the cross referencing of data and precedents. Perversely, such models can be harder, not easier, to review. We strongly advocate the principle of using call-ups and self-contained calculation blocks. However they should be used sensibly to assist the audit trail and transparency, and not so much that they create a fog.

The benefit of self-contained modular calculations is that they create an architecture that is more readily updated and amended, as illustrated in Figure 8. This is particularly important in a fast moving deal context, where it is not unknown for swingeing changes to be requested in the face of shortening deadlines. A modular structure allows new workings to be incorporated rapidly (piggybacking the existing hard wiring) while also allowing original workings to be retained in case the model sponsor/user changes their mind, or needs to create a bridge back to earlier analysis.

Figure 8: Illustration of change management with modular structure



8.3 Consistency

Consistency is a critical component of a best-practice model. It shares elements in common with the principle of linearity, in particular, the idea that the model should in many respects be designed to be read like a book: left to right and from the top of the sheet down.

8.3.1 Formula consistency

Applying the principle in practice, there should be only one formula per row under the active timeline, and they should be consistent from left to right along that row. (Paying heed of course to the need for the formula to have been designed in such a way as to be relevant to each period.) The discipline of left-right consistency within the main calculation block should be applied rigorously; it has a number of advantages:

- From a review perspective – reviewing a single cell can provide comfort over the row.
- From an error prevention perspective – corrections can be made once and copied across without fear of overwriting hard-coded data or uniquely altered formulae.
- From an efficiency perspective – it facilitates the use of keyboard shortcuts (see chapter 14).
- From a programming perspective – it makes life easier, avoiding the need to code around inconsistent time periods, which can add significant complexity to a formula (see earlier comments on timeline in section 7.3.1).

One particular approach we often see is forecasting models prepared on a monthly basis but with annual or even quarterly subtotals along the timeline. Where such subtotals are included within the timeline as new columns, there would be an inconsistency in formulae going along the timeline. (ie, reading from left to right there would be calculations, then the subtotal formula, then back to calculations again.) This not only causes inconsistency due to the presence of the subtotal formula, but also it is likely that the first column after the subtotal will need to be different from its neighbours if it is referencing the previous period which is now offset by two columns due to the subtotal column. A better approach to this requirement is to calculate the subtotals separately from the main calculation block, thereby allowing the main block to retain row-wise consistency. From a presentation perspective these sub totals can easily be reinserted into the timeline using consistent formulae on the output sheets. (See section 9.3 and Figure 10, opposite).

This principle can also be applied from top to bottom for contiguous rows used to perform the same fundamental calculation on a batch of like data, as illustrated in Figure 9. The example also illustrates the use of $ signs to anchor an input row used in a calculation.

We have not discussed absolute references in Excel in any detail. However, one piece of advice in relation to absolute references is to avoid over anchoring – using absolute references on both rows and columns when only one or the other is needed. The disadvantages of over anchoring are that it can make the copying and recycling of blocks of code more time-consuming, it can make formulae harder to read, and it can create unnecessary breaks in calculation block consistency which in turn artificially increase the required review effort.

Figure 9: Illustration of formula consistency

	A	B	C	D	E	F	G
1	PERIOD		1	2	3	4	
2							
3	VAT RATE		17.5%	20.0%	22.5%	25.0%	
4							
5	PRICE - £						
6		A	5	5	7	7	
7		B	10	12	15	15	
8		C	15	15	18	19	
9							
10	VOLUME - units						
11		A	100	80	50	70	
12		B	50	55	70	80	
13		C	40	30	45	25	
14							
15	TOTAL REVENUE WITH VAT						
16		A	=C6 * C11 * C$3	80	78.75	122.5	
17		B	87.5	132	236.25	300	
18		C	105	90	182.25	118.75	
19							

The snapshot above shows that a single formula structure can be used to calculate the twelve sales outputs.

8.3.2 Timeline consistency

In addition to formulae consistency, the timeline units should be consistent. (See section 7.3.1.)

By way of example, if a lender needed to evaluate monthly cash flows during the early period of a prospective loan but semi-annual or annual figures would suffice for later forecast years, the model should be built throughout on a monthly basis. The monthly results for the later years could then be grouped together and aggregated for presentation/output purposes for the relevant periods.

Figure 10 shows an illustration using SUMIF to consolidate months into quarters. The same result can be achieved using other approaches such as SUM combined with INDEX.

Tip

Although not required for this example, SUMIFS can achieve the same results as a SUMIF formula, but provide the added benefit of extra flexibility if the modeller wishes to add new criteria to the end of the formula; if a modeller wished to add an extra condition to a SUMIF statement, they would need to restructure the formula before changing to SUMIFS and adding criteria.

Figure 10: Presentation and aggregation of time series results

VLOOKUP | =SUMIF($I$3:$T$3,D$3,$I10:$T10)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2				Quarterly summary										
3	Quarter number			1	2	3	4		1	1	1	2	2	2
4	Period Start			01/01/2016	01/04/2016	01/07/2016	01/10/2016		01/01/2016	01/02/2016	01/03/2016	01/04/2016	01/05/2016	01/06/2016
5	Period End			31/03/2016	30/06/2016	30/09/2016	31/12/2016		31/01/2016	29/02/2016	31/03/2016	30/04/2016	31/05/2016	30/06/2016
6	Days in Period			91	91	92	92		31	29	31	30	31	30
7														
8	INCOME STATEMENT													
9		Revenue			324,772	344,651	365,746		100,000	102,000	104,040	106,121	108,243	110,408
10		Cost of sales		=SUMIF($I$3:$T$3,D$3,$I10:$T10)	(257,780)	(271,952)	(286,903)		(80,000)	(81,440)	(82,906)	(84,398)	(85,917)	(87,464)
11														

Similarly, if you expect later to need to evaluate quarterly figures but only have annual data to start with, it is generally better to build the model on a quarterly basis, disaggregating the annual data (on an approximate or pro-rata basis) and using it in quarterly calculations from the start of the model's development, than to try to retrofit quarterly calculations later.

8.3.3 Worksheet consistency

The other key tenet of consistency is that the sheet layouts should, as far as possible, mirror each other in terms of their columnar layout (for a horizontal timeline model). In other words any given column on any given sheet should represent the same time period in the modelled timeline. The point of this is again to aid review and assessment of the model and to speed up programming. Having a common base column (the first time period) means that deviations from this in formulae will be more apparent and it will be readily memorable when writing new formulae. This point is revisited in section 9.3 and illustrated in Figure 19.

8.4 Transparency and simplicity

Transparency and simplicity follow the fundamental KISS principle: 'Keep It Simple, Stupid!' As noted earlier, the acid test of a simple, transparent model is that the user can readily understand and replicate its results from a print out of the workings – based on the row descriptions/labels and the results shown – using nothing more than a calculator.

8.4.1 Transparency

To achieve transparency, the modeller should:

- Keep formulae down to bite-sized chunks, breaking compound calculations into multiple separate parts.
- Use clear and meaningful row descriptors.
- Identify units and the flow of logic clearly.
- Avoid cross-sheet references within formulae wherever possible, in other words, call cross-sheet links up in separate cells onto the relevant worksheet then cross reference them within that same sheet.

It can also be useful to use spaces, carriage returns, particularly to break out the different legs of an IF test, or to make mathematical operators more obvious. However, breaking formulae into smaller calculation steps is most important and can obviate the need for spacing.

Similarly, it is helpful to separate out repeatedly used logic tests (for example, date-based logic to flag VAT quarters or accounting year ends). Code these as a separate block at the start of the calculations page rather than repeating them within each formula to which they apply. (See also 7.3 and 14.4.)

8.4.2 Simplicity

Excel 2007 onwards provides a vast area of spreadsheet real estate to work with compared to earlier versions which were limited to 255 columns. It is much preferable to use multiple blocks and lines of simple self-explanatory code than to try and cram the logic into a single jumbo-formula, and with the huge amount of space available in Excel 2007 onwards there is no reason not to make use of the additional space and make formulae clearer.

Formulae should be kept as short as possible – and should ideally not span multiple lines in the formula bar. As we highlighted earlier, the 'rule of thumb' provides a good guide: if your formula as shown in Excel's formula bar is longer than your thumb, then consider breaking it into separate steps.

Sometimes a particular calculation will require the use of advanced features, but sometimes the same result can be achieved using a series of simpler formulae, perhaps spread over a few lines. There may also be a temptation for some modellers to demonstrate their knowledge of the software, by making use of advanced features which, whilst cosmetically efficient, may not allow all users of the spreadsheet to easily follow the calculation trail. As such we would advocate using simpler features where possible, to make the model more accessible to non-specialist audiences.

In section 5.5 we considered how modelling relationships dynamically can simplify the range of inputs a user is required to review and manipulate in order to achieve a given desired result. We see the dynamic expression of business drivers and their interrelationships as fundamental to any proper financial modelling exercise. A static model based on a large number of individual values will be cumbersome and complex and negate many of the benefits that a good financial model can bring to a transaction process.

In practice, many relationships can be reduced to two simple components: a base value and a set of instructions as to how it will behave or evolve over time. The base value can be a simple input (for example baseline annualised sales) or a link to a calculated value elsewhere in the model (for example calculated sales for each given year). The instruction set would comprise the other inputs and coding required to generate a time series vector from the base value.

For example, for sales, the inputs and coding might comprise an input annual growth rate and (separately) the calculations which apply that growth rate to the baseline annualised sales (on a compounding basis) for each year in the forecast. To calculate debtors, the inputs would comprise the ratio of debtors outstanding to sales and the annual sales calculated as set out in the previous sentence. The calculation would link the two components to work out closing debtors and as a consequence of that, cash receipts. You can see that three inputs (baseline sales, sales growth, and debtor days) can be used to generate time series results for sales, debtors, and cash receipts. Further, you can see that if well structured, the effect of a change to any one of these three individual inputs would automatically be calculated and reflected in the financial results of the model.

From a user perspective, such dynamism simplifies the user interface and facilitates the rapid scenario and sensitivity analysis which is a key benefit that a well-constructed model brings to a transaction.

8.5 Linearity

The flow of logic within the model should be linear wherever possible, flowing down the worksheet, and from left to right. This makes the model intuitively easier to follow and also helps Excel to process the calculations more quickly. Occasionally formulae necessarily have to refer to results further down the page (for example, opening balances in a corkscrew working as illustrated in Figure 22), and where this is required such 'contra- flow' calculations should ideally be clearly sign-posted, for example through distinct formatting.

8.5.1 Circular references

At its most extreme, a non-linear calculation results in a circular reference.

A circular reference arises where a calculation in a cell depends in turn on its own result (whether directly or indirectly). The classic example is the calculation of interest on cash balances: we often see the interest calculation expressed as the average of the opening and closing cash balances in the model multiplied by the interest rate.

The closing balance should properly include interest earned on cash during the year and so if this approach is used, the closing balance depends on interest earned which in turn depends (in part) on the closing balance, which depends on ... and so on (see Figure 11).

Figure 11: Simple illustration of circular reference

	A	B	C	D	E
1	Opening Balance		100		
2	Interest		10	= (C1+C3) / 2 * $C$5	
3	Closing Balance		100	= SUM(C1:C2)	
4					
5	Interest Rate		10%		
6					

Another example is forward looking financial covenants. For example, the level of dividend distribution allowed may depend on the forecast cash position in future periods – those cash positions in turn depend on how much is distributed now, which depends on the future cash position, etc. In our experience, circular references can introduce instability, disguise other possibly unintended circular references and, in extremis, result in incorrect results being presented.

8.5.2 Iteration

Excel allows the user to solve circular references through iteration. Iteration refers to the Excel setting whereby the affected formulae are calculated multiple times until they converge on a result. While iteration may seem an appropriate solution in certain simple circumstances, it gives rise to a number of problems.

Hint

If Excel has the word 'calculate' in the status bar and it stays in view after 'F9' has been pressed, then iteration is turned on and it is likely the model you are looking at has a circular reference.

Figure 12: Simple illustration of iteration

	A	B	C	D	E
1	Opening Balance		100		
2	Interest		10.53	= (C1+C3) / 2 * $C$5	
3	Closing Balance		111	= SUM(C1:C2)	
4					
5	Interest Rate		10%		
6					

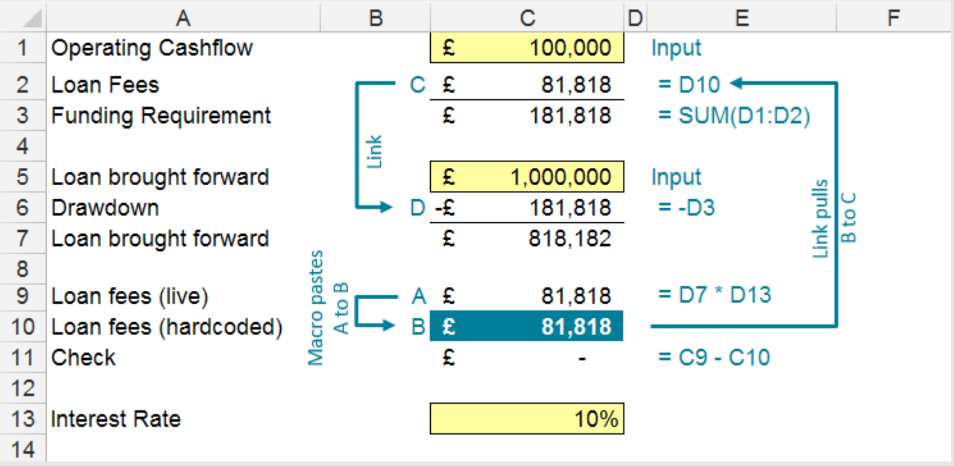
The first potential problem with iteration is that there may be more than one mathematical solution to the problem or even that convergence may not be possible, which may mean results are inherently unstable. The second problem is that once you turn iteration on for the first piece of circular logic, it is then possible for unintended and/or erroneous circularities to be introduced without you becoming aware of them. Thirdly, debugging and tracing circular references can be onerous and time-consuming. The final major concern with iteration is that it is memory and processor intensive, ie, just plain slow.

Circular references should, therefore, be avoided at all costs.

Tips to achieve a linear model:

- **Planning** – at the specification stage, set out the layout of the workings and plan them such that they naturally flow in a linear order through the worksheet tabs and workbook (ie, revenue before costs before profit before tax…).
- **Consistency** – apply formulae consistently from left to right and top to bottom within blocks of code (ie, all formulae should mirror the top left formula in their contiguous block).
- **Use a time weighted approach to interest calculations** – rather than use the average balance, use the opening balance plus time weighted adjustments for the cash flows arising during the period (see section 10.4).
- **Control iteration** – if the use of the model requires an element of inherently circular computation, rather than use uncontrolled iteration, break the circular reference and use a simple copy/paste macro to control the iteration, as illustrated in Figure 13.

Figure 13: Breaking circular references



8.6 Integrity

It is intrinsic to financial modelling best practice that a model should exhibit integrity. This is a function of completeness (an integral whole) and logical robustness (reliable, accurate calculations).

8.6.1 Completeness

Most deal models reflect one or more actual business entities in their entirety. As noted elsewhere, unless circumstances are unusual, such models should always include a fully integrated balance sheet, income statement and, cash flow statement. We sometimes see models that just have cash flow or income statement data. Adding a balance sheet (even at a simplified proforma level) gives assurance that the underlying data and the construction of the income statement and/or cash flow are sound: that they have inherent integrity. We consider it a must-have.

The model should also ideally contain all the data and calculations relevant to its purpose; gaps, linked files, or multiple sister files are generally to be avoided because they leave models prone to unintended consequences of changes to the source file. For example, if the source file is updated to a new version, renamed or its file path changes, the destination file will not update with it. Similarly, if structural changes, like inserting a row, are made to the source file while the destination model is not open, the destination file will not update any linked formulae and will most likely result in errors or incorrect results.

In circumstances which require multiple files (either due to file size or work flow considerations) a good alternative to live links is input 'landing pads'. The concept is that in the source file there is a dedicated output sheet which has links to data elsewhere in that file and a corresponding input sheet of the same format in the destination file. The modellers do not amend the format of either source, or landing pad, and data is transferred by a copy and paste operation which can be manually controlled and managed. This approach lends itself to automation with Visual Basic for Applications (VBA), using a simple macro to source, copy and paste the data upon a manual intervention by the user. The other advantage of this technique is that by keeping the format of these sheets the same, the other sheets which connect to them can be edited and yet the integrity of the system maintained.

8.6.2 Robustness

Integrity is not just about the balance sheet. Integrity is also about the flow of data through the model – ie, that inputs survive their journey through the calculations to the outputs and results with their integrity (ie, meaning) intact. Testing for this takes two routes:

analytical review or simple proof-in-total of the results; and inbuilt checking formulae. As well as logic-based checks, it is useful to build in a range of commercial alerts and warning flags to highlight to users when modelled outcomes fall outside given parameters.

We advocate including error and commercial sense checks in the model as it is developed, and collating the results on the model's dashboard or a dedicated check sheet.

Suggestions for enhancing integrity:

- Build in integrated financial statement outputs at the outset, with a simple balance sheet check. Each new working should be plumbed across each of the three primary statements as appropriate (for example, fixed asset purchases impact both the balance sheet and cash; depreciation impacts the income statement and the balance sheet).
- Build in error checks as you develop the model and come across problems or risks. For example, in calling-up data across sheets, there is a risk that individual items are missed or incorrectly referenced; a simple answer is to sum the individual items called-up and compare that total with the total on the reference sheet. It may sound basic (or even overkill) but it is effective at preventing simple lookup, summation, and data collation errors.

Figure 14: Error checks and conditional formatting

Checks	
Balance sheet balances	Ok
Bank covenants are met	Check
Cash ok or overdrawn	Check
Loan repaid in full	Ok
Target DSCR achieved	Ok
Summary	Check

- Build in alerts or warnings to look for adverse results/variances. For example, in certain sensitivities you might expect to see covenant breaches, nevertheless it is useful to highlight that they are there and when they occur. Furthermore it is possible that such a breach arises for reasons other than expected (eg, coding error, incorrect inputs) and these tests act as a helpful prompt to direct investigations.
- Collate all the checks in one place and include a model status summary check for both error checks, and sense checks or alerts (ie, a formula that looks at all the error check cells and returns an error if any of them are breached). (See Figure 14.)
- Include a reference to the summary check at the top left of each worksheet to provide an immediate visual warning to developers and users should they inadvertently introduce a coding or input change that results in an error or alert.
- Building on the idea of checking and displaying errors, and potential errors or alerts: use conditional formatting and meaningful user messages to flag up errors boldly.

8.7 Protection and validation

Excel's protection and validation tools should be used to control inputs and prevent misuse or corruption of a model.

A typical model has scores of individual inputs and a colossal number of formulae. With each new formula the number of permutations of outcomes increases exponentially.

In addition, some combinations of inputs can impact a model's answer in unforeseen ways and it is not practical through logic tests alone to screen out all such eventualities. Furthermore, user error – overtyping formulae, putting text in numeric cells, using 'out of range' data, and in some cases simply inserting rows or columns – can contribute to further problems.

Hint

In 2012, AstraZeneca inadvertently released confidential company information to analysts, by accidentally embedding it into a spreadsheet sent to sell-side analysts.

If there is a risk that a spreadsheet might contain undetected confidential information then it may be more appropriate to selectively copy the data to be shared to a new Excel document, or to share a PDF version.

Source: EuSprIG

To avoid these issues, we advocate (early on in the process) designing restrictions on what users can see or change in the model. This can be done using Excel's protection and validation functionality. Note we caution against overuse or unnecessary use of these features, while they have a useful function in helping manage risk, once a model is in circulation in a deal context they can cause irritation and wasted time.

8.7.1 Protection

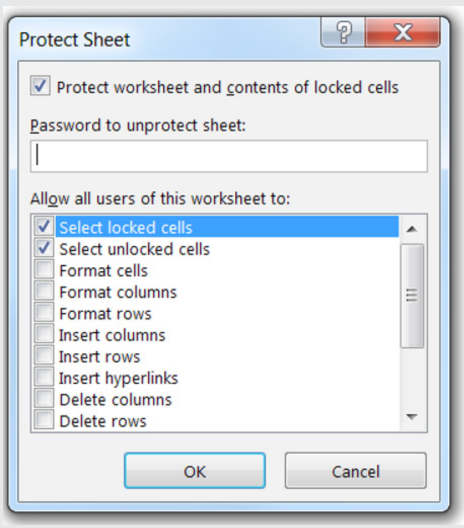
Using Excel's 'protection' functionality, the modeller can implement access restrictions at a variety of levels:

- Cell protection. In all bar strict input cells (which should be separately identified and formatted) set cell protection to 'on'. This has the effect of 'protecting' those cells, and preventing them from being edited or amended when worksheet protection is enabled.
- Worksheet protection. 'Right-clicking' a worksheet tab reveals a drop down menu with the option to 'protect sheet'. (This can also be found under the review menu on the ribbon.) This gives the option of adding a password. During development, when sharing the model, it is often useful to set protection to 'on', without a password, to help prevent accidental changes. The user is still able to disable the protection and make amendments but this then becomes a more purposeful act. Post-development or when models are circulated more widely there may be reason to use a password.

  Note that worksheet protection needs to be separately enabled for each tab. Excel offers a menu of options (see Figure 15).
- Workbook protection. In addition to protection at worksheet level, under the review tab of the ribbon, Excel provides the option to protect the workbook. This offers two settings, 'structure', which prevents users from adding or deleting worksheets or from displaying hidden worksheets; and 'windows', which prevents users from changing the size or position of worksheet windows (although Excel 2013 and 2016 do not support protection of window structure). As with worksheet protection, the option to apply a password is given.
- Workbook encryption. The final layer of protection offered by Excel is to password protect and encrypt the workbook to prevent unauthorised access.

Using these tools in combination can assist the modeller to prevent unintended or deliberate misuse of the subject matter model. Be aware that selecting certain options can restrict the user or reviewer's ability to navigate through the model, which can inhibit the review process.

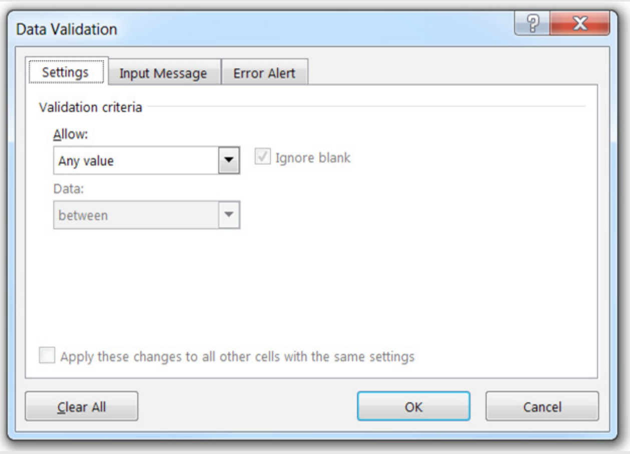
Figure 15: Worksheet protection menu



8.7.2 Validation

Data validation refers to the functionality offered by Excel to control and restrict the values a user can select in certain cells. It can be found under the data tab on the ribbon and is controlled through a dialogue box and menu of options.

Figure 16: Data validation menu



Data validation allows the model author a wide range of choices as to how to limit inputs for a given cell, which will be helpful in assisting users of the model to input valid data. For example, it allows the user to specify ranges of numbers, types of input (text, integers, dates) or even user-defined drop down lists. However, it should also be noted that it is possible to bypass the validation, by copying values from elsewhere and then using the 'paste as values' option into the cells with validation applied.

The control of these choices is through manual entry into the data validation dialogue box (see Figure 16). One issue this creates is that any restrictions the modeller selects are themselves inputs of a sort, but they become hidden – embedded within the validation settings for the relevant cell. Such assumptions require careful upkeep and can become cumbersome to maintain.

We recommend that when validation is used to limit the user to a specific value or inputs (ie, other than where the type of input is restricted) the 'list' option is selected and linked to an explicit range in the workbook (see Figure 17). The reasons for this are to make the assumptions explicit, visible, and easier to edit (the linked range automatically updating for insertions/deletions).

Combined with the SUMIF or INDEX and MATCH functions, inputs selected through data validation lists can contribute to the control of scenarios and selection of options. For example, the option or label description selected from the validation drop down list for an input cell can be cross-referenced against the source list of options to which it belongs. Its relative position in that list can be identified (using MATCH) which can then be used to select a linked data point or data set to be used as a consequence of the initial selection or choice made (see Figure 18 for an example, in this case using the SUMIF function rather than INDEX and MATCH).

A war story

The London Olympics 2012 organising committee oversold four synchronised swimming sessions by 10,000 tickets, due to incorrect manual data entry to a spreadsheet.

While data entry can be subject to human error, this risk can be minimised by utilising data-validation techniques. For example, by selecting the venue from a pre-defined list, this could automatically pull through the venue capacity.

Source: EuSprIG

Figure 17: Data validation example using lists

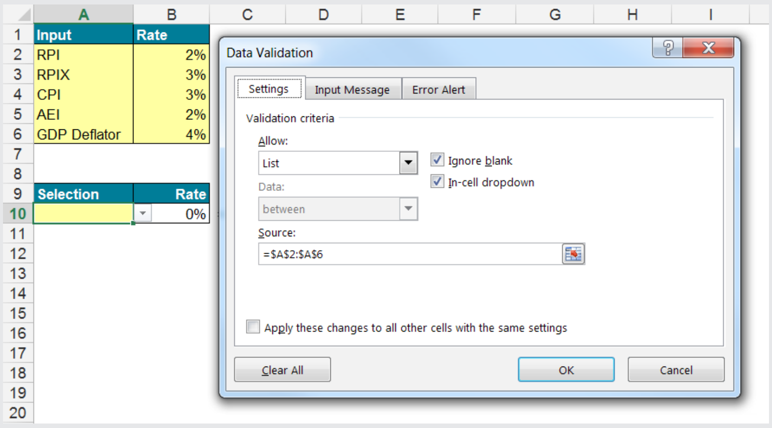
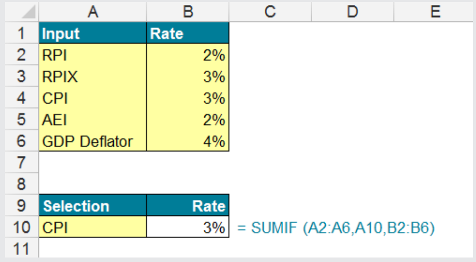


Figure 18: Use of SUMIF



Another useful feature that can be created is a contextual validation list. For example, by using dynamic named ranges within your validation specification, you can create linked validation lists, whereby the first drop-down selection box acts as a filter for the second box. It should be noted that this is an advanced technique, which is not natively supported by Excel, and as such should undergo careful testing.

By way of example, the user may be looking for the 'fixtures and fittings' item. By selecting 'balance sheet' in the first selector, the second list would update to include balance sheet items only. This could then be taken further by introducing additional levels of classification, such as 'assets' and 'fixed assets'. This can improve usability and help to guide users towards using the appropriate classification when inputting financials.

8.8 Key points

- Keep inputs strictly separate from calculations and outputs.
- Structure each calculation to be self-contained (call-up inputs, process them, summarise results).
- Adhere to the principle of one formula per row within the timeline calculations.
- Use consistent time periods throughout the model and lay out the worksheet structure consistently.
- If periodic subtotals are required in the timeline, avoid breaking row consistency by placing them before or after the core timeline.
- Use meaningful row descriptions.
- Keep formulae short and simple (prefer many bite-size formulae to few large ones).
- Highlight contra flow calculations and avoid circular references and uncontrolled iteration.
- Build fully integrated financial statements and a full suite of error checks.
- Make appropriate use of protection to stop accidental user amendments or deletions.
- Use data validation selectively, to control and limit input values, and to select options, using it as an error trap to stop users inputting data types or values that are not calculable.

9 Model structure and flow

9.1 Overview

This section considers the architecture of a spreadsheet model and looks at the basic building blocks available to the modeller: workbooks, worksheets, and calculations.

An Excel file is also known as a workbook, which in turn comprises a number of individual worksheets or 'tabs'. The cells in the worksheet contain the formulae or calculations and grouped together, the 'calculation blocks'. This section looks at:

- the overall structure of a workbook – by reference to worksheet's purposes and functions;
- the layout of individual worksheets; and
- approaches to building up calculation blocks.

9.2 Workbooks

Within a typical financial model, the worksheets can be classified according to one of four functions:

1. **Information**: worksheets that provide user guidance, reference information, and navigation/printing support (eg, disclaimer and contents sheets).
2. **Inputs:** worksheets that contain raw data and input drivers. A further distinction being that inputs are frequently best handled when split between static (or non-time-based inputs) and dynamic (or time-based inputs).
3. **Calculations:** worksheets that reference the inputs, process them, and calculate the required results.
4. **Outputs:** worksheets that present and communicate the results; in the case of dashboards, sometimes with an input linkage too.

From a model development perspective, we would typically group like sheets together and lay them out in the workbook from left to right in the above order; a common exception being the inclusion of a dashboard or summary outputs worksheet at or near the start of the file. In structuring the workbook there are three other practical considerations:

- **Volume of worksheets**: in some models, the calculations are laid out over numerous worksheets, in others they are grouped on a single sheet. Too many worksheets and the model becomes hard for users to navigate or to orient themselves in (though this can be partially mitigated); too few, and the complexity of the main calculation worksheets increases and they themselves become difficult to navigate.

  Generally speaking, models with fewer 'deeper' worksheets calculate more rapidly and this can be an advantage in complex applications. There is no right or wrong answer. While we would generally advocate using fewer sheets, in practice it depends on careful consideration of the subject matter, the structure of the data, and the entity level building blocks.

- **Worksheet naming convention:** In Excel, the worksheet names are listed on the tabs at the foot of the page. For ease of use, it is helpful to have all the main worksheet names for a model visible within the active window – this greatly aids orientation, navigation through the model, and model use. A simple observation: using shorter (though still meaningful) abbreviations allows more worksheets to be visible and readily accessible to the user and helps keep formulae shorter. It is common to use short prefixes such as 'I_' for input sheets and 'C_' for calculation sheets to classify tabs according to their function. Colour coding the tabs according to their function described above can also help. The need for visibility and accessibility is an argument in favour of using fewer sheets.

- **Consolidating entity templates.** While the number of worksheets should ideally be constrained, for businesses with multiple subsidiaries or operating units, it can be convenient for each subsidiary to be modelled (at the operational level) on a single worksheet. The advantages of this are:
  – a comprehensive entity level view that can be extracted and shared; and
  – that the subsidiary worksheets can be 'bookended', and the results can be summed together across the worksheets.

  The latter point can greatly assist consolidations and facilitate the swapping in and out of businesses/divisions for scenario analysis. Note however, that if the template changes – all relevant worksheets plus the bookend sheets need to be amended consistently (and preferably as a batch) otherwise data may be missed. Accordingly, we would only recommend this approach is used by experienced modellers and in conjunction with thorough inbuilt error checking routines.

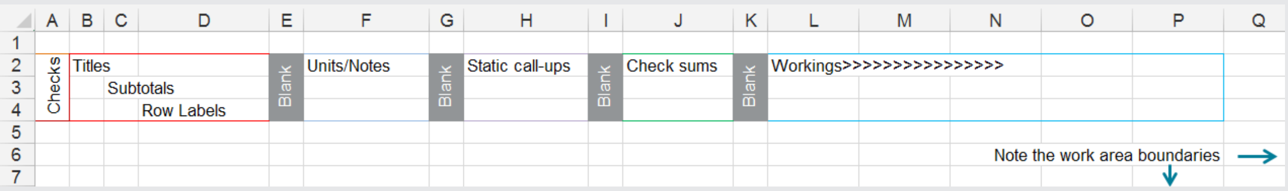
9.3 Worksheet

Worksheets are the structural building blocks of an Excel workbook. As a fundamental design and development principle, these building blocks should be as consistent as possible.

Generally speaking, worksheets should be laid out using a horizontal timeline and a consistent columnar format. An illustration is shown in Figure 19 but note that this is not intended to be an absolute guide, rather to illustrate some general principles:

- To bring meaning to the calculations, a number of columns to their left will be needed, to set out and to order the labels and row descriptions. We would also use this area for communicating other information related to the calculation in that row (units, standing inputs that have been called up, etc), and for checks and check totals. The specifics are a matter of context and individual preference.
- Once the modeller has decided which columns to reserve for informational purposes, the consistent timeline based calculation columns should be laid out. We generally advocate leaving a gap or blank column between the informational and computational columns to facilitate corkscrew calculations (see section 9.4) and/or call up of standing data.
- The actual calculations should be performed once and then referenced, be consistent from left to right, and start in the same base column on all worksheets that use a common timeline.
- Selecting the columns and rows outside the active modelling area (ie, to the right and below) and hiding them (or zeroing their width/height respectively) places a fixed boundary around the area in use. This is helpful in facilitating efficient use of Excel's keyboard shortcuts to navigate around the worksheet, and mitigates users and reviewers 'overshooting' the calculation area.

Figure 19: Conceptual illustration of worksheet column layout



The illustration above focuses on the left-right columnar layout of the workings on a sheet. Consideration should also be given to the inclusion and layout of standing information generic to the model. We advise reserving an area at the top of each sheet containing information common to each worksheet, as well as navigation links, error checks, and tools that assist the user. It is also often helpful to the user for the model author to ‘freeze panes’ such that the row descriptions and the timeline/header rows remain visible as users scroll through the sheet. Possible items for the header area include:

- **Model name and sheet description**. For printing purposes and ease of review, it is helpful to have some basic information about the model (project name, model version, selected scenario, run date, etc) visible in the top pane.
- **Timeline:** for most sets of calculations/forecasts, the time period will be a key piece of information needed to put forecast performance in context; the header area of the model may include the start/end dates for each period and/or month, quarter or year indicators.
- **Timing flags:** In many cases, there will be relevant information relating to dates and classification of periods that will be key to an understanding of the calculations. For example, if a model presents a combination of actual historical data, with short-term forecasts and midterm projections, it may be useful to show this distinction at the top of each worksheet. Similarly, in project finance transactions, it is important for model users to distinguish between the construction and operational phases in a model’s timeline.
- **Error summary:** if the model contains error checks, a link to a summary check total (such as discussed under Integrity in section 8.6), clearly visible in the header area gives an immediate warning to the user/modeller if they make a change that causes one of the model’s error tests to be breached.
- **Navigation:** for larger, more complex models, the header area might include hyperlinks to a contents sheet or model map to facilitate speedier navigation between worksheets. For convenience we generally advocate including such links in the header area/pane.

So, as a practical matter, it is worth the modeller spending time designing the sheet layout upfront and creating a template (with the column layout, key headers, format styles, print settings, and column widths/boundary all set), before duplicating it as necessary for the relevant worksheets. This can seem time-consuming, but it aids comprehension (which alone helps to avoid errors), and if addressed at the planning stage should save significant time later on.

Hint

Excel does not have a convenient way of presenting the file name or sheet name without showing the full file path. You may use the following formulae to achieve this:

=MID(CELL(“filename”,A1), FIND(“[“,CELL(“filename”,A1))+1,FIND(“]”,CELL(“filename”,A1))-FIND(“[“,CELL(“filename”,A1))-1)

=MID(CELL(“filename”,A1), FIND(“]”,CELL(“filename”,A1))+1, 31)

Figure 20: Practical illustration of columnar worksheet layout

	A–H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Example model / Navigation												
2	Income Statement Worksheet												
3													
4	No errors in worksheet												
5	2 Errors in workbook												
6													
7		Annual Summary						Monthly Figures					
8	Start Date	01/01/2016	01/01/2017	01/01/2018	01/01/2019	01/01/2020		01/01/2016	01/02/2016	01/03/2016	01/04/2016	01/05/2016	01/06/2016
9	End Date	31/12/2016	31/12/2017	31/12/2018	31/12/2019	31/12/2020		31/01/2016	29/02/2016	31/03/2016	30/04/2016	31/05/2016	30/06/2016
10	Year	2016	2017	2018	2019	2020		2016	2016	2016	2016	2016	2016
11	Period							1	2	3	4	5	6
12	Actual / Forecast	Actual	Budget	Projections	Projections	Projections		Actual	Actual	Actual	Actual	Actual	Actual
13	Column Counter							1	2	3	4	5	6
14	Month within calendar year							4	5	6	7	8	9
15													
16													
17	Budget												
18													
19	REVENUES												
20	Fixed	17,400	31,200	45,600	60,000	74,400		1,000	1,100	1,150	1,250	1,300	1,400
21	Variable	17,650	25,800	24,150	24,300	24,150		500	800	1,000	1,200	1,200	1,500
22	Other	1,000	1,500	3,000	3,600	3,000		-	-	-	-	200	-
23	Intercompany	434	8	1	0	0		60	59	85	121	31	45
24	TOTAL REVENUES	36,484	58,508	72,751	87,900	101,550		1,560	1,959	2,235	2,571	2,731	2,945
25													

9.4 Calculation blocks

In the same way that worksheets are the building blocks for workbooks, so calculation blocks represent the bricks and mortar of a worksheet. Extending the analogy with house-building a model should start with the foundations and then layer up the workings step by step. We would normally suggest the following broad sequence for the build phase of a model's development:

1. Model infrastructure

- Set up standard sheet templates and layouts (including print areas and styles).
- Set up and name individual identified worksheets.
- Set up the core timeline inputs and calculations.

2. Model outputs

- Set up the output templates, typically proforma financial statements and key performance indicators (KPIs).

3. Model inputs

- Collate and layout known input data (and placeholders for 'to be obtained' data).

4. Timeline

- Code the timeline calculations (unless already set up at stage 1).
- Programme anticipatable timing based calculations that will be used repeatedly. Eg, flags for a VAT quarter.

5. Calculation engine

Build up the financial calculations required to support outputs, typically starting at the top of the income statement, begin sequentially to plumb in each calculation, considering two dimensions:

- Logic flow – through inputs to calculations to outputs. In other words, for each calculation: enter the inputs on the input sheet, call-up or link to the inputs from the calculation sheet, perform the calculation, collate and summarise the output(s) on that sheet, and link them to the financial statements and KPIs on the outputs sheet.
- Integrity – address the entire impact of each item in turn, so when plumbing in revenue, also code the cash flow (cash receipts) and balance sheet (debtor) components, for costs, also code the payments and creditors. The point of this is that at each step, the balance sheet should balance – this provides an in-built check on the implementation of the coding at each stage of the model's development.

We comment further below on the sequence and techniques by which the modeller approaches the layering up of calculations. The next subsection provides an example order that we have found works well in most situations. See the later subsection for a description of the typical corkscrew working used to model items with a balance sheet impact.

Sequence

As a general rule, we suggest approaching the build up of calculations in the sequence shown in Figure 21. This is an oversimplification, but it is intended to illustrate the broad approach: start with the operational business flows and then progress to funding, financial and fiscal flows. It shows the impact of each item on each of the income statement, the balance sheet and the cash flow statement (noting that some items only affect two of the primary statements). Under each item, the highlighted aspect is best addressed first, in our view.

Figure 21: Recommended sequence of development (illustrative)

Step	Income Statement	Balance Sheet	Cash Flow Statement
1	**Revenues**	Trade receivables	Cash receipts
2	**Cost of sales / Gross margin**	Stock/Trade payables	Cash payments
3	**Overheads / EBITD(A)**	Creditors	Cash payments
4	-	Fixed assets	**Capex/fixed asset purchases**
5	**Depreciation / EBT(A)**	Fixed assets	Free cashflow
6	-	Loan	**Drawdowns**
7	-	Equity	**Investment**
8	**Interest / EBT(A)**	-	Interest paid
9	**Taxation due/PAT**	Tax creditor	Tax paid
10	-	Loan	**Repayments**
11	**Dividends**	Retained reserves	Dividends paid
12	-	Equity	**Redemption**

(Note that the highlighted item in each step represents the suggested starting point.)

Generally speaking we consider it good practice to separate the calculation of operating cash flow from funding cash flows. The model should first calculate the investment opportunity and then look at the funding needed to match the investment profile, and this is implicit in the sequence above.

Corkscrew workings

To model items with a balance sheet impact we recommend using a simple four-line working as illustrated below in Figure 22. Such workings are colloquially known as 'corkscrews,' 'control accounts,' 'T-accounts,' or 'Spirals,' as illustrated by the audit traces shown.

Figure 22: Illustration of corkscrew working

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1				Year		Units		Static call-ups		Check sum/avg		2016	2017	2018	2019	2020	2021	2022
2																		
3		Debtor working																
4																		
5			Call-ups															
6																		
7				Revenues		£						£45,000	£52,250	£58,650	£63,750	£67,600	£72,800	£78,000
8				Receivables months				2										
9																		
10			Calculations															
11																		
12				Receivables BF		£						-	7,500	8,708	9,775	10,625	11,267	12,133
13				Revenues		£						45,000	52,250	58,650	63,750	67,600	72,800	78,000
14				Cash receipts		£						- 37,500	- 51,042	- 57,583	- 62,900	- 66,958	- 71,933	- 77,133
15				Receivables CF		£						7,500	8,708	9,775	10,625	11,267	12,133	13,000
16																		
17			To outputs															
18				Cash receipts		£						37,500	51,042	57,583	62,900	66,958	71,933	77,133
19				Receivables CF		£						7,500	8,708	9,775	10,625	11,267	12,133	13,000
20																		

This approach sets out the impact of a specific working or group of workings in a robust and transparent manner. The technique is relevant to all balance sheet items and can be simplified to three lines for static items (or where data/information is limited). Once set-up and plumbed in, adjustments can be made to the corkscrew working which will automatically ripple through to the balance sheet (and where relevant, the cash flow).

6. Scenarios

Having got the basic plumbing in place, you can begin to add bells and whistles. Inputs for multiple scenarios can now be tabulated in a control sheet, linked through to the static inputs and calculations, with the ability to select 'batch input' changes controlled with a simple switch. Scenario management is further considered in section 13.4.

7. Dashboard

Most deal models have an executive summary or 'dashboard' summarising the key data points and outputs for decision makers. It is usual that these are also dynamically linked to the model inputs through levers and switches (for example using scroll bar and spin button controls, which are found under 'insert -> form controls' on the developer ribbon in Excel 2007 on) to flex key sensitivities and to allow real-time/rapid analysis. Chapter 12 provides further commentary and illustrations in relation to dashboards.

8. Charts

Charts have a useful role to play in displaying results and data; typically they are best added once the primary development of the model structure is substantially complete and stable. They will likely also form part of the dashboard.

9. Orientation

The final stage of the development of the model is to add any tools and functionality that will make it easier for a user to operate the model. For example, we advocate using Excel's 'group' and 'outline' functionality, hyperlinks, and other techniques to help a user find their way around the model and avoid visual overload and disorientation. At this stage any automation of print routines, user interfaces, and worksheet protection should also be finalised.

9.5 Key points

- Worksheets should each have an identifiable function (information, inputs, calculation, and outputs) and be logically labelled, grouped and formatted.
- We recommend using fewer, longer sheets rather than multiple shorter worksheets.
- Worksheet names should be concise but self-explanatory.
- Worksheets should, as far as is reasonably possible, be laid out consistently and set out so that information necessary to interpret results (eg, timeline, active scenario, error checks) is immediately visible.
- The model build should be approached in an orderly manner, generally starting with its overall infrastructure/shape and output templates, before moving through inputs to calculations.
- Calculations should be coded in a logical sequence (typically following the primary financial statements from top to bottom), and should be coded in their entirety (ie, each individual calculation's impact across the three primary statements should be modelled at the same time).
- Corkscrew calculations are recommended for balance sheet items.
- User summaries (dashboards and charts), and aides (scenario control, navigation, etc), should ideally be added once the underlying model is stable.

10 Useful functions and techniques

10.1 Overview

While Excel is designed as a general purpose spreadsheet package, it does include a number of functions which are particularly useful in a financial modelling context.

There are also a number of useful techniques which we return to time and again when building financial models. Some techniques make use of these financial functions in Excel, while others simply use the more basic Excel functions. This section will look at a few of the more common useful functions and techniques.

10.2 Context

In designing and setting up a model, the developer and sponsor will need to give detailed consideration, in particular to revenue forecasting (and the drivers, building blocks, and factors that build up a picture of income), and to the costs and overheads associated with delivering that revenue (their build up and nature, for example whether fixed, stepped, or variable). These aspects will vary significantly by industry and company. There is a vast difference between a support services company, operating a small number of high volume, low margin facilities management contracts, and a software developer providing a transaction platform for the financial services sector.

Owing to the sheer variety of modelling applications and needs within the wider deal environment, a detailed treatment of revenue and cost forecasting is therefore beyond the scope of this guideline. Instead the focus here is to provide guidance on practical and applied modelling tools and techniques that the modeller can use to translate their understanding and modelling specification into a live and dynamic modelling solution suitable for the given transaction or context.

10.3 Debt calculations

Excel includes several functions which are designed to simplify calculations with respect to debt/borrowing. However, in order to implement them correctly within a model it is important to understand what they do, and how they differ, so that the most appropriate formula for the task at hand may be selected and implemented. This section will cover the PMT(), IPMT() and PPMT() functions.

PMT()

This function is used to calculate the fixed monthly payment required to satisfy an annuity or mortgage-style loan at a fixed rate of interest over a fixed term. The standard form of this function is as follows:

rate Interest rate/number of model periods the rate relates to (eg, if the rate is 5% per annum, and the model has 12 periods per year, then rate = 5% / 12 = 0.00417)

nper Number of repayment periods (eg, if the loan is repaid over three years and the model has 12 periods per year then nper = 3 x 12 = 36). Note, this function ensures that each period is of equal length, therefore a consistent model timeline is required.

pv Initial capital value

Payment PMT(rate, nper, pv)

The PMT() function has two additional optional parameters: PMT(rate, nper, pv, **fv, type**).

fv Final value of the loan at the end of the repayments period. For example if the loan has a balloon payment at the end, then this input can be used to force the calculated payments to leave the required balloon value as a residual. (Note, if omitted then FV is assumed to be zero.)

type Specifies whether the loan payment is made at the start or end of each period (ie, is interest for the period calculated before or after deducting the payment from the rolling balance?). 0 = end of period, 1 = start of period. (Note, if omitted, then type is assumed to be end of the period.)

IPMT()

This function calculates the interest charged (or earned) on a mortgage-style loan (or regular savings account) by reference to a similar set of inputs.

rate as per PMT() function

per period in which the interest calculated is charged (or earned)

nper as per PMT() function

pv as per PMT() function

fv optional, as per PMT() function

type optional, as per PMT() function

Interest in period IPMT (rate, per, nper, pv, fv, type)

PPMT()

This function is similar to IPMT, except instead of calculating the interest charged it calculates the capital or repayment element of the fixed payment value for a given period. This is equivalent to the total payment in a period, found through PMT() minus the interest charged in that period, found through IPMT() or a simple interest calculation. The inputs are as follows.

rate as per PMT() function

per as per IPMT() function

nper as per PMT() function

pv as per PMT() function

fv optional, as per PMT() function

type optional, as per PMT() function

Capital paid in period PPMT(rate, per, nper, pv, fv, type)

Hint

In programming with PPMT over a time series, one can program the period values in two ways:

- Period 1 of x remaining periods (where x starts as the total number of repayments and steps down incrementally each period); or
- Period x of n repayments (where n is the total number of repayments and x steps from 1 to n over the repayment term).

Note that the first uses the opening loan balance for each period and the latter uses the starting loan balance.

Practical considerations

A weakness of all of the functions listed above is that they do not allow for mid-term drawdowns or early (partial) redemptions. Therefore we recommend the following technique which will allow for a more flexible loan model:

Per the illustration in Figure 23 the interest rate and periods per year are input, with the rate calculated as the interest per period. The loan maturity period is entered, and in this case is a reference to the model period in which the loan must be fully repaid. In reality this maturity period could be calculated from a target maturity date which would be looked up against the model timeline.

This setup enables the calculation of a ‘remaining periods’ line, shown at row 8 of Figure 23. It is this value which is fed into a PPMT calculation on row 15, which picks up the pv value of the loan from row 10. In effect this is recalculating the capital payment required in this month, to settle the opening loan in the number of periods remaining until maturity. This would be the same value as through the PMT function, assuming there were no additional drawdowns or repayments. However, by rebasing the loan each month in this way, it means that additional drawdowns or repayments are factored into the capital payment value generated by PPMT. One additional consideration is that any additional drawdowns or redemptions are treated as rateable with the original principle, ie, that the loan maturity date is unchanged and the new loan value is repaid over equal instalments.

Interest is calculated on the opening balance and is assumed to be paid in the month. The capital payment varies according to the remaining periods, but does so in a way which keeps the total payment, comprising capital and interest, at a constant value through the life of the loan. It only varies in the illustration below when additional drawdowns or repayments are made.

Figure 23: Interest rate inputs per period

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Interest rate	5%													
2	Loan maturity period	36													
3	Periods per year	12													
4	rate	0.407%													
5															
6		**Col D Formula:**													
7	Period counter	=C7+1	1	2	3	4	5	6	7	8	9	10	11	12	13
8	Remaining periods	=MAX($B$2-D7+1,0)	36	35	34	33	32	31	30	29	28	27	26	25	24
9															
10	B/F	=C16	(10,000)	(9,742)	(9,482)	(9,221)	(8,960)	(9,697)	(9,403)	(9,108)	(8,811)	(7,513)	(7,250)	(6,985)	(6,719)
11	Interest	=D10*$B$4	(41)	(40)	(39)	(38)	(37)	(40)	(38)	(37)	(36)	(31)	(30)	(28)	(27)
12	Drawdown						(1,000)								
13	Redemption										1,000				
14	Interest Payment	=-D11	41	40	39	38	37	40	38	37	36	31	30	28	27
15	Capital Payment	=PPMT($B$4,1,D8,D10,0)	258	260	261	262	263	294	295	297	298	264	265	266	267
16	C/F	=SUM(D10:D15)	(9,742)	(9,482)	(9,221)	(8,960)	(9,697)	(9,403)	(9,108)	(8,811)	(7,513)	(7,250)	(6,985)	(6,719)	(6,452)
17															
18	Total payment	=SUM(D14:D15)	299	299	299	299	299	334	334	334	334	294	294	294	294
19															

There are a range of other financial functions available in Excel and typically aimed at financial analysts in a capital markets setting. These are beyond of the scope of this guideline.

10.4 Time-weighted interest

Overdraft interest is most accurately modelled by reference to the average overdraft balance during a period, as opposed to just the brought forward or carried forward value. A circular reference will normally be produced if a simple average is taken between the opening and closing balance, since the closing balance is often contingent upon interest (see section 8.5.1).

A possible solution is instead to calculate the average value over a period as the opening value, plus time weighted adjustments for cash flows in the period. For example, if revenues accrue evenly during the year, give them a 50% weighting, similarly for costs. Repayments on the final day of the year might attract a 0% weighting as they have no impact on cash during the year from the perspective of calculating interest. By the same token, equity injected at the start of the year would have a 100% weighting. The weighted flows should be added to the opening balance and then multiplied by the interest rate to give interest for that period – without causing circular references.

10.5 Working Capital

Almost every model will require a set of working capital calculations and quite often the same fundamental drivers will be applied. For example, trade debtors to be driven from a sales figure, VAT rate, and trade debtors' assumption. (This example will exclude bad debt for simplicity.)

There are two different approaches which will yield the same results for a trade debtors working, if programmed correctly, although they differ in approach and ease of coding.

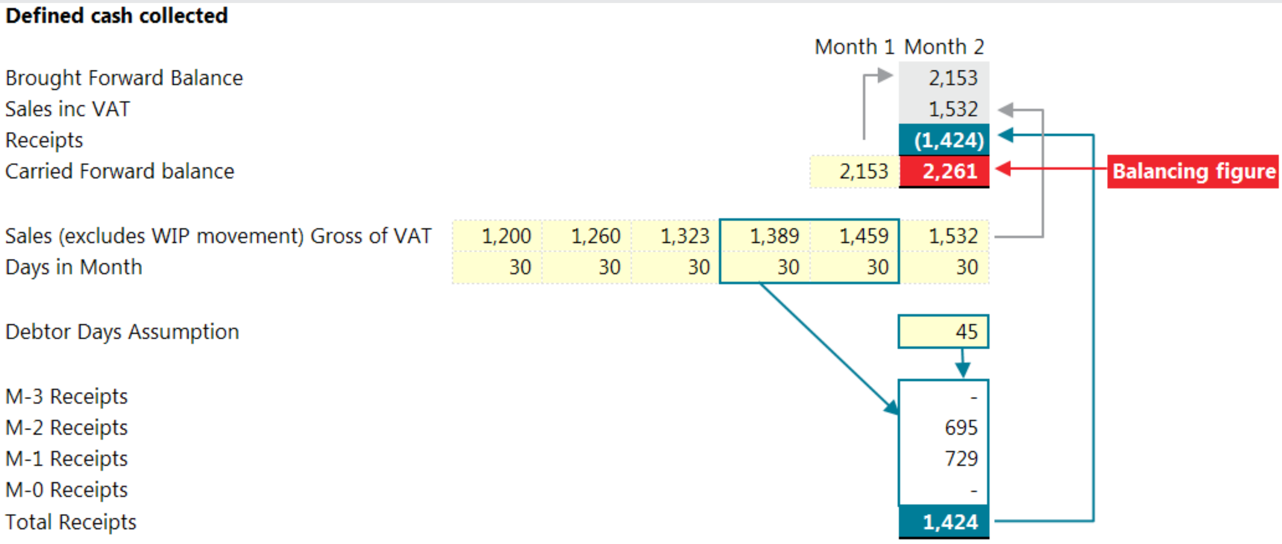
10.5.1 Defined cash collected

Defined cash is the simplest conceptually. Under this method the brought forward balance is simply the closing value from last month, the increase is the current month invoicing plus VAT, and the cash received is independently calculated, with the carried forward balance being the sum of the above.

The cash received in a month is calculated as the invoiced figure from X days ago (plus VAT), where X is the debtor days assumption. For example, if the assumption is 60 days, and the model periods are 30 days, then it is the invoicing and VAT from two months ago. This is also the easiest to code when the debtors days assumption is an integer number of model periods, and unchanged across the model timeline.

It can be coded by linking the cash received value in a month to the corresponding prior months' VAT inclusive invoiced figure. (Either directly linked, or by an INDEX or OFFSET formula, which will 'look back' the appropriate number of columns.)

Figure 24: Defined cash collected method



This approach can be made to work with non-integer debtor days assumptions by splitting the 'look back' across adjacent periods. For example, if the assumption was 45 days, in a monthly model simplified to assume 30 days per month, then the cash receipt would be 50% of invoices plus VAT from one month ago, and 50% of the value from two months ago.

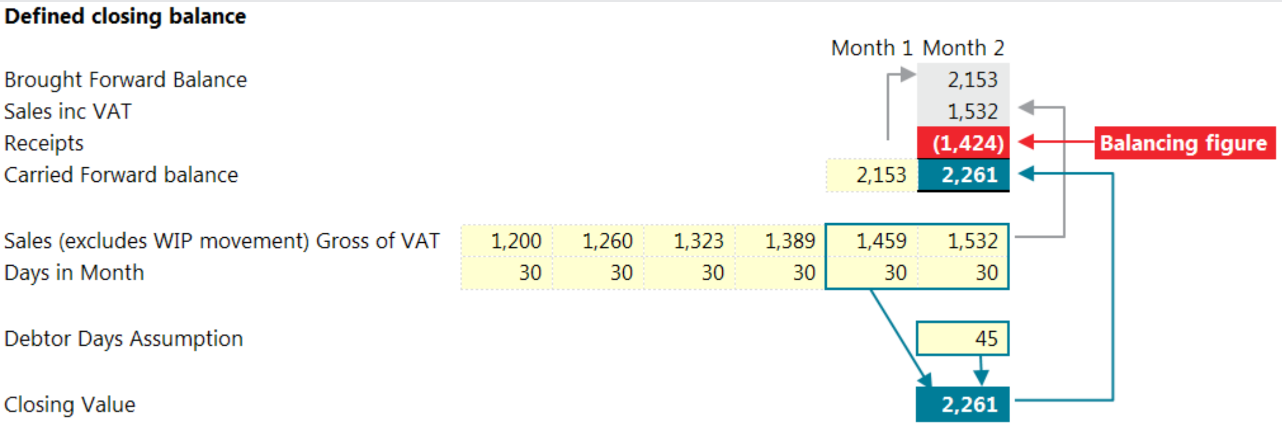
Disadvantages of this approach are that it does not lend itself well to varying the assumption over the timeline, because it can easily result in cash from invoices in a given period being collected in more than one future period, or not at all. It also does not deal with a misaligned opening balance (ie, an actual opening balance which happens to be inconsistent with the stated assumption), rather it balances increases and decreases in the debtor value relative to the previous month, and does not address the overall quantum. As such it is common that when trade debtor days are calculated from the debtor balance, they are shown to be different (usually higher) than the stated assumption.

10.5.2 Defined closing balance

Defined closing balance is slightly more complex conceptually. The difference compared to defined cash is that in this method the closing balance is defined and cash received is the balancing figure required to reconcile to the opening position plus current period invoicing.

The calculation of the closing balance is done against invoices on a count back basis. For example, if the assumption is 45 days, then the closing balance is assumed to comprise all the sales in the current month (ie, 30 days), plus invoices from the last 15 days of the previous month. Using this mechanism of calculating the closing balance as a proportion of invoicing from several previous periods, allows this method to use non-integer periods for the debtor days assumption.

Figure 25: Defined closing balance method



The formula for this is more complicated than for the defined cash method, but the defined closing balance has several advantages which justify the additional coding effort. For example, the closing balance on the balance sheet will always reflect the debtor days assumption more precisely.

This method also deals with a misaligned opening balance; because it is always aligned to the assumption in period one of the forecast period. In such cases it is likely that if the forecast assumption is different from the actual historic reality, then the transition in the model will likely span a couple of periods, thereby necessitating a change in debtors days assumption over time, which is something this method handles as well.

10.6 Indexation

10.6.1 Definitions

In modelling terms, inflation and indexation are sometimes used interchangeably. Inflation is the term used by economists to describe a general increase in prices and fall in the purchasing value of money. Indexation is a technique to adjust income payments by means of a price index, in order to maintain the purchasing power of the public after inflation.

In economics, a nominal value is an economic value reflecting the impact of general price level changes over time. From a forecasting perspective, a nominal price is the expected actual price of a good or service, inclusive of the expected changes to the general price level (ie, inflation), at a future date.

10.6.2 How to apply indexation

The process of converting between nominal and real terms is known as inflation adjustment. Real prices is the term used to describe the value or values obtained by removing the effect of general price level changes from the nominal value, which can give a more meaningful picture of economic trends. Conversely a real value would be adjusted (typically uplifted) by an index or inflation factor to derive nominal values.

For short-term models in the two to five year range, separately and specifically identifiable indexation or inflation may not be a material concern. However, for longer-term projects in the range five to fifteen years and above, the effects of indexation are likely to be much greater and should be explicitly separated out and considered in the financial model specification.

10.6.3 How it is used and why

If there are factors in the model which will be affected differently by indexation then it becomes particularly important to include an indexation calculation within the model. A common example would be in the case of a long term contract where nominal rates are agreed well in advance. In such circumstances there is a risk that if inflation runs at a higher rate than that which was used to set the nominal contract value, then the real value of income falls. This is a problem because if revenues are only growing at the fixed 2% but inflation and costs grow at 4% then the profitability of the project is reduced.

10.6.4 Practical guidance

A typical approach to modelling for indexation is to allow the user to input either real or nominal figures into the assumptions sheet, along with populating a switch to indicate which basis has been selected. In the case of nominal figures being entered, no indexation calculation will be required, because the indexation of the values has been 'baked into' the assumptions. On the other hand, if the inputs are real values then they must be inflated at an appropriate rate so that the model will include them at their nominal value.

These mechanics can be achieved by using two sequential cost workings; the real costs are pulled through from the assumptions sheet, and then a compounded indexation rate applied to them, as they are translated down into the second nominal cost working to determine nominal values. The rest of the modelling is as standard. This method can accommodate inputs of nominal values, because in these cases the modeller will also set the 'apply indexation' flag to false, and therefore no inflation is added, and the values are translated directly into the second cost working unadjusted.

If there is ever a need to translate nominal calculations into real outputs, then the same principles apply; ie, to calculate a deflation factor and replicate the calculation block using that conversion. This technique may be applicable if the model needs to report real figures for a public sector procurement project.

One critical issue is to be internally consistent with the discount rates applied to a given set of cash flows, for example nominal cash flows should be discounted with a nominal discount rate, and real cash flows with a real discount rate. Mistakes in the treatment of real and nominal cash flows have been at the heart of some of the more serious modelling errors to emerge in recent years. We cannot emphasise enough the importance of getting this right and of reviewing this area closely during any assurance process.

10.7 Depreciation

Depreciation calculations within financial models can be a difficult area to get right. However there are a few standard approaches which can be applied to cover off most requirements within typical financial model builds.

Depreciation policies will, of course, vary across asset classes, but they generally fall into the broad categories of reducing balance, or straight line. Other less used options might be a 'defined value,' where the depreciation value on a monthly basis is typed in as a hard-coded assumption, or calculated by reference to some other cost, for example, expressed as a ratio of CAPEX, or revenue, in a simple investment analysis model. In the case of financial assets an annuity-style depreciation method may be applicable. This guideline covers some simple, practical, modelling techniques which are applicable to the most common reducing balance and straight line methods.

10.7.1 Reducing balance

Modelling for reducing balance depreciation is relatively straightforward in so far as the fixed asset value will increase with purchases, decrease by the net book value of disposals, and depreciate over time by a fixed proportion each month.

This simplified approach to calculation means that the associated calculations are comparatively straightforward; the monthly depreciation charge is calculated by applying a policy assumption to the brought forward net book value in that month.

This approach will automatically prevent one of the more common depreciation modelling errors, that being the 'over depreciation' of assets, such that a negative value is presented on the balance sheet, since as soon as the net book value (NBV) closes in to zero, so will the calculated charge. Although the depreciation charge calculation only requires the carrying NBV, it is still necessary to keep a record of the carrying cost and accumulated depreciation for the purposes of proper disclosure in the financial outputs.

10.7.2 Straight line

Straight line depreciation may appear straightforward to calculate, since the concept is that each asset is simply written down at a fixed rate each month until it is fully depreciated, and indeed these calculations are straight forward when calculated on a per-asset basis on a fixed asset register.

However, the modelling complexity arises when a financial forecast model does not model fixed assets on a per-asset basis, but rather on an asset class basis; which includes a mix of assets of different ages. Therefore within the opening cost, and accumulated depreciation balances, there will be assets at different stages of depreciation.

Assets which have been owned for a long time will have become fully depreciated, and therefore no further depreciation charge should be made in respect of them, others may be near the end of their useful economic life and will therefore reach nil NBV soon into the forecast period, and others will be nearly new and require writing off over the full depreciation period. These differentials make the modelling of straight line charges potentially complex. The two methods set out below will explain a simplified and a more sophisticated method of modelling these scenarios.

Opening non-fully depreciated assets — simplified

A simplified approach to resolving this problem is to take the following model inputs:

- cost of fully depreciated assets;
- cost of non-fully depreciated assets;
- accumulated depreciation of fully depreciated assets (which by definition is equal to their cost); and
- accumulated depreciation of non-fully depreciated assets.

The simplified modelling approach is then to treat the non-fully depreciated assets as a single cohort and depreciate the cost straight line until they become fully depreciated. Ie, no distinction is made for assets of different ages within the opening non-fully depreciated balance, and it is effectively treated as a single purchase.

Opening non-fully depreciated assets — detailed

To model the opening balances in a more sophisticated way it is necessary to capture information about the ageing of the assets within the opening balance. This information will be available from the company's fixed asset register. Inputs to the model will be:

- cost of fully depreciated assets;
- cost of non-fully depreciated assets;
- accumulated depreciation of fully depreciated assets (which by definition is equal to their cost);
- accumulated depreciation of non-fully depreciated asset; and either
- timeline of depreciation per month on assets in the opening balance; or
- analysis of opening balance into aged cohorts.

Hint

MAX(calc, 0) and MIN(calc, 0) functions are useful in ensuring the that depreciation charge calculated is:

i) never a negative charge, and ii) never exceeds the available NBV, which would cause the NBV to go negative.

If a timeline of depreciation is provided, this will have been calculated on the basis of the fixed asset register. The total depreciation on this timeline will equal the opening NBV of the brought forward assets, and it will fully account for their depreciation over time. The values must be supplied as a time series of data, as opposed to a single value, because the monthly charge will drop periodically as assets within the opening balance each become fully depreciated and no longer contribute to the charge.

If an analysis of opening balance into aged cohorts is provided, then the timeline discussed above can be derived by calculating the monthly charge for each cohort, based on the purchase price, and then applying it in the forecast for the appropriate amount of time, until that cohort is fully depreciated.

The depreciation value in the model will comprise the above amounts in relation to the opening balances, as well as depreciation due on assets purchased during the forecast period.

Assets purchased during the forecast period

In these cases the model will capture the value spent on fixed assets and the timing of that spend, therefore the use of an OFFSET, SUM and INDEX, or SUMIF function can be used to easily sum purchases in the previous X months, and multiply that value by the depreciation rate to calculate the charge in a given month. (For example, if a depreciation policy is straight-line over two years, then the offset function should add up purchases in the 24 months prior, and divide by 24 to get the depreciation charge in that month.)

Figure 26 below illustrates how this could work in practice using an OFFSET function. The key formula is row 21 which uses an OFFSET() function to determine the address of cells which include the last 24 months' worth of purchases, from which the depreciation on purchases can be calculated by simply dividing by 24. The formula has been made more complicated in this example by including a MAX/MIN function to determine how far left the range called up by the OFFSET() can begin, such that it remains within the confines on the sheet (ie, doesn't go further left than column A), and therefore how many periods it should include.

Figure 26: Depreciation example – accounting for asset acquisitions using the OFFSET function

1	**Depreciation example - all purchases within the model timeline**									
2										
3	**Straight Line Depreciation Period (years)**			2						
4	**Straight Line Depreciation Period (months)**			24						
5										
6	**Model period**	1	2	3	4	5	6	7	8	9
7										
8	**Cost**									
9	b/f		12,000	12,000	12,000	12,000	12,000	12,000	12,000	-
10	Acquisitions	12,000								
11	Disposals								(12,000)	
12	c/f	12,000	12,000	12,000	12,000	12,000	12,000	12,000	-	-
13										
14	**Depreciation**									
15	b/f		(500)	(1,000)	(1,500)	(2,000)	(2,500)	(3,000)	(3,500)	-
16	Depreciation Charge	(500)	(500)	(500)	(500)	(500)	(500)	(500)	-	-
17	Disposals								3,500	
18	c/f	(500)	(1,000)	(1,500)	(2,000)	(2,500)	(3,000)	(3,500)	-	-
19										
20	**Depreciation charge calculation**									
21	Value purchased in last X months	12,000	12,000	12,000	12,000	12,000	12,000	12,000	-	-
22	Charge	500	500	500	500	500	500	500	-	-
23										
24	**Net Book Value**	**11,500**	**11,000**	**10,500**	**10,000**	**9,500**	**9,000**	**8,500**	**-**	**-**

Assets disposed of during the forecast period

Calculations relating to the disposal of assets during the forecast period will depend upon which modelling approach has been applied, and whether the disposed assets were purchased during the forecast period or were part of the opening balances. There are a number of approaches available, but the number of options and complexity of such calculations puts them outside of the scope of this best-practice guideline.

10.8 Data manipulation

Generally, a reporting/analysis model will not require a great deal of manipulation, but it can play a very important role in certain circumstances. For example, model suites that support rail franchise bids require numerous complex supporting models to crunch data, which then feed into the core model, therefore requiring various data manipulation techniques.

Excel's grid is not always the ideal tool for data manipulation, indeed the 'get' and 'transform' features added to the data model in Excel 2016 onwards make light work of the types of manipulation which would otherwise be time-consuming to perform on a worksheet.

However, we recognise that many readers of this guideline will be using previous versions of the software, or would be unfamiliar with the process of manipulating data in the Excel data model. Below are a number of potentially useful Excel worksheet functions which can be used to manipulate data.

- **IF()** – A very basic function but helpful in data manipulation as it allows differential treatment of imported data rows based on certain criteria.
- **INDEX(MATCH())** – Performs a similar function to VLOOKUP() but with the added resilience to inserted columns and ability to have the lookup column to the right of the data being returned. This combination can be used to cleanse data. For example, to cleanse a field on an incoming data table it is possible to extract the relevant column to a new sheet, remove duplicates and then manually assign a clean 'mapped' value to each item. Using an INDEX(MATCH()) function you can return the cleansed value to the source data table, ready for using in pivot tables and other reports.
- **OFFSET()** – Similar in function to the INDEX() function, with the key differences being that OFFSET() can move in a negative direction (up or left) from a reference point, and OFFSET() does not require the source data range to be defined, rather the target will move as far as required from the reference point. Useful where a periodic inflation rate is to be applied, this function can be used to retrieve the value from a given number of periods previous. However this function should be avoided unless absolutely necessary because of the difficulty in tracing precedents and auditing such formulae.
- **INDIRECT()** – This is an advanced feature of Excel and essentially allows the user to construct a formula reference, using a formula. However, as with OFFSET(), this function should be used with caution because of the difficulty in tracing precedents and auditing such formulae. It can be useful for creating fully locked references where a normally anchored formula is not enough, for example to return a value from a given cell reference regardless of row or column insertions or deletions. It can also help when pulling data from different worksheets, where the sheet selected is an input variable.
- **TRANSPOSE()** – This multi-cell array formula takes a one or two-dimensional array as an input and transposes it. Helpful when source data is input in a columnar fashion, but for calculations it will be easier to work with by row, or vice versa.
- **SEARCH()** – Function scans a text string and returns the position of a required sub string. Useful when combined with LEFT(), RIGHT() and MID() functions.
- **LEFT(), RIGHT(), MID()** – Returns a sub-string of a text string, either the first X characters, the last X characters, or X characters from the middle, starting at position Y.

When used with the SEARCH() function these can be used to split individual text strings into sub-strings.

- **YEAR(), MONTH(), DAY()** – Each function will take a date as an input and return the component part as an integer year, month (1 to 12), or day (1 to 31). Useful to take a serial date and extract period information.
- **DATE()** – Takes year, month and day integers as inputs and returns the serial date which corresponds to that date. Useful to construct a serial date which Excel understands for use in pivot tables, etc.
- **EOMONTH()** – Returns the last day of the calendar month for a given input date, useful for taking source data tagged by time or day and aggregating at a month level for reporting. Also useful when creating model timelines.
- **WEEKDAY()** – Returns an integer value corresponding to the day of the week of a serial date. Several options are available for convention, 0 to 6, or 1 to 7, and with any weekday being considered the 'start' of the week.
- **WEEKNUM()** – Returns the week number in the year of a serial date, which can be helpful to mapping data to timelines in weekly models.
- **DATEDIF()** – This undocumented feature of Excel returns the difference between two dates, either as complete years, months, or days; or as 'days ignoring months,' 'days ignoring years', or 'months ignoring years'. This function can be helpful when calculating the number of complete years or months between dates. Whereas, subtracting dates from each other will only provide the number of days; and other techniques using, for example, Excel's MONTH() or YEAR() functions are more complicated to implement.
- **IF(COUNTIF())** – By combing a COUNTIF function with an IF function it is possible to check a value on a source data row against a list of criteria. If the value is found on the list, then the count will be non-zero and a certain action taken by the IF statement; and if the value is not found on the list, then the count is zero and the other action is taken by the IF statement. In short, it can be used to check if a given value is on a list or not, and take action accordingly.
- **SUBSTITUTE()** – Substitutes target text with replacement text within a text string, and can be useful in recoding imported data in a more efficient manner than looking up to a cleansing table.
- **TRIM()** – Removes leading and trailing space from text.
- **CLEAN()** – Removes non-printable characters from text, for example line breaks.
- **SUMPRODUCT()** – Takes several identical size ranges as inputs, and multiplies corresponding elements of each array, and sums the results. Useful for allocating costs amongst divisions by reference to a defined proportional split.
- **Semi-anchored ranges** – The concept is that a formula can reference a range where only one point is anchored. For example, =MAX($C$1:$C10). Here the top of the sum range is anchored to row one, but the bottom of the range is not anchored, so the range will grow as it is copied down a column. This can be combined with a helper column. For example, to index each unique value in a list, you could use an IF statement to return zero, if the helper column is zero, and MAX($C$1:$C10)+1 if the helper column is one. This function looks for the max index value which appears in the list so far, and increments by one. An example of the use of semi-anchored ranges is shown in Figure 27, overleaf.

Figure 27: Semi-anchored ranges

SUM | =SUM($H$10:M10)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1				Month		Units		Jan	Feb	Mar	Apr	May	Jun	
2														
3		Revenue												
4														
5			Division A											
6														
7				Sales Volume		000s		1,000	1,100	1,150	1,250	1,300	1,400	
8				Unit price		£		45	48	51	51	52	52	
9														
10				Revenues		£'000s		45,000	52,800	58,650	63,750	67,600	72,800	
11				Revenues to date		£'000s		45,000	97,800	156,450	220,200	287,800	=SUM($H$10:M10)	
12														

10.9 Excel versions 2013 and 2016

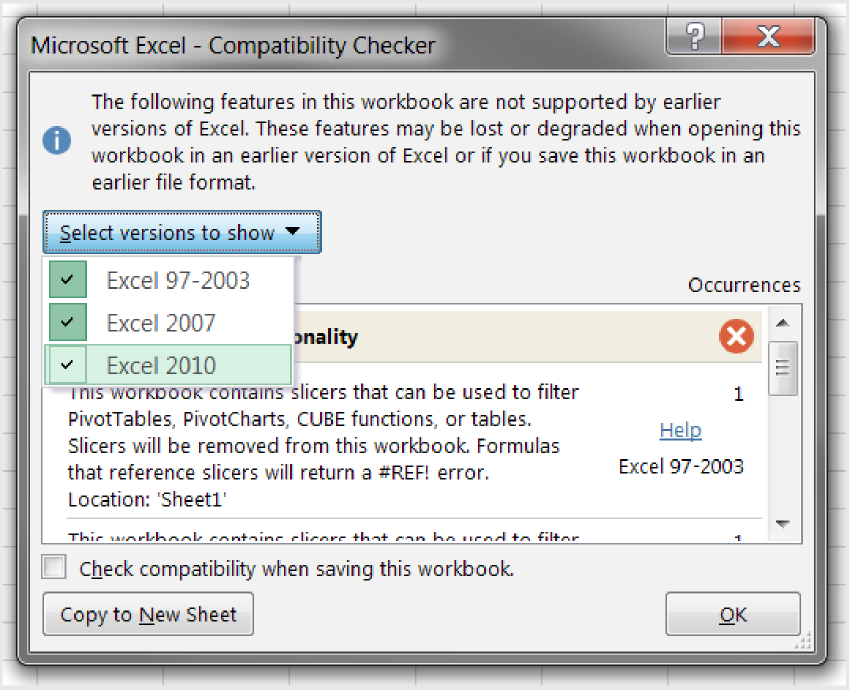
It is important for modellers to accommodate the requirements of their audience and to provide them with models which adhere to modelling best practice and, at a practical level, which can be opened, calculated, and used by other stakeholders. It is not particularly helpful when a model which has been produced is incapable of being opened by the reviewer or client.

While we immediately look to file size and calculation time as benchmarks of compatibility with other people's hardware, there is also the issue of software compatibility to bear in mind. While it is unlikely that in 2016 you will come across many people still using Excel 2003, it should be remembered that not all features of the recent 'big grid' versions of Excel are backwards compatible as far as Excel 2007. For example, 'sparkline,' a miniature chart in a cell, is available in Excel 2010 onwards, but not in Excel 2007.

Careful consideration should be made when choosing to make use of the added functionality of the latest versions of Excel. Usually this can be established at the planning phase of the work, where an agreed version or standard can be selected. When working using a later version of Excel than the agreed standard it can be helpful to use 'compatibility checker,' (click 'file,' then 'info,' then 'compatibility checker' from the 'inspect workbook' menu). This tool will allow checking of compatibility with various previous versions of Excel, so that appropriate remedial action can be taken by the user if required.

Bearing the above in mind, there are some very interesting new features which have been added in Excel 2013 and Excel 2016, notably in the areas of data processing, data visualisation, and collaboration, which may be of benefit if the model brief allows. Appendix B includes a brief overview of some of those key features introduced in the latest versions of Excel.

Figure 28: Compatibility checker



10.10 Key points

- Make use of PPMT to allow flexibility in loan modelling.
- Avoid circular references by using time-weighted calculations for interest.
- Consider appropriateness of defined cash compared to defined balance in modelling working capital balances.
- Indexation calculations are vital to get right for evaluating long-term projects.
- Consider the modelling requirements carefully with respect to depreciation, avoid the risk of over-complication through careful planning.
- Make use of data manipulation functions to cleanse imported data.
- Avoid using INDIRECT() and OFFSET() functions unless it is unavoidable, due to the reduction in transparency such formulae present.
- Use 'compatibility checker' to ensure that the model will be accessible to other stakeholders.

11 Communication and presentation

11.1 Overview

This chapter reviews some practical presentation and communication tools, considering:

- the importance of a model as a communication tool;
- data tables;
- charts;
- format/style;
- conditional formatting; and
- print set-up.

11.2 Context

One of the primary roles of a deal model is to communicate financial information and decision parameters to relevant stakeholders so it is not just the content that is important but how that content is presented.

The presentation of content is relevant for two main audiences: users of the model; and the model sponsors. For the former, they need to know how to find and change assumptions and information within the model and how to manipulate key model settings (for example sensitivities and scenarios). For both groups, but particularly the latter, they need the information and financial results presented clearly, relevantly, and succinctly, and the model design should support this.

- **clearly:** the information needs to be readable and transparent, with dates and descriptions to aid understanding.
- **relevantly:** the information needs to be relevant to the decision process and should be presented in a meaningful context (for example, with prior year comparatives).
- **succinctly:** the information that is needed and nothing more, and at a level of detail appropriate to the timeframes.

This chapter considers some of the tools and techniques available within Excel to help achieve these objectives.

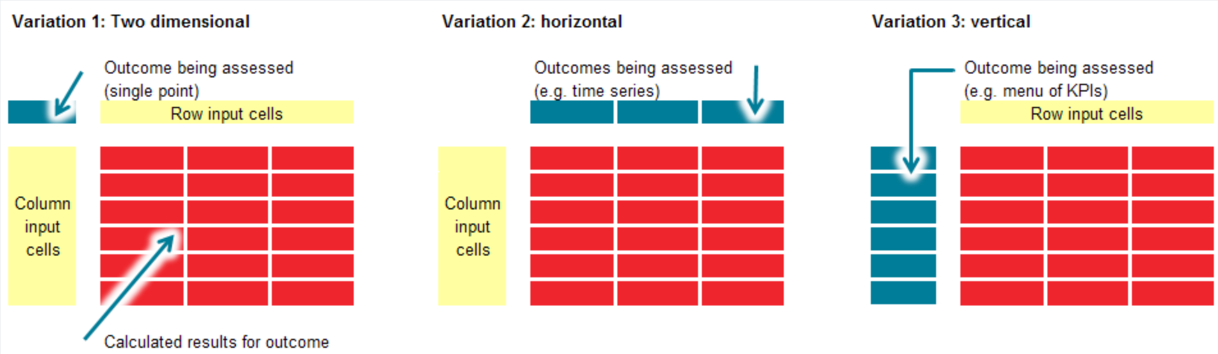
11.3 Data tables

In a deal context, there is often a need to compare and present the results from multiple scenarios in parallel. Data tables allow simple sensitivity results for key financial indicators to be presented and flexed side by side, live within the model. This avoids the need either to paste and store hard-coded results, or to run multiple separate linked model versions; both of which approaches are prone to error and create version control and model management risks.

Data tables are relatively simple to set up and use. Lay out your data and calculations as illustrated in Figure 29 then select 'data table' from the 'what if?' analysis menu under the data tab on the ribbon. The dialogue box simply prompts for the input cell whose value you want to vary, and whose impact you therefore want to assess, (the yellow blocks show the relevant assumption range that will be linked to the target input cell or cells).

Figure 30 shows an example of a data table that flexes two variables (in this case, exit year and exit multiple), and presents the internal rate of return (IRR) under each combination. There are 30 possible permutations, which would otherwise require 30 model runs.

Figure 29: Data table structure and layout



Data tables can also be combined with conditional formatting to provide strong visual cues when combinations of input variables generate a result outside expected or desired parameters. These are also illustrated in Figure 30, which shows both the native Excel heat map formatting (on the left), and a bespoke 'cross-hairs' type formatting (on the right), highlighting the results under the base assumptions.

Figure 30: Illustrations of data table and conditional formatting

Exit multiple: 6
Year: 5

Year	Exit multiple 4	5	6	7	8
2	-14.37%	-4.26%	4.88%	13.28%	21.11%
3	-6.91%	0.28%	6.56%	12.18%	17.28%
4	-2.94%	2.62%	7.41%	11.63%	15.42%
5	-0.48%	4.06%	7.92%	11.30%	14.31%
6	1.19%	5.03%	8.27%	11.08%	13.58%
7	2.41%	5.72%	8.51%	10.93%	13.06%

Year	Exit multiple 4	5	6	7	8
2	-14.37%	-4.26%	4.88%	13.28%	21.11%
3	-6.91%	0.28%	6.56%	12.18%	17.28%
4	-2.94%	2.62%	7.41%	11.63%	15.42%
5	-0.48%	4.06%	7.92%	11.30%	14.31%
6	1.19%	5.03%	8.27%	11.08%	13.58%
7	2.41%	5.72%	8.51%	10.93%	13.06%

At face value, since data tables operate in a maximum of two dimensions, Excel only allows you to:

- analyse a single point KPI for the impact of variations in two variables (variation one, in Figure 29); or
- analyse a series of KPIs (typically a time series) for the impact of variations in a single variable (variations two, and three, in Figure 29).

However, if an input variable is linked to a cell that determines the selection of scenarios under which themselves multiple other variables are adjusted (rather than simply being linked to a single 'point' input value), then data tables can be used to enable dynamic comparison of results for quite detailed individual scenarios (see also section 13.4).

This can be particularly useful if you have to compare and chart, say, gross profit, under different cases and sensitivities (see the chart shown to the right in Figure 33 for a simple example).

The major limitation of data tables is that for larger, more complex models, they can add significantly to the processing time. This is because by default Excel tries to recalculate the whole model for each and every permutation of variables each time the base model recalculates. The good news is that this setting can be temporarily disabled until needed under the 'formulas tab' (under 'calculation options', select 'automatic except for data tables').

The other limitation is that the input variables to which data tables link have to be on the same worksheet. Using data tables to their fullest often requires some deliberate breaches of some of the best-practice principles, but as indicated earlier, once the rules are understood, they can be stretched in a managed and controlled way.

Data tables are not a universal solution. For more complex applications and models, and where a modeller has good VBA literacy, a simple macro routine can be written to achieve the same outcome.

11.4 Pivot tables

Excel provides pivot tables to facilitate analysis and presentation of large data sets. We mention these in passing, because readers may be familiar with, or aware of, them. In our experience, they are rarely relevant in a transaction modelling context, except perhaps as part of the underlying 'feeder' analysis carried out by data owners.

While pivot tables can be very powerful and useful tools for analysing data, this capability is rarely needed in a deal context. The dynamic structure, layout limitations, and formatting constraints of pivot tables make them generally unsuited for time-based transaction modelling and we mention them here purely for completeness.

11.5 Charts

Excel provides an extensive range of charts that can be used to show data and results under the 'insert' tab on the ribbon. We do not explore them all in detail here but we make a few practical points:

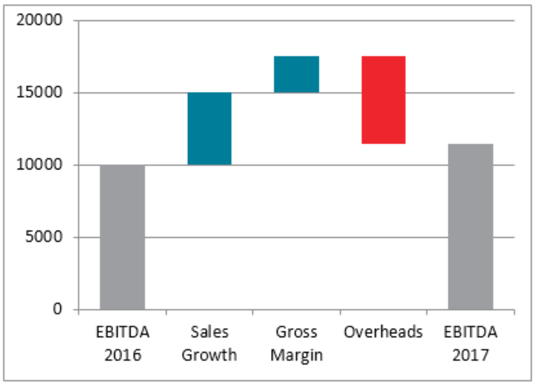
- Excel's default chart settings are sometimes clumsy in use and may need to be adapted to specific corporate branding/style guides, or be sized to suit different document types. Editing chart formats and settings in Excel can be time-consuming, and achieving consistency difficult, particularly if you are trying to achieve the same look over multiple charts. Accordingly, once you have established the colours, format, and settings for a chart, we recommend you use the ability within Excel to 'save [your chart] as a template,' enabling you to reselect and apply that style multiple times.
- Using charts to communicate results carries the same issues as numeric data – they need to be clear, relevant, and focused on the key outputs. Take great care to select a chart type suitable to the data and which presents information in a meaningful and not a misleading way. Stephen Few's book *Information Dashboard Design* provides some excellent guidance in this regard.
- Chart titles and axes titles can be linked to cells within a model, so that they can be automatically updated when the linked cells change. This can be helpful when used in conjunction with data tables and scenarios.
- Charts can also be augmented using Excel's 'camera' tool, to take a snapshot of results or inputs elsewhere in a model, which can then be overlaid as a linked 'live' picture on the chart area.
- Bridge charts, also known as waterfalls, have not naturally been supported by Excel until Excel 2016, but they are frequently a useful tool in displaying variances across model scenarios, or comparing financial performance between years (for example, preparing EBITDA bridges in a deal context). Creating them in earlier versions of Excel can be achieved using 'blank' bars and some (relatively) simple programming to calculate the height and direction of the bridge bars. Figure 31, below, illustrates the basic principle, we will let the interested reader work out the coding. Alternatively, upgrading to Excel 2016 provides this function natively.

Tip

Readers with Excel 2016 will be able to use a new feature called 'waterfall charts', which are much simpler to create and explored further in Appendix B, along with other new useful features.

Figure 31: Illustration of bridge chart

	A	B	C	D	E	F
1		TOTAL	BASE	BLANK	BLUE	RED
2	EBITDA 2016	10,000	10,000			
3	Revenue Growth	15,000		10,000	5,000	
4	Gross Margin	17,500		15,000	2,500	
5	Overheads	11,500		11,500		6,000
6	EBITDA 2017	11,500	11,500			



In addition to the main charts palette in Excel, versions of Excel from 2010 onward allow the user to use 'sparklines'. These are miniature charts generated and held at cell level within the worksheet, with specific formatting designed to highlight exceptions and trends. They can be very helpful in developing informative and easy-to-read dashboards. They can also have a role as a simple quality assurance tool when used alongside row totals in the calculations and output sheets, to help highlight unusual movements or peaks and troughs along a given row of calculations. One word of caution, adjacent sparklines may not use vertical axes of a consistent scale and so care must be taken interpreting them.

Figure 32: Illustration of sparklines as a dashboard output

	A–B	C	D	E	F	G	H	I
1	**Income Statement**	**2016**	**2017**	**2018**	**2019**	**2020**	**2021**	**Trend**
2		**Act**	**Proj**	**Proj**	**Proj**	**Proj**	**Proj**	
3	Revenue	204,401	235,618	205,785	237,900	269,661	306,439	
4	Cost of sales	156,109	176,494	159,801	187,964	218,813	241,878	
5	**Gross Profit**	**48,292**	**59,123**	**45,984**	**49,936**	**50,848**	**64,561**	

Other visualisation charts released in more recent versions of Excel include 'treemaps,' 'sunbursts,' 'radars,' 'powermaps,' and '3D maps,' which can be linked with time series data to provide a time-lapse playback of the progression of, for example, regional sales.

11.6 Formats/styles

We talked earlier about the importance of how a model communicates with users, stakeholders and reviewers. It greatly assists the user if the function or purpose of a cell can (in part at least) be divined by its context and format. It is helpful to colour code categories of cells to provide a simple visual key (indeed we would usually recommend setting out a 'key' on the user guide/information worksheets). The aspects that we view as most helpful to highlight are:

- input cells (distinguishing between 'vanilla' inputs and inputs linked to a scenario page or dashboard);
- call-ups of inputs on calculation sheets;
- error checks; and
- section and subsection headers etc.

In addition, some advocate highlighting blocks of contra-flow calculations and calculation results that are themselves called-up.

The only note of caution is not to go over the top: too much formatting/colour can be distracting and it can also become high maintenance and contribute to memory issues within Excel, particularly heavy use of conditional formatting and styles.

Using the 'styles' functionality will allow the modeller to achieve some efficiencies through creating a library of formats and we would recommend setting up the styles and formats at the start of the modelling exercise, formatting as you go. The big benefit of using styles (available from Excel 2007 onwards) is that once the style is set-up; any changes to the master style are automatically implemented to all cells to which that style has been applied.

11.7 Conditional formatting

Excel provides increasingly powerful conditional formatting tools. These tools allow the modeller to assign different formats according to the result or value shown in the relevant cell. Their value lies in enabling the modeller to set conditions that highlight specific cells (through different formatting) when outputs or results fall inside or outside specific bounds. Practical applications include:

- Combining conditional formatting with data tables (as illustrated earlier), to show, for example, which combinations of inputs result in returns above a target threshold, or to highlight the base case result within a grid of sensitivities.
- Using conditional formatting with scenario tables to highlight which data set is active, or indeed to highlight variations in other data sets from the base case data. This is useful to help orient the model user.
- Highlighting error checks, for example flagging balance sheet errors so that the user can quickly identify any periods in which tests are breached.
- Generating 'heat maps' (or simple embedded bar charts) of data, to highlight relative areas of difference (see Figure 30).
- Establishing timeline-related formatting, specifically allowing the formatting of a cell to be linked to the relevant timeline period. Eg, highlighting the year in which a deal is transacted, or differentiating between historical and forecast periods. The benefit of using conditional formatting for this is that (properly executed) the formatting will remain in place when formulae are updated and copied across a row. Whereas, a manually overlaid (and by extension non-conditional) format would be wiped out and replaced with the leftmost column's format in a given block when cells were filled or copied across.

As you might imagine, conditional formatting is very powerful. The risks are that in versions of Excel from 2007 onward, the number and implementation of rules can run out of hand, become difficult to manage and maintain, and can lead to high processor and memory usage. (Earlier versions of Excel restricted the user to three rules which were applied in a strict hierarchy.)

Poorly constructed rules relying on hard-coded values can also fall out of step with the model and draw the users' attention incorrectly. In a similar vein, for more complex uses of conditional formatting, you risk setting up a kind of model within the model and the logic for the formatting tests is embedded implicitly in the cells and may not be immediately obvious to a user. In practice, conditional formatting does not interfere (yet) with the calculation flow, but it can lead to misleading presentation of results if it is not carefully managed/applied. The usual caveat applies: proceed with a measure of caution.

11.8 Print areas

From the point of view of user friendliness, it is beneficial to programme in and preset print areas early on in the use of a model. Significant time can be wasted by multiple parallel users each having to fiddle around with print settings each time a new version of the model is released. This may seem a small point but it can save a lot of time, frustration, and wasted effort.

One weakness of Excel is that you cannot adjust print area or print title rows and columns for multiple worksheets as a batch. Accordingly, we recommend at least setting up print area widths and boundaries, then replicating this template as the starting point for any relevant worksheet.

As a practical note, it is rare that all worksheets within a given model will need to be printed. Core calculation sheets are a good example of this. The requirements in this regard should be considered when drawing up the model specification, with the relevant sheets (typically inputs, outputs, and dashboards) pre-configured as recommended above.

11.9 Key points

- Use data tables to analyse KPIs under different inputs and combinations of inputs without proliferating model variants.
- Link data tables to scenarios to quickly review key results of different model cases concurrently.
- Create, format, and store custom chart templates to save time.
- Use clear formatting to identify the functional purpose of cells and groups of cells. Implement formatting consistently.
- Use conditional formatting to highlight anomalies in results, error messages, and active data. Be careful not to let conditional formatting proliferate.
- Set up print areas and headers early in the modelling process on template sheets to avoid multiple reprogramming and resetting of print settings.

12 Dashboards

12.1 Overview

This chapter considers the key aspects of dashboards within financial models, including:

- the purpose of a dashboard;
- the functions a dashboard may fulfil; and
- ideas and techniques for construction.

12.2 Context

To get best value out of a financial model in a transactional context, it is generally essential to have a form of executive summary – the 'dashboard' (or in some quarters the 'cockpit'). Much as in the automotive context, the purpose of the dashboard is to:

- provide a control interface: allowing limited flexing of key decision variables;
- give real-time feedback on what happens when you change control inputs: presenting summary outputs and KPIs; and
- generate warnings when things break down: containing error flags and checks.

To fulfil these various functions, the dashboard has to contain a mix of inputs and outputs – a relaxation of the best-practice principles described earlier. Similarly, in the interests of presentational clarity, results may be grouped, and/or presented in layouts that are structurally inconsistent with the underlying model – a further derogation from best practice.

These deviations reflect the specific role of the dashboard within a given model to act as an information and control hub for users. Provided they are programmed in a way that is linear and does not interrupt the primary calculation flow within the model, we believe the benefits outweigh the risks. To achieve this, the dashboard should sit outside the calculation flow. It should take outputs from the outputs sheet and, to the extent that it is used to control inputs, it should only be referenced by the main inputs sheets – not directly by the calculations sheets. This preserves the underlying plumbing of the model ie, its flow of data and logic.

12.3 Development of a dashboard

12.3.1 Audience

The target audience of the dashboard will be the sponsor, reviewer and any other stakeholders, who may be senior executives who could be relatively time pressured in their review of a financial model. Therefore it is important that the dashboard be designed with their needs in mind, and is able to provide a bite-size summary which can be rapidly digested.

The dashboard is also commonly the first port of call for any new user of the model, and as such must be constructed so as to be accessible and comprehensible to users who have limited background knowledge of the underlying model.

In practice this means that the dashboard must be:

- **Clear** – free from clutter, ‘headlines’ clearly signposted, and with appropriate use of charts or graphics to illustrate trends.
- **Coherent** – uses consistent formatting and a logical layout.
- **Self-explanatory** – includes appropriate labels or notes to enable a user to interpret the outputs without recourse to the underlying schedules.
- **Relevant** – includes only key inputs and outputs, excludes extraneous content.
- **Robust** – for example, makes use of locked cells, cell validation etc.
- **Printable** – as it is often useful to be able to summarise the key drivers and outputs into a printed format for discussion with stakeholders.

12.3.2 Planning the content

A good dashboard will be as simple as possible, but no simpler. This means that the inputs and outputs contained therein should be carefully considered, and focused on those which are of key importance. A cluttered dashboard, or one with too much information, will likely lead to a user struggling to see the wood for the trees.

This is not to say that a dashboard should be sparse, but rather its content should be prioritised. A good dashboard will be used to identify any issues with the project or business being modelled, and prompt questions. However, the dashboard may not hold all the answers, it is likely that the user will review the more detailed model outputs in order to determine the underlying cause of unexpected observations.

The content of the dashboard should be considered by the sponsor at the scoping/specification stage, to ensure that key views and metrics will be achievable using the proposed model structure. It may be difficult to build in new calculation drivers if they are identified by a reviewer or sponsor by their absence on a dashboard within a near completed model. As such careful planning is required from the outset.

12.3.3 Timing

While a dashboard layout might be designed in outline early on, we would recommend leaving the coding and programming until the core model is fully up and running.

The reason for this is that the dashboard will typically be designed to pull through key outputs from the interim output sheets of the model. While the model is in development, those interim outputs may be subject to change. Therefore to mitigate the risk of unintended consequences of changes further up the calculation chain, it is better to leave the coding of the dashboard until the interim outputs are locked down.

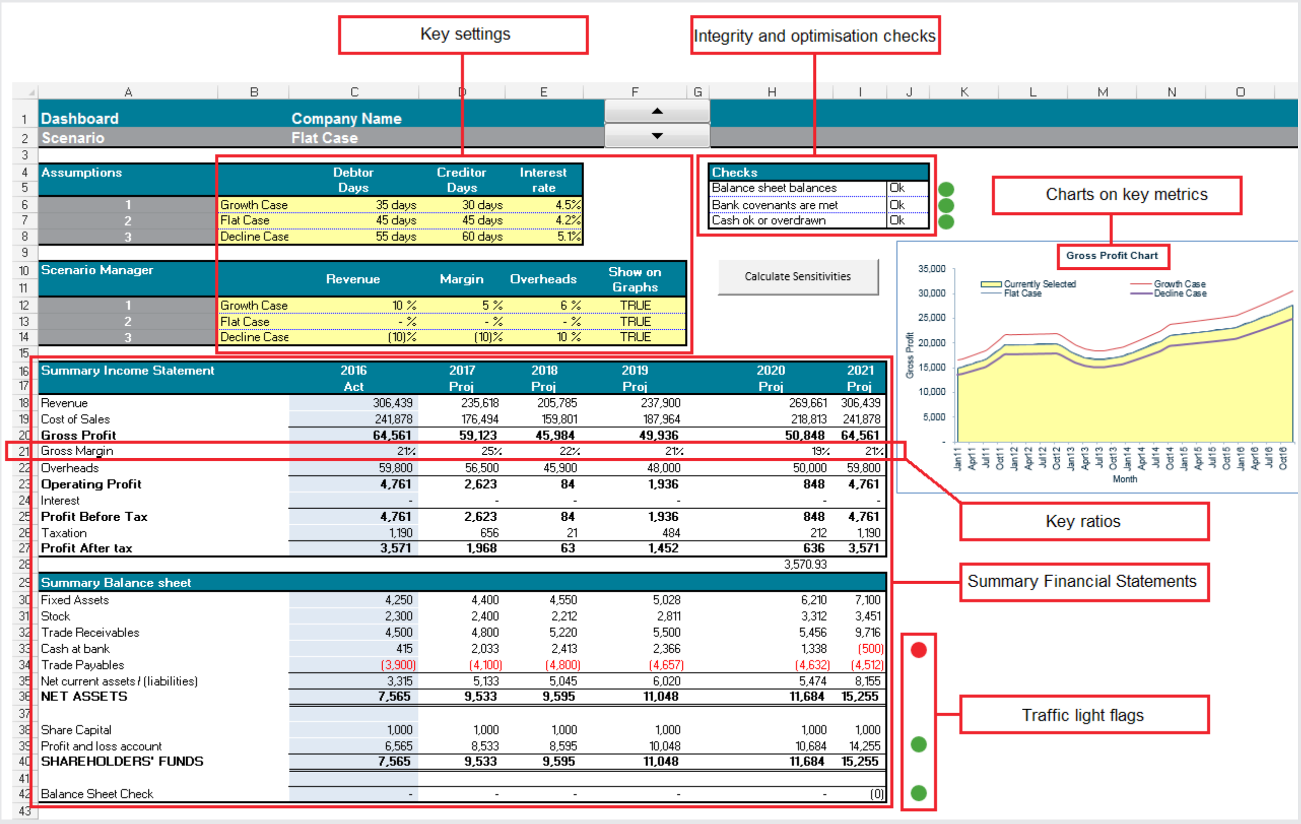
12.4 Ideas and techniques

The content of the dashboard will be dependent on the nature of the model being constructed, but a starting point would be to consider some of the following ideas for outputs reported, or key inputs captured.

12.4.1 Suggestions for outputs

The dashboard needs to show a high level snapshot view and also maintain a visual appeal, so prioritisation of outputs is crucial. Common outputs might include:

Figure 33: Simple dashboard example



- **Key settings** – an on-hand display of key assumptions and settings used, such as the chosen scenario.
- **Summary financial statements** – ie, a consolidated income statement, balance sheet, and cash flow on a quarterly or annual basis, depending on the extent of the timeline in the model. These statements will include less granularity compared to the underlying outputs.
- **Key ratios** – the choice of KPIs will depend on the nature of the model but could include, for example, trade receivables/payables days, revenue or profit growth metrics, funding covenants, debt service cover ratio, etc.
- **Optimisation check outputs** – Some models may include processes designed to iterate certain assumptions to meet targets. For example, a given shareholder return target, funding headroom, gearing ratios, or effective interest rate. Optimisation checks measure the effectiveness of the model in achieving these targets, and will likely report on each of the optimisation targets explicitly, including the target value, achieved value, and variance.
- **Integrity check outputs** – These are designed to identify the presence of accounting or arithmetical errors within the model, for example, that the balance sheet balances, reserves movement matches the income statement, and that the cash flow agrees to the balance sheet, etc. Integrity checks will also verify the validity of inputs, for example, if a proportional split input does not total 100%. Unlike the optimisation

checks, it is sufficient to include a simple 'ok/error' result on the dashboard, because any problem identified will necessitate investigation beyond the dashboard to resolve.

- **Flags and traffic lights** – visual enhancements to reflect check statuses and grade simple KPIs.
- **Charts** – for example visualisations of KPIs, illustrations of trends in moving KPIs over time, or illustrations of changes in sales mix by product or territory through the use of a pair of pie charts at two different points in time.

12.4.2 Suggestions for inputs

The dashboard is not the primary point of entry for assumptions into the model, but an area for *key* inputs on the dashboard is sensible as it makes them accessible to the dashboard audience.

Typically dashboard inputs will be business or project drivers which are critical to the outcome of the model, and potentially those which are subject to estimation or negotiation. Their proximity to the outputs adds value by enabling 'what if?' questions to be explored, and the effect of changes in estimates or renegotiation of terms to be reflected in the model quickly. Common inputs might include:

- **Start date** – for a project model the ability to flex start date of the project, go live, completion date, or other critical dates within the model.
- **Funding terms** – facility headline value, interest rate/margin, target ratios, LIBOR or base rate projection, funding mix, interest treatment, repayment timescale.
- **Business drivers** – growth rate, margin, product mix.
- **Working capital assumptions** – to amend payables and receivables days assumptions, if relevant.
- **Scenario chooser** – facility to amend several inputs en-bloc.
- **Form controls** – spinners or sliders and buttons which can be used to control input values on other worksheets.

12.4.3 Fetching source data

It is possible to use an intermediary dashboard workings sheet which will enable mapping of financial statement headings to their less granular dashboard counterparts using a SUMIF/SUMIFS formula, and also to allow the timeline to be flexed, for example to aggregate at a quarterly or annual level.

The dashboard workings sheet might also be home to the source data required by charts for the dashboard, although these could equally be contained in a dedicated section at the foot of the relevant output sheet. The key here is not to link charts directly to the workings sheets which may be subject to logic changes or rewiring at a later date, but instead, using a section linked to outputs, will ensure that the charts remain in sync with the values presented in the financial statements. This method also makes updating the charts simpler, since the chart source data section of the output sheet can be redirected to different output rows – ie, without having to modify the chart itself.

12.4.4 Returning input assumptions

A similar concept applies in terms of inputs on the dashboard; the model calculation sheets should not link to the dashboard itself. Rather the main model input sheets should link to the dashboard where appropriate. This has the dual advantage that the model can be developed independent of the dashboard, and the inputs are collected together on the input schedule for review. It is important to keep input cells in an easy to find location, even if (in the case of a dashboard link) the values are not directly entered there.

The process for linking input sheets to the dashboard assumptions will generally be as straightforward as including a link from the input sheet to the dashboard sheet. Similarly in the case of a scenario analysis, the 'scenario picker' (see section 13.4) will summarise the inputs selected and the input sheets will link to those. In these cases it is important to change the cell formatting on the input sheet to show that those 'input' cells which are now linked, are now 'call ups'. This flags to the user that these are no longer the primary input cells for those values.

However, simple linking is not always possible, for example, if a model calls for more than one dashboard view it may be helpful to have an input for the same parameter available on both dashboards. In these cases selective use of macros can be used to keep the main input sheet and the two dashboards in sync, by replicating inputs across the sheets programmatically.

Another alternative is to use 'form controls', which are buttons, for example 'spinners' which can be linked to an input cell. It is possible to link multiple spinners to an input cell, so it can be controlled from either spinner regardless of sheet location.

12.5 Key points

- Keep dashboards outside the main flow of logic.
- Give due consideration to the type and level of information shown on the dashboard.
- Develop dashboard with audience in mind, using formats and validation to make inputs more intuitive.
- Clearly identify key project inputs.
- Focus on required outputs and KPIs, avoiding clutter.
- Include high-level master checks to flag output errors.
- Use 'form controls' to amend input values from multiple places.

13 Model management

13.1 Overview

In this section, we consider some approaches to aid management of the model, assisting with:

- navigation and organisation;
- version control; and
- scenario control and analysis.

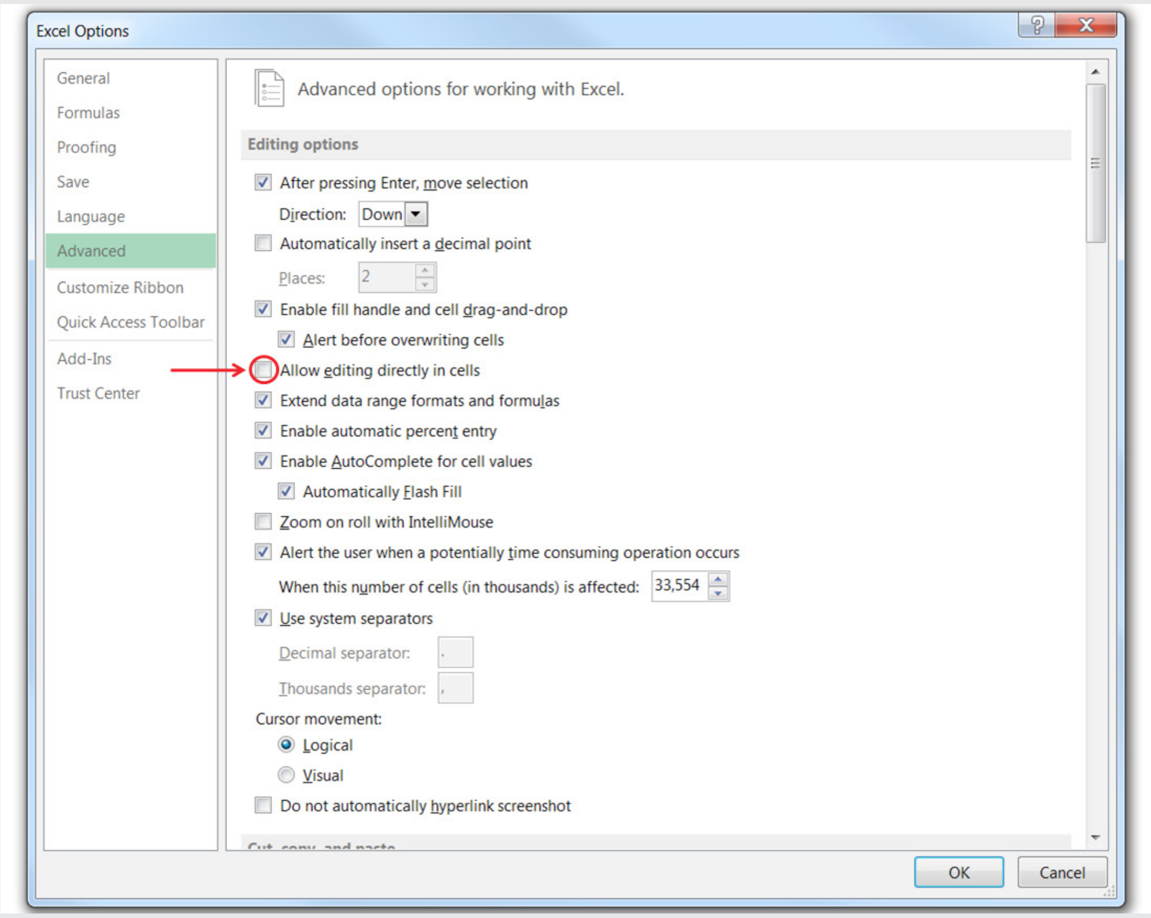
13.2 Navigation and organisation

Models can become very big, very quickly. Including tools to navigate easily around a model and to organise that model's content can save significant developer and reviewer time. Some basic tips to assist with user friendly movement through the model include:

13.2.1 Contents/ index pages

We recommend including among the information worksheets a hub for navigating quickly to other sheets and/or calculation areas within the model. The general approach apes the contents page of a book or technical manual: a list of worksheets, and under the worksheet headings, lists of the main components of those worksheets. The trick is linking the headings to the relevant sections, so that they are only a click or two away; there are two techniques:

Figure 34: Disabling 'editing directly in cells'



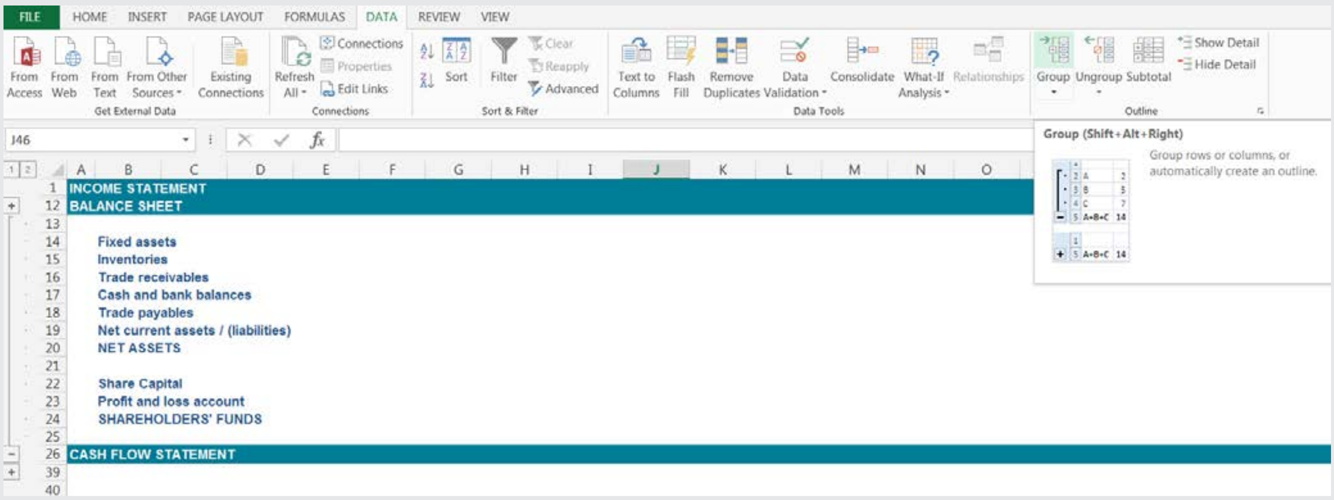
- **Hyperlinks.** The list of contents can be linked at a cell level by inserting hyperlinks referencing the referred sections. Hyperlinks can be inserted to a cell using 'ctrl+k'. In addition to links from text lists, the author can create a model structure map using drawing objects, and embed hyperlinks in them, so that clicking on map items takes the user straight to the desired model area. Note, hyperlinks do not follow cells when rows or columns are inserted, but this can be mitigated using named ranges.
- **Edit in cell.** The alternative is to create the contents by referencing the source sections' row labels or descriptions directly (using simple direct cell references). If 'allow editing directly in cells' is disabled under Excel 'options' (advanced), then double clicking on the cell will automatically jump to the relevant section (see Figure 34). This approach is quicker to set up than hyperlinks and the live link to the source label means that the links will remain valid if rows or columns are inserted.

To complete the loop, it is helpful to add hyperlinks on the individual worksheets to jump back to the contents page or model map – this should mean that any critical model area is only one to two clicks away; which is significantly more efficient than using a mouse and/or scroll bar to get to target areas.

13.2.2 Group and outline

Excel includes functionality to group and collapse rows and columns. Implemented smartly, calculation blocks can be collapsed (ie, hidden) to leave just section headers visible on the screen, which makes it easier to overview the structure and content of a model. It can also help the reviewer/developer when paging through a worksheet as they can opt only to have the subject matter of immediate interest visible at any given time.

Figure 35: Grouping data

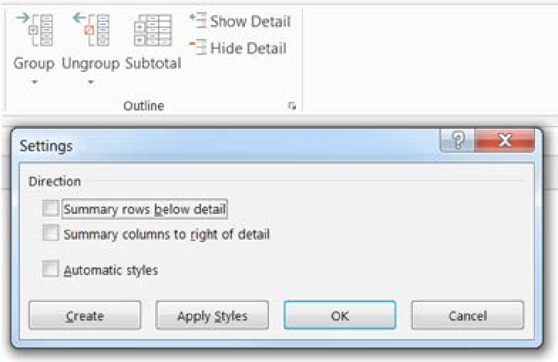


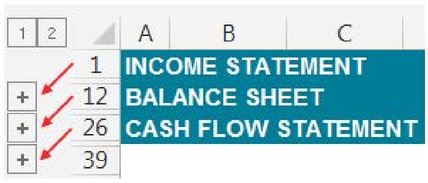
The grouping tools can be found under the data tab on Excel's ribbon (see Figure 35). Far quicker is the use of keyboard shortcuts:

- Shift+spacebar selects the entire rows to which the current active cells belong; or
- Ctlr+space bar selects the columns to which the current active cells belong.
- Shift+alt+right arrow then groups the selected rows or columns; and
- Shift+alt+left arrow will ungroup a selection or subset of a grouped selection of rows/columns.

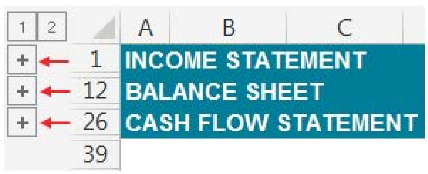
In addition, when grouping calculation blocks we suggest going to the outline settings (see Figure 36, on the opposite page) and unticking the check boxes. This puts the collapse/expand buttons to the left or top of the grouped area, which is more intuitive to use than the default settings.

Figure 36: Adjusting Excel's Group and Outline settings





The default settings place the ungroup buttons below the relevant rows, which is counter-intuitive



Unticking the direction boxes move the buttons to the row above the hidden area, aligning with row headers more intuitively

One note of caution when using the group and outline tools is that by default series on charts will disappear from the chart if the row or rows on which the source data resides are hidden using grouping. This behaviour can be modified so that the chart includes hidden data, by entering the 'select data' menu in the 'design' section of the ribbon, under 'chart tools'. Enter the 'hidden and empty cells' options, and tick the check box to 'show data in hidden rows and columns'.

A further practical consideration is moderation. Overuse of group and outline can obfuscate and irritate users. We would advocate using no more than one to two tiers of grouping, three in extremis.

13.2.3 Hidden rows and columns

It may be tempting to hide rows and columns for presentation purposes. However, we recommend avoiding this temptation, it is not immediately apparent to users when rows or columns have been hidden. By comparison, the 'grouping' function explained above provides a clear visual indication, by way of collapse/expand buttons on the left/top of the sheet. This achieves the same aesthetic result, but through an easily identified method, making it the preferable option.

A war story

When Barclays sent over its offer to buy Lehman Brothers in the wake of the firm's collapse, it included a list of assets they were willing to buy in an Excel spreadsheet. The modeller hid, rather than deleted, nearly 200 cells. When the Excel file was converted to a PDF and sent to court, the hidden parts of the spreadsheet reappeared and Barclays was forced to acquire an additional 179 toxic deals it never intended to buy.

Source: www.fortune.com

13.3 Version control

13.3.1 Version control in a deal context

In chapter 4 we talked about the importance of version control. This is a particular challenge in a deal context. The reasons for this are that there are typically: multiple model users/stakeholders, timescales tend to be pressured, and the analytical requirements (eg, sensitivities) may vary up to and including the last minute.

These specific challenges manifest themselves as follows:

- **Multiple users:** having multiple stakeholders creates two issues:
  - that parallel streams of model development occur and the underlying models on which different parties to a deal are relying become out of step (possibly dangerously); and/or
  - that inputs, calculations or settings are changed without clear notification or documentation and become (incorrectly) embedded in the base case model.
- **time pressure:** changes to requirements and model updates can become frequent and changes often have to be made at speed. Good version control maintains a clear audit trail, providing back-up information for any changes made (who changed what and when), as well as archived versions, providing a safe position to fall back to, if models become lost, files are corrupted, or if a modeller simply loses their way.
- **Variant analysis:** the need to consider and incorporate new scenarios and sensitivities (sometimes complex in nature) can tempt the modeller or model users to build parallel model versions for different scenarios. The issue this creates is that changes to the master model need to be replicated across any sister version in use. Depending on the nature of deviation away from the base case, the implementation of changes in variant versions may not be easily or quickly executed.

Tip

By using a macro, you can interrupt the save process, to request inputs for version control and take a snapshot of key model data, before saving. This can help to prompt users to comply with this discipline.

13.3.2 Practical approaches to version control

Dealing with version control intelligently relies on basic common sense and project management skills combined with some helpful programming techniques:

1. Try to limit the circulation of models:

- where possible circulate hard-coded output summaries for review;
- for collation of inputs, a simple 'cold' template can be circulated, with the relevant assumption-owners submitting data for incorporation by the model developer rather than uploading it directly to the model;
- make full use of the protection and validation functions within Excel to limit how users can manipulate the model (discussed in section 8.7); and
- communicate clear instructions on use and version control (within the model and in covering notes on model protocol).

2. Use a clear, simple and comprehensive file naming convention (as illustrated in Figure 37) covering:

- project name, ie, a short simple identifier;
- model version number/code, ie, ideally differentiating between major releases and minor updates;
- date, ie, a date and time stamp;
- variant, ie, an additional text field should further differentiation of purpose/version be required for ad hoc branches of analysis; and
- status, ie, an indication of the development or circulation status of the current version (eg, 'draft, final for review').

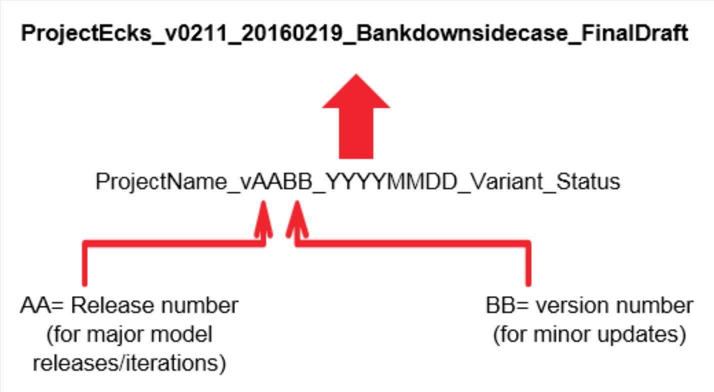
A war story

In 2011, a government tax assessor in California overlooked, then found, land worth $1.26 billion, generating about $12 million a year for taxpayers.

The incident was caused by a clerical error in which an experienced person used the wrong spreadsheet to calculate an oil producer's tax bill. While the oversight was spotted and corrected in time this remains an illustration of where good version control can help to protect users from costly errors.

Source: EuSprIG

Figure 37: Example file naming convention



3. Ensure all change requests are logged and routed through one person:

- The model author should be solely responsible for making any substantive changes to the model code and architecture.
- In relation to changes to inputs or assumptions, a clear, simple template should be used to capture information from data owners.
- Data owners should be listed in the model next to the key inputs so that responsibilities for review and sign-off of inputs are clear. As a further refinement, conditional formatting or a status indicator could also be used to indicate whether inputs have been updated or are still subject to review, and to differentiate actual inputs from 'plug' inputs used by the modeller as placeholders during a model's development.

4. Contain scenarios and model variations within the master model rather than proliferate multiple parallel model versions for scenarios, ad hoc analysis and sensitivities:

- Take advantage of the modular approach when adding new workings or ad hoc analysis. Think carefully before deleting old workings, instead build in the new ones alongside, with a switch to toggle between them (see Figure 8). The advantage of this is that it preserves the audit trail and ability to prepare bridge analysis for stakeholders to show the impact of input and coding changes between model versions. The cost can be a slight programming overhead (to retrofit switching assumptions to earlier workings/scenarios), and a possible adverse impact on the model's file size and run speed.
- Use a scenario control sheet to manage and contain different assumption cases and sensitivities. Essentially tabulate the different permutations of the key assumptions and use a 'picker' cell to tell the model which set to use. This is explored in more depth in section 13.4 on scenario control, below.
- Use data tables to analyse the impact of multiple sensitivities in parallel. If the scope of results is beyond the capability of data tables or their approach is not practical, use a simple macro to step through the scenarios and store the results in a 'dead' worksheet within the model, so that this data can be easily accessed and presented to stakeholders.

5. Set-up a version or 'change' log, embedded in the master model, to store details of changes between versions and their impact on key outputs:

- Store basic details about the model run (date, version number, etc), and a note of any substantive changes made, together with a summary of key results, each time a change is made.
- This process can easily be automated by a simple macro; we use a proprietary version log tool to do just this. It interrupts the normal Excel save routine to prompt the user for change details before incrementing the release or version number, storing a snapshot of key results and saving the file with an updated file name (in line with the convention above).

13.4 Scenario control

In the previous section we highlighted the usefulness of a scenario control sheet in managing version proliferation. Perhaps more importantly, good scenario control allows more rapid evaluation of alternatives, an improved audit trail, and the ability to carry out side-by-side comparisons more readily.

As intimated earlier, the basic architecture of a scenario control sheet is tabular. In general, we would structure the sheet with inputs that will vary by scenario/case listed to the left of the sheet, with a picker column for the live scenario next, followed by a grid of the assumptions for the variant scenarios to the right of that, with scenario number/references and descriptions listed across the top of the page from left to right. Figure 38 illustrates the principle.

Figure 38: Example scenario control grid

	A	B	C	D	E	F	G	H	I	J
1		Scenario Analysis								
2										
3		Project Parameters								
4										
5		Scenario #	2		Flex 1	Upside				
6										
7		Data	USED		Base	Flex 1	Flex 2	Flex 3	Flex 4	
8					1	2	3	4	5	
9										
10										
11		Case description	Upside		Original model	Upside	Downside	Expand Div 2	Expand Div 3	
12										
13		General model set up								
14										
15		Model start date	01 Jan 2016		01 Jan 2016	01 Jan 2016	01 Jan 2016	01 Jan 2016	01 Jan 2016	
16		Assumptions base	Updated		Original	Updated	Updated	Updated	Updated	
17										
18		Revenue (£'000)								
19										
20		Division 1	15,000		11,000	15,000	10,000			
21		Division 2	9,500		8,800	9,500	8,200	14,000		
22		Division 3	7,500		7,500		7,000		12,000	
23		Division 4	4,000		3,500	4,000	3,000			
24		Division 5	5,000		5,000					
25										

By way of explanation, the selection of scenarios is controlled by cell c5 in the example shown above. In the column below that cell, the grey cells include a simple 'picker' formula. That formula refers to the selector cell; and it counts that many columns across the grid of case-based inputs (columns e to i), and returns the value it finds.

In relation to implementation, there are a couple of aspects that can be approached in different ways:

- In respect of the picker column, this can be coded using a variety of Excel functions: CHOOSE, LOOKUP, OFFSET, and INDEX. The simplest and most transparent is INDEX and there is no compelling reason to use any of the others. The only limitation on INDEX is that it is constrained to look at a specific range. If you set that range to end in a column beyond the input grid in use, then you can insert columns at the end of the grid and the range should update automatically[4].
- With regards to the data grid, this can be set-up in one of two ways:
  - comprehensively: the full set of assumptions is shown for each and every case;
  - on a base case and exceptions basis: the full data set is shown for the first column (the base case), and data is only shown under other scenarios where it varies from the base (see Figure 38).

4 The OFFSET formula does not face a range limitation and is popular where there are a large number of scenarios. We would generally advocate care when using OFFSET as described in section 14.2, but this is a less opaque or error-prone example of its usage.

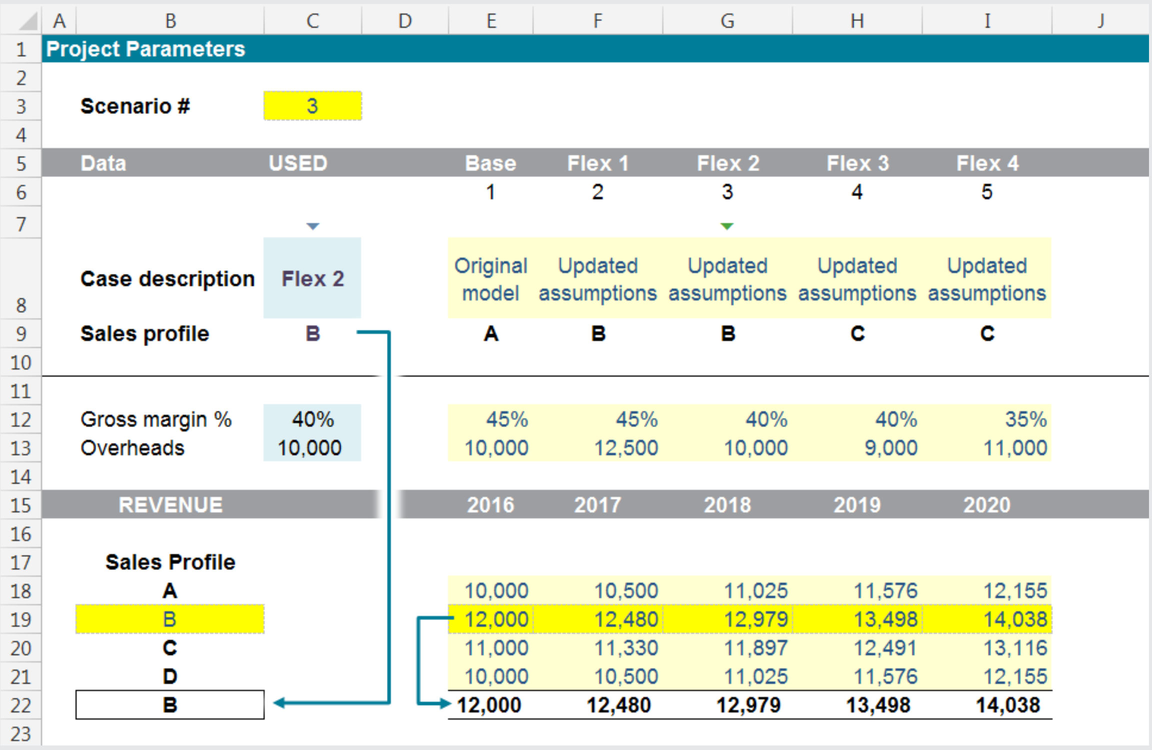
The latter makes scenarios quicker to set up, and means that there is always a default data set for each scenario input. Some would argue it is more readable. It requires slightly more complex coding in the picker column[5].

The usability/readability of a scenario grid can be enhanced using conditional formatting to highlight the live case or to highlight variances from the base or active case. (See Figure 38, where the red arrow on row 9 is used to highlight the active case, and orange arrows in column A denote variances from the base case.) And, as with calculation sheets, Excel's group and outline functionality can help to organise and navigate a lengthy scenario grid.

From a practical perspective, the way we recommend developing a scenario grid is first to develop the basic underlying model, with regular inputs sheets and their plumbing through calculations to outputs. Having set up a working model, we would then add a scenario grid, and as analytical needs evolve, add affected inputs to the grid, before linking the original input on the inputs page back through to source data from the picker column in the relevant row on the grid.

This approach preserves the integrity of the inputs page and flow of logic through to the calculations; it also means that a printout of the inputs page will capture all the live inputs used in that run of the model (and in a printer and reader friendly layout). It does of course stretch the separation principle, insofar as inputs are selectively replaced by formulae cells (albeit just single cell cross references). We therefore suggest colouring inputs that are linked back to scenarios inputs differently, so as to distinguish them on the inputs worksheets.

Figure 39: Scenarios within scenarios



	A	B	C	D	E	F	G	H	I	J
1	Project Parameters									
2										
3		Scenario #	3							
4										
5		Data	USED		Base	Flex 1	Flex 2	Flex 3	Flex 4	
6					1	2	3	4	5	
7										
8		Case description	Flex 2		Original model	Updated assumptions	Updated assumptions	Updated assumptions	Updated assumptions	
9		Sales profile	B		A	B	B	C	C	
10										
11										
12		Gross margin %	40%		45%	45%	40%	40%	35%	
13		Overheads	10,000		10,000	12,500	10,000	9,000	11,000	
14										
15		REVENUE			2016	2017	2018	2019	2020	
16										
17		Sales Profile								
18		A			10,000	10,500	11,025	11,576	12,155	
19		B			12,000	12,480	12,979	13,498	14,038	
20		C			11,000	11,330	11,897	12,491	13,116	
21		D			10,000	10,500	11,025	11,576	12,155	
22		B			12,000	12,480	12,979	13,498	14,038	
23										

5 In the example above, the formulation is '=IF(INDEX(Range,Scenario#)=0,Base Case Figure, INDEX(Range,Scenario#))'.

Two other aspects of scenario analysis that modellers should consider are:

- **data tables.** Scenarios can be used with data tables to great effect. Specifically, link one of the data table inputs to the scenario picker cell to enable 'batch' changes to inputs to be compared (as described earlier). One particularly powerful technique is to use scenarios and data tables to generate the results and data for use in a bridge chart.
- **scenarios within scenarios.** The restrictions of the columnar or tabular format are such that for very granular data sets they can become quite cumbersome. For long time series data sets, it may be more convenient and user friendly to set these out in horizontal format along the time series, with variants one above the other on a separate time-based inputs sheet. The scenario manager then simply needs one input in each scenario column – a reference number telling the model which set of data to use for that scenario (see Figure 39). This approach requires some care as it may lead to a stretching of the consistency principle.

Note that Excel has its own inbuilt 'scenario manager' function, which lets you store variant sets of inputs. It can store scenarios with up to 32 changing cells, but stores them outside the worksheet itself. Its key limitations are therefore a lack of both capacity and transparency; we have rarely seen it used in practice.

13.5 Scenario analysis

This guideline focuses on techniques for simple sensitivity analysis and scenario analysis; essentially, flexing key variables to understand their impact on dynamically modelled cash flows, and KPIs, whether single point changes (sensitivities), or involved combination sensitivities (scenarios). Testing of this nature is usually aimed at answering the question 'what if?', and tends to focus on breach testing ('by how much can this variable change before we breach a covenant, or run out of money, or need to step up fixed costs?'); or on relational testing ('how does a given result vary in relation to the flexing of a given input?').

Fundamentally, the point of these techniques is to contribute to an understanding of risk. However, while they may help to understand the impact of a particular combination of inputs (should that risk crystallise), what they do not generally do is to ascribe probability or likelihood to that set of circumstances arising. By their nature such sensitivities are deterministic.

To evaluate risk in more depth, one can apply simple subjective probabilities to a set of deterministic scenarios, or use statistically based simulation techniques.

13.5.1 Subjective probabilities

To help decision makers get a sense of the expected impact or most likely outcome, one can apply simple weighed probabilities to the outcomes from a series of potential scenarios. A set of cases or scenarios can be defined with the sponsor with the associated probability of each case occurring (such probabilities to add to 100%). The KPIs for each case can then be weighted according to their respective probability and summed to give an idea of expected outcome. This technique could be applied as part of a decision tree analysis process.

The challenges and limitations are that the cases used are subjective, individually deterministic, and typically limited in number. In addition, the respective probabilities are likely to be subjective. Finally, it is important to note that by weighting together a series of possible outcomes, the expected outcome is unlikely to represent any single specific actual case.

13.5.2 Simulation

For completeness, the other approach to assessing likely outcomes is through simulation, which, in a modelling context usually means Monte Carlo analysis. This technique requires the modeller to ascribe a set of characteristics to each input that describe the population (and distribution) from which it is drawn. The analysis is performed by running a series of simulations. Under each individual simulation, the sensitised inputs are replaced with randomly generated inputs from within each input's specified population parameter. The results for specified output KPIs are typically analysed and presented in the form of a distribution curve, which gives a view on the likely range and distribution of outcomes.

Monte Carlo analysis is typically run using thousands of individual simulations and can be time-consuming to compute. It can be run within Excel and there are a range of tools available as Excel add-ins[6] to allow such statistical (or stochastic) analysis of results.

We do not consider a detailed review of these techniques (and the tools that support them) in this guideline but, in brief, they can be very powerful in helping sponsors and other stakeholders to get a sense of likely outcomes. However, they face the same dangers as any modelling exercise: the results are only as good as the inputs. In this regard there are a number of challenges:

- They require a larger range of inputs, which are by their nature more technical (for example the parameters for different distribution curves), and which may not be readily understood by the model's audience.
- They require the modeller (and sponsor) to understand the distribution curves available and their relevance or appropriateness to each input, and the inputs being used may not naturally fit the distribution curves available.
- They require the modeller to properly identify and understand any correlation between the inputs.
- They may ignore the impact of corrective actions that would be taken in response to extreme values (depending on the sophistication of the model).

For these reasons we rarely see such analysis being applied in a transactional context except in more specialist sectors. This is a complex area and we would therefore advise modellers to proceed with care. And while we note that such techniques are being applied more widely due to the easy availability of simulation tools, the problem we have seen on occasion is that their use can be naive or overly simplistic, betraying a lack of understanding of statistics or of the nature of the populations being simulated.

13.5.3 Analysis Toolpak

Excel includes an add-in called the 'analysis toolpak,' which can be activated by going to 'options', then 'add-ins', then 'manage Excel add-ins', and ticking the box for the 'analysis toolpak,' and clicking 'ok'. Once activated, it can be accessed from the 'data analysis' button, from the analysis section of the data tab on the ribbon. The add-in can be used to help perform various statistical analysis functions.

6 Crystal Ball (Oracle) and @Risk (Palisade) are the two most commonly seen.

13.6 Key points

- Use linked contents pages to enable rapid navigation to the different areas of a financial model (either with hyperlinks or by disabling 'edit directly in cell' and using direct cell references).
- Use group and outline to enable users and reviewers to collapse model components that they are not currently using, and to provide a worksheet content overview.
- Version control is a major challenge in a fast moving, multiple stakeholder, iterative deal process.
- Manage version control by: limiting model circulation, using clear filename convention, logging and implementing change requests through one person, incorporating all scenarios and model variations within a single master file, and maintaining a change log (preferably automated).
- Scenario control can assist with capturing multiple variants in a single file.
- Scenario-based inputs can be used to great effect with data tables.
- Consider scenarios within scenarios to manage time series data in a user- and presentation-friendly manner.
- Prefer scenario tables laid out on worksheets over Excel's inbuilt scenario manager.
- Monte Carlo analysis can help assess risk and likely outcomes but should be used with care.

14.1 Overview

Productivity is a general underlying theme running through this document: lots of the techniques and advice are aimed at reducing repetition, saving wasted time, and increasing efficiency.

There are four further aspects to explore in relation to productivity:

- general modelling dos and don'ts;
- keyboard shortcuts;
- masks, counters, and flags; and
- automation.

14.2 General 'dos and don'ts'

While the link with productivity may seem tenuous; in practice we find that observing the dos and don'ts in this section results in models that are easier to understand, easier to update, and easier to review. As a result, they reduce risk and save time, both of which can be important in a transactional context.

Note that some of the techniques we advocate here (and in earlier chapters) can adversely impact on run time. For example, using extensive SUMIF or indexing-type calculations may slow an Excel file's calculation speed; it can be a trade-off between speed, and applying best practice. While it is important to be aware of this point, in our view, best practice and user friendliness will usually trump run speed. In any case, there are usually workarounds available and with modern processor speeds we have not generally seen this become a problem, even under the time pressures associated with a deal.

The practices listed below are generally to be avoided or in some cases, where used, used with caution, and an awareness of the potential risks or issues.

- **nested IF statements.** Nesting IF statements refers to the use of one or more IF statements within the two 'then... else...' legs of an IF statement. In fact, IF statements can be nested many times over. It is very much the case that IF statements are the building blocks for developing the decision trees in a financial model and we consider them invaluable in that role. However, used too liberally within an individual formula, they can result in difficult-to-decipher code, where the outcomes of different permutations of precedent formulae and inputs are all but impenetrable, and errors can lurk. In most cases where multiple solutions or outcomes are needed, other functions usually provide better alternatives:
  - Use the logical functions AND/OR to test for multiple decision conditions.
  - Use INDEX or CHOOSE to replace multiple IF statements used when there are more than two options/results to choose between.
  - Use results from 'mask' calculations to apply simple yes/no, or do/don't, logic to results of other calculation rows (see section 14.4).
  - Use simple Boolean logic to replace the IF function where one of the outcomes is zero. For example, replace: =IF(test, calculation, 0) with =calculation * (test). Excel returns the results of a simple bracketed logic test as TRUE or FALSE and treats the results as 1 or 0 when used in this way.[7]

7 Note, while logical tests which evaluate to Booleans can be used successfully to multiply other figures (thereby returning the relevant figure, or zero) they are not a suitable substitute for 1 and 0 in all circumstances. For example, they cannot be used in a quantum comparison, "=1>(1=0)" will evaluate to FALSE. To get the required answer you must either multiply the Boolean result by 1 "=1>(1=0)*1", or include it within a proper IF statement "=1>IF(1=0,1,0)", both of which evaluate to TRUE.

- Use MIN and MAX to replace IF tests when used to test for positive or negative values. For example:
- MIN(Value, 0) only returns the value when it is negative.
- MAX(Value, 0) only returns the value when it is positive.

- **external links.** In our experience multiple linked workbooks are a frequent source of problems, they can become unstable, result in orphan links, require even more disciplined version control than would otherwise be applied, and can make formulae overly long and difficult to interpret. As described in section 8.4.1, if they must be used, we recommend treating them like inputs: call up linked data on the inputs sheet or on a dedicated worksheet, with simple references (ie, do not link cells in external files directly into calculation formulae), and make sure they are clearly labelled and identifiable.
- **no balancing figures.** This is key to the model integrity principle discussed above. The balance sheet should not contain any balancing 'plug' figures (for example, forcing any balance sheet differences through the reserves or cash lines). Instead, each of the balance sheet items should be independently calculated and included in the balance sheet statement: this particularly applies to cash balances (which should be linked to the cash flow statement), and retained reserves (which should be linked to the income statement account). The point of this is that the balance sheet can then properly fulfil its role as a natural check on the model's logical integrity.
- **offset** / **indirect.** The OFFSET and INDIRECT functions should be used with care. They return values or series of values based on a set of virtual grid references. The issues are that the referred cells are not always clear from the results and that they cannot specifically be seen or identified using Excel's audit tools. Furthermore, these functions are volatile functions, and therefore will be recalculated on every operation, regardless of whether or not they are affected by such operations, which can introduce additional, unwarranted calculation time. Programmers find these functions useful because they allow a great deal of flexibility to manipulate and move data within a model. If they are to be used, we strongly recommend breaking out and labelling the constituent elements (or sub-calculations) that determine the grid references and ranges of cells that are picked up.
- **lookups.** Excel has some powerful functions for identifying and selecting data from tables of inputs, specifically LOOKUP, HLOOKUP, and VLOOKUP. These are very popular and widely used, two notes of caution:
  - The use of the various lookup functions to reference data between worksheets can significantly slow down calculation times, each formula having to page across the workbook multiple times to return a result. In addition, the tracing of precedents and dependents can become messy, with lookup formulae referring to multiple target cells (see Figure 7).
  - The lookup functions need to be properly set up with target data organised in order[8], or containing precise and matching values, otherwise the function may not return a meaningful answer.

Generally we prefer INDEX and MATCH to manipulate and call up data from grids and arrays. If lookup functions are used, they should come with clear warnings.

- **round** (and similar). Only use these selectively, or for presentational purposes, if at all. Excel's number formats can achieve the same effect. Avoid using ROUND in the calculation or logic flow, since it will truncate accuracy and may throw out erroneous results, potentially triggering balance sheet error checks. One area where ROUND,

8 Note that when working with an approximate match, Excel LOOKUP functions match on the last value that is less than the lookup value. This needs to be borne in mind when designing formulae and lookup tables that seek to apply values (for instance growth factors) between sets of dates. In other words, when searching a simple list of dates and annotated assumptions, Excel treats each assumption as applying 'from' the annotated date, and not 'to' the annotated date, and the lookup table is assumed to be in ascending order. When working with an exact match, the ordering of the lookup table will not affect the returned result, except where there are duplicates, in which case the first instance is matched.

or related functions, such as CEILING or FLOOR, may be required, is for non-financial calculations, where units cannot be subdivided, or are subject to step change; eg, staff numbers, or plant and machinery with constrained capacity.

14.3 Keyboard shortcuts

The most significant contribution to productivity and speed of model development in Excel is probably the use of its keyboard shortcuts in lieu of the mouse-based, point, click, drag approach. We can't overstate how much more efficient the key strokes are in navigating, selecting, copying, pasting, and filling worksheet cells.

The principles and shortcuts are pretty straightforward; there is no substitute for practice and experience. We have included a separate list of the most commonly used and helpful shortcuts in Appendix C.

14.4 Flags, masks and counters

In transaction models, it is common to find formulae that test against dates in the timeline. For example, modelling the repayment of a bullet loan on a certain date; rent quarters; applying stepped interest margins; terminating time-bound contracts.

Three techniques are introduced below. The purpose of the illustrations is to give readers an idea of how the principles work so that they may experiment and explore their use.

14.4.1 Flags

In many cases, date-based tests are used repeatedly in multiple formulae rows. Rather than repeat what can be complex logic each time (with the attendant risk of error/inconsistency), it is generally more efficient and transparent to model such tests once at the head of a worksheet and then use a simple yes/no, or pass/fail, result (typically reflected as 0 or 1), when the test is used in calculations elsewhere on that sheet.

For example, setting up a calculation row to return a value under a specific date can be used to act as a flag to the model to perform a particular calculation in that period. The repayment of a bullet loan, mentioned above, is a good example (expressed in English: if repayment flag is one, repay the loan, otherwise do not). The applications and uses of flags are numerous and this is just a simple illustration.

Figure 40: Use of flags

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Timeline		2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	
2													
3	Bullet repayment year		2023										
4	Interest rate		5%										
5													
6	Bullet repayment flag		0	0	0	0	0	0	0	1	0	0	
7													
8	**Interest working**												
9	Opening balance		10,000	10,000	10,000	10,000	10,000	10,000	10,000	10,000	-	-	
10	Repayment		-	-	-	-	-	-	-	(10,000)	-	-	
11	Closing balance		10,000	10,000	10,000	10,000	10,000	10,000	10,000	-	-	-	
12													
13	Interest paid		500	500	500	500	500	500	500	500	-	-	
14													

14.4.2 Masks

Taking the principle a step further, if a modeller models a date-based test, such that it returns a series of ones and zeroes along the row (reflecting pass/fail respectively), then, instead of using an IF test in the calculations affected, the modeller can simply multiply through those calculations by the date-based test row (also known as a 'mask'). This is illustrated by the simple example in Figure 41.

Figure 41: Use of masks

	A	B	C	D	E	F	G	H	I	J
1				01/01/2016	01/02/2016	01/03/2016	01/04/2016	01/05/2016	01/06/2016	01/07/2016
2										
3	Repayment start date		01/01/2016							
4	Period counter		3							
5	Quarterly rent		100		=IF(MOD(MONTH(D$1)-MONTH($C$3),$C$4)=0,1,0)					
6										
7	Quarterly flag			1	0	0	1	0	0	1
8	Quarterly rent paid			100	-	-	100	-	-	100
9										
10					=D$7*$C$5					
11										

14.4.3 Counters

Closely related to simple '0/1 masks', the next application of the principle is to set up counters for calculations whose outcome varies according to the relative position of a period along a timeline.

Annuity loan repayments provide a good illustration. To calculate how much principal to repay in a given period, Excel's PPMT formula needs to know how many repayments are made, the interest rate, principal amount and which numbered repayment is being made in the current period. Hardcoding these values into the formula is risky and error prone — coding the date information, tests and period numbering into the formulae can result in overlong/complex logic.

Accordingly, we recommend setting up separate counter rows that contain the logic to determine period and repayment numbers for use by the PPMT formula. This principle is illustrated in Figure 42. An alternative approach is described on page 59.

Figure 42: Use of counters

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Timeline			2016	2017	2018	2019	2020	2021	2022						
2																
3	Payment start date		2017													
4	Interest rate		5%													
5	Principal amount		500,000													
6	Payment periods		5													
7																
8	Flag to identify repayment per			0	1	1	1	1	1	0	=IF(AND(D1>=$C$3,D1<SUM($C$3+$C$6)),1,0)					
9	Period Counter			0	1	2	3	4	5	0	=SUM($D$8:D$8)*D$8					
10	PPMT			-	(90,487)	(95,012)	(99,762)	(104,750)	(109,988)	-	=IF(D$9=0,0,(PPMT($C$4,D$9,$C$6,$C$5)))					
11																

14.5 Automation

Macros can be a contentious topic. They can take spreadsheet models more explicitly towards the world of computer programming, and not all modellers or model users are comfortable with that: it is a distinct discipline.

We do not consider macros and VBA programming in detail in this guideline; instead we offer some general words of advice on their use and when and where it is appropriate.

Where macros are used for calculations they can be opaque, and results and workings leave the boundaries of the worksheets, to be stored as variables in the VBA environment. One of the benefits of the worksheet layout is that the 'live' results of the current case and calculations are visible, stable, and easily auditable.

As a result, in a deal context, we recommend avoiding using macros in the calculation flow wherever possible. One possible exception to this principle being their use to break circularities (where this can be done transparently and the intervention is clearly visible on the worksheet) as described in section 8.5.

Where macros come into their own is in automating simple repeated tasks and model administration. For example: running print routines, cycling through and storing scenario results, automating model review procedures, managing model access permissions, creating black box user interfaces, importing and exporting data, and building tools to help navigate through a model.

14.6 Key points

- There are a number of programming techniques and functions that should generally be avoided, including nested IF statements, external file links, and INDIRECT.
- Functions such as OFFSET, ROUND, and the lookup variants should be used with care.
- Balancing figures should be avoided at all costs.
- Many logic tests can be simplified using AND, OR, MAX, or MIN.
- Multiple choice logic tests should use INDEX or CHOOSE in preference to nested IFs.
- Using Excel's keyboard shortcuts massively increases productivity and speed of development and orientation in the model.
- Flags, masks, and counters can be used to separate out time-based calculations and logic tests and contribute to simplicity and transparency.
- Macros have a role to play in automating routine modelling tasks but we would recommend caution if the calculation path is dependent on macro-generated results.

15 Model speed

15.1 Overview

In an ideal world, financial models would be detailed, flexible, fast to build, and fast to calculate. In the real world every model represents a compromise between these goals. The very fastest models may be less flexible, but building a very flexible model may necessitate less efficient formula choices.

There are various reasons why a workbook may slow down:

- The ready availability of detailed source data can often lead to a temptation to increase the level of detail within a financial model to accept such data, for the additional flexibility it can provide.
- Models may be passed amongst various editors, not all of whom may be familiar with best-practice modelling or use of the Excel software, introducing problems.
- Data may be copied/pasted between workbooks, bringing with it formatting, names, styles, and other unnecessary baggage.
- Complex models can involve extended calculation chains, such that changing a single input can have large implications and affect a great many cells.

Through careful choice of formulae and techniques there are often many ways to create an equivalent working which can result in much faster models, operating at 10, or even 1,000 times the speed. In this section we provide insight into how you can improve model calculation speeds.

15.2 Relevance

In the 1990s, it was not uncommon to have to start a model running and then go and make a cup of tea or grab a sandwich while it cycled through the calculation tree. In more extreme situations, you might have had to leave a model running over night. With modern computers these challenges are largely behind us, but in certain more data-intensive corporate finance fields, such as securitisation, and rail franchise bidding, speed can remain an issue. It is also fair to say that expectations of performance have shifted, and sponsors and users can be intolerant of even relatively short calculation delays. With these points in mind, this chapter provides guidelines to help improve model speed and efficiency.

15.3 Choice of formulae

In many models we see formulae that, while they provide the correct output, may not be the most appropriate choice of formula and that faster, or more flexible alternatives may be used to achieve the same result.

One important feature in Excel's early development was the launch of its 'smart recalculation engine,' which improved efficiency by tracing calculation chains to determine which cells to recalculate, instead of recalculating all cells each time. To do this Excel maintains a list of cells that have been changed and cells that are dependent on these cells. Of course, while efficiency was the goal, it was fundamentally important that Excel maintained its accuracy and produced correct results, so certain calculations need to be performed each and every time and may be described as volatile, such as RAND(), TODAY(), and even conditional formatting. This means that they recalculate after any common calculation trigger events, such as changing the contents of any cell, inserting rows, renaming worksheets, etc — regardless of whether that particular cell was part of the calculation chain of the amended cell. This can then trigger further calculations where non-volatile, dependent cells are impacted, even if it doesn't result in a change to the results.

By choosing non-volatile operators over volatile (eg, INDEX over OFFSET; and CHOOSE over INDIRECT), you can reduce the time spent waiting for unnecessary calculations that Excel performs as a matter of precaution.

The following is a list of formulae which Microsoft identifies as being volatile:

- NOW
- TODAY
- RAND
- OFFSET
- INDIRECT
- INFO (depending on its arguments)
- CELL (depending on its arguments).

15.4 Structural optimisation

Similarly, simple structural changes can help cut down the time spent calculating formulae. By simply re-ordering arguments, or separating formulae, calculation times can be vastly improved.

15.4.1 Re-ordering arguments

Some simple consideration for the order in which arguments are entered into a formula can make a big difference. Take for example a series of filters applied in a SUMIFS statement. This function adds up the values in a given range, contingent on the values in corresponding ranges matching a given set of criteria. Unlike the SUMIF formula, SUMIFS allow the modeller to apply multiple filters through the single formula. The SUMIF formula only adds up values which meet all the criteria. Each row of data is tested against the criteria in order, until one of them is FALSE. Therefore, by using the most restrictive criteria first, a greater number of values can be excluded earlier in the evaluation process.

For example, should you wish to sum the value of insurance claims made by male drivers, between the ages of 17 and 25, you have the choice of ordering the criteria tests by age and then gender, or gender then age. By using age as the first test in the series, many more drivers would be assessed as 'FALSE' at the first stage, and therefore not proceed to the second test. In comparison, if gender were to be tested first, approximately 50% of the tests would proceed to the second stage of assessment.

Where applied to a large number of cells, the difference in calculation times can be quite significant.

15.4.2 Minimising repetition of calculations

The normal method to construct a formula will be to draft it in the first cell of a required range, confirm its function, and then copy/drag the formula to fill the range. Assuming the formula is drafted correctly this tried and tested approach will mean that formulae used are consistent across the range and the results are as required. However, it might not always be the most efficient in terms of calculation speed.

Example 1 – Reference to a volatile function

For example, if you wished to assess thousands of dates in column A, and label those that were in the past, you may use an IF statement similar to the following:

=IF(A1>TODAY(),"Future","Past")

For each cell that contains this formula, Excel will calculate today's date through the TODAY() function; it will compare this to the record date using the > greater than comparator; and it will then return either 'future,' or 'past,' through the IF statement – ie, three operations per date. So, for 1,000 dates in the list, there would be 3,000 operations.

A faster way to structure this would be to split the formula out into two parts, identifying the duplicate operation, and then placing the duplicated section in a separate cell so that it is only calculated once. For example, you could restructure the above formulae, by placing the formula =TODAY() in cell d2 and calling up the result of d2 for use in numerous other formulae of the following form:

=IF(A1>$D$2,"Future","Past")

Using this structure, there would be only two operations per date: the comparison using ">"; and the evaluation of the IF statement. Therefore 2,001 operations for the 1,000 comparisons, saving 999 operations.

Example 2 – Fetching data

There are many circumstances where it is necessary to reference a particular row of source data, to be pulled across into another row elsewhere in the workbook. For example a data table may have sales by unique type as rows, and by month across the columns. When the modeller needs to pull data for a given sales type to another area of the workbook there are several options available, but each with their own pros and cons.

SUMIF – This is one of the slowest because even once the required row is found, all subsequent rows are still checked.

VLOOKUP – This is better, because once a matching row is found, no other rows need to be checked, making it faster. It also retains the advantage of flexibility in row location, but it does require labels to be unique. VLOOKUP has the disadvantage that inserting columns into the source data table will cause formulae to refer to incorrect columns.

INDEX/MATCH – This is better than VLOOKUP because it will run faster and also, as the 'index' and 'match' arguments are independent, it will cope with inserted columns in the source data.

'Match' once, 'index' many – The normal method for coding an INDEX/MATCH function would be to embed both parts within a single formula, ie:

=INDEX(xxx,MATCH(xxx,xxx,0))

However, when each column in a timeline is to reference the same row of source data it is more efficient to separate the functions. Ie, use the MATCH function at the left of the destination table to look up the required row in the source data, and then use multiple INDEX functions across the column, but each referencing the row number already found. This reduces, by a significant number, the number of costly MATCH search functions required. This function is therefore faster than the previous options, whilst retaining the flexibility in row number, but does have the same limitation that it can reference only a single row. It has an advantage for auditing the model, because the source row referenced is explicitly shown to the left of the destination data.

Direct link – This is the fastest method to fetch data, ie, single link to a single cell, because there is only evaluation of the link to do. This method has the disadvantage that the source data always needs to be on the same, single row. However it is good for audit trail because the reviewer can use 'ctrl+[' to jump to the source cell.

15.4.3 Off-sheet references

From an efficiency optimisation perspective, it can be much more efficient to reference cells on the same spreadsheet than to refer to off-sheet, or off-book, sources. Of course there are certain core concepts that we advocate in all circumstances, such as separation of inputs from calculations. However, it is worth keeping this impact in mind when designing resource-intensive models.

15.4.4 Large lookup ranges

When creating lookup functions, it is worth considering how Excel looks up the value. If the value exists in the lookup range, Excel will on average check the first 50% of rows before finding a match, however if the value does not exist, it will check all the way to the bottom of the range and return nothing. In some cases, users may specify whole columns as the lookup ranges. While Excel is intelligent enough not to analyse the entire column, and only analyses the used range of cells, this can still be an inadvertently large number of cells, for example due to users' indiscriminate use of formatting, etc, increasing the size of the used range. Therefore, it can be beneficial to be more specific, or to use dynamic lookup ranges to minimise calculation time.

Changes within lookup ranges can cause thousands of calculations to be recalculated, regardless of whether the change impacts the results. For example, if you have a column containing 10,000 lookups referencing a lookup range of 100 rows by five columns, a change to one cell in the lookup range would cause all 10,000 lookups to be recalculated, even if the changed cell did not affect the results.

15.5 Other time savers

15.5.1 Calculate on save

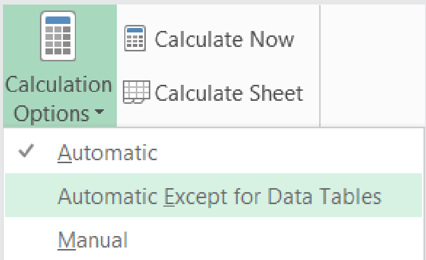
Each time Excel saves your model, its default behaviour is to recalculate prior to saving. If your model is particularly large or complex, and the time taken to recalculate your spreadsheet is problematic (for example, when Excel auto saves mid-session), you could consider changing this behaviour within 'tools>options>calculation>manual,' by unticking 'recalculate workbook before saving'. Of course, we would always recommend using this feature with caution, since undetected errors may be present in an uncalculated workbook. Note, whilst calculation is set to manual you can press f9 to cause a recalculation of all open workbooks.

15.5.2 Working with data tables

As discussed in section 11.3, it is possible to use data tables to model several different combinations of input variables simultaneously. Excel does this by sequentially substituting each assumption (or pair of assumptions in the case of a two dimensional table), into the calculations, and re-evaluating and storing the result in the table. As such, calculation times during development can be adversely affected because Excel recalculates all the scenarios each time a change is made.

Therefore it may be advantageous to change the calculation mode during development from 'automatic,' to 'automatic except for data tables,' in order to prevent unnecessary recalculation of all scenarios. This change can be made from the formulas tab on the ribbon, as illustrated in Figure 43 below.

Figure 43: Changing calculation modes



Note, whilst the calculation is set to 'automatic except for data tables,' you can press f9 to cause a recalculation of all open workbooks, including data tables. Naturally if the use of data tables creates significant performance issues, there are alternative solutions available to provide the same analysis, for example use of manually triggered VBA routines.

15.5.3 Other volatile functions

It is not only formulae that are volatile, but also other functions such as conditional formatting. Formats will therefore be re-evaluated upon each recalculation, regardless of whether their calculation chain has been modified. Removing conditional formatting may therefore increase the calculation speed of your workbook's macros.

Use of macros can have huge advantages in automating manual processes to increase accuracy and speed. However, there are also some opportunities to increase the speed of macros themselves.

For example, turning off calculation during a macro run can make it run a lot faster by eliminating the automatic calculation performed after each workbook change. Extreme care must be taken to ensure that calculation chains are not artificially broken through the use of manual calculation mode or selective recalculation by the macro.

Disabling events during a macro run can also make it run faster, as event triggered macros will not be called. Take care to confirm that this is the required behaviour, particularly when automating manual processes.

By far the most effective method to speed up macro performance is to turn off screen updating during the macro. While some find it can be satisfying to watch a macro perform actions at lightning speed, the additional overhead of forcing constant screen re-drawing adds an additional unnecessary burden to the processor.

15.5.4 User defined functions

User defined functions (UDFs), are worksheet functions which are not native to Excel, rather they have been written in VBA by a user of Excel.

UDFs can be helpful in providing functions which could not easily be recreated using native formulae, but they have two significant disadvantages.

- They are more difficult to review than native functions, because doing so requires the reviewer to understand the VBA code which could be quite complex; and
- UDFs will generally evaluate a lot more slowly than native Excel functions, because the calculation engine in Excel is highly optimised to process native functions.

UDFs should therefore be avoided unless absolutely necessary, and particularly in cases where a model's recalculation speed is adversely affected.

15.5.5 Used range

Worksheets in Excel 2007 onwards allow for more than 16,000 columns and more than 1,000,000 rows. Most business computers will not have sufficient memory to simultaneously use all the rows and all the columns on a worksheet. Indeed most spreadsheets only use a fraction of the available space. Therefore, to save memory Excel keeps a track of the used range, as it only has to calculate values and formatting in the area in use.

However, through copy/paste operations, it is sometimes possible for a worksheet's used range to get out of sync with the range which the modeller actually intends to use. Ie, Excel can mark almost all rows or columns as being in use. This will cause the model to take up more memory and run more slowly than it needs to, and it should be cleaned up.

To check if a worksheet has this problem, pressing 'ctrl+end' will activate the last cell in the used range; if it is a long way from the actual work on the worksheet then there is an opportunity to clean up. The simplest way in Excel 2013 and later is to use the 'clean excess cell formatting' tool from the 'inquire' toolbar, which is discussed more in section 16.6.2.

If you are using an earlier version of Excel or do not have the 'inquire' toolbar then excess used area can be reset by first highlighting the first blank row after your work, and press 'ctrl+shift+down' to select from there to the end of the worksheet, and then press 'ctrl+-' to delete these rows. Repeat a similar process on columns if required (using 'ctrl+shift+right arrow'). Then save, close and reopen the workbook and the used area will be reset.

15.5.6 Linking to non-continuous areas

Sometimes it is necessary to call up data from another worksheet, where the rows which are being linked to, are not adjacent; for example, if there is data on rows 11, 21, and incrementing by ten to row 1,001, and you would like to call data from those 100 rows to a sequential block of rows 1 to 100 on another worksheet. There are several indirect methods which can be used, for example with functions like SUMIF, SUMIFS, INDEX, OFFSET, etc. However, suppose it is necessary to do so through direct links, for example for speed or ease of review. Clearly it would be possible to set up the 100 direct links one at a time manually, but there are methods which can be used to set up these links in a more efficient way.

Method 1 – Using FIND command

It is possible to use the FIND command to identify and then select non-continuous areas of cells which can then be copied. A continuous block of links can then be created using 'paste special' and 'paste links'.

Method 2 – Using a formula to write a formula

An alternative method is to use a formula to create a formula. The concept is to carefully construct a formula, whose output is a text string which represents the required link formula. Once created these formulae can be copied and pasted as values, and then converted to live formulae. The images below illustrate how this could work to create links to every 10th row on sheet one.

These are the formulae which have been entered:

Figure 44: Using a formula to write a formula – step 1

	A	B	C
1	=Sheet1!A11		
2	="=Sheet1!A"&B2	21	
3	="=Sheet1!A"&B3	=B2+10	
4	="=Sheet1!A"&B4	=B3+10	
5	="=Sheet1!A"&B5	=B4+10	
6	="=Sheet1!A"&B6	=B5+10	
7	="=Sheet1!A"&B7	=B6+10	
8	="=Sheet1!A"&B8	=B7+10	
9	="=Sheet1!A"&B9	=B8+10	
10	="=Sheet1!A"&B10	=B9+10	
11	="=Sheet1!A"&B11	=B10+10	
12	="=Sheet1!A"&B12	=B11+10	

These are the values which the formulae evaluate to:

Figure 45: Using a formula to write a formula – step 2

	A	B	C
1	A		
2	=Sheet1!A21	21	
3	=Sheet1!A31	31	
4	=Sheet1!A41	41	
5	=Sheet1!A51	51	
6	=Sheet1!A61	61	
7	=Sheet1!A71	71	
8	=Sheet1!A81	81	
9	=Sheet1!A91	91	
10	=Sheet1!A101	101	
11	=Sheet1!A111	111	
12	=Sheet1!A121	121	

15.5.7 Alternative to SUMIFS

As discussed, the SUMIFS function adds up values from a given range where several conditions are met. Generally this is used to sum a particular column of a table, on rows where values in adjacent columns match criteria.

The same result can be achieved by adding a new helper column to the source data table, with this column reflecting the same logical filters which were part of the SUMIF. For example, the helper column could return a TRUE/FALSE result, based on values in other columns on that row. It is then possible to use a SUMIF or single criteria SUMIFS formula to add up data from the values column, based solely on TRUE matches from the helper column.

This technique can be advantageous, depending on the number of rows of source data, the number of formulae referencing the table with SUMIFS functions, and the number and nature of match criteria.

15.5.8 Macros

While macro coding is not within the scope of this document, readers may benefit from understanding some basic tips to improve the processing speed of their macros.

Screen updating – In financial models, macros are often used to automate manual processes, such as repetitive copy/paste operations, and as such they do not require the user of the model to view the results in real time. Therefore it may be useful to temporarily pause updating of the screen, which will allow the macro to make its changes much faster because the computer no longer has to re-draw the screen after each operation. This can be achieved using the VBA code 'Application.ScreenUpdating = False' at the start of the macro. Screen updating should be turned back on at the end using the line 'Application.ScreenUpdating = True'.

We recommend using this technique on almost all macros as the time saving can be significant. However, be aware that if the macro crashes or is interrupted during processing then screen updating will remain turned off until the user turns it back on, or restarts Excel. As such this technique is best used on robustly tested macros, or those which include a suitable error handler to restore the setting in the case of an error.

Macros can also have a role enhancing processing speed for models. As well as the data table example noted above, for large data sets or major blocks of repeated calculations, macros can be used to scale those calculations efficiently. This involves using macros to replicate a seed row of calculations, to size them for a given data set before converting the calculated results to static figures to reduce file size and calculation volume. Using VBA and with a partial audit trail, and with careful control and management, this approach can reduce calculation times significantly.

15.6 Key points

- Choose faster formulae options where possible.
- Choose non-volatile formulae instead of volatile where possible.
- Match once, and index many times.
- Put the most restrictive parameters first in SUMIFS and COUNTIFS formulae.
- Don't be afraid of helper rows/columns.
- Avoid user defined functions (UDFs).
- Set calculation to 'semi automatic' (excludes data tables, if used) during development.
- Keep used range of cells to a minimum, as small as practicably possible, and clean up if the range extends far beyond the actual working area.
- When using macros, turn off screen updating, events, and calculation while the macro is running.
- If other techniques fail and a model calculates only slowly, set calculation to manual during development.

16 Model assurance

16.1 Overview

We have talked about model assurance from a process and timing perspective. In this chapter we provide an overview of:

- modelling risk;
- the reviewer role and timing for the review;
- the different approaches to model review that can be applied;
- the nature and use of review tools available to the model reviewer (both those native to Excel and third-party add-ins); and
- a list of some of the more common modelling errors.

16.2 Model risk

As highlighted in chapter 2, financial modelling can be a risky activity. Transactions only exacerbate this risk through time pressure, additional stakeholders and, typically, a wide breadth of analytical requirements. Quality assurance should therefore receive commensurately more attention in the modelling process.

A war story

A certain organisation re-used a standard investment appraisal template for all their deals. It transpired that certain key decision metrics, in particular the investment IRR calculation, had become hard coded with a static value. A number of deals were assessed and/ or transacted before the error was spotted and corrected.

Source: RSM

We summarised some of the risks that arise through the decision process in Figure 2. These risks are not unique to financial modelling but the difference is that financial modelling allows developers to make more errors more quickly.

Indeed, research into spreadsheet errors shows that the rate at which humans make errors when coding spreadsheets is little different from the rate of error in the wider computer programming world, and that that propensity to make mistakes is itself consistent with error rates in other spheres of human activity.

In other words, we humans are just as likely to mess things up with a spreadsheet as without – contrary to widely held perceptions. So how big a problem is it? Research into computer programming and spreadsheet programming error rates suggests that one in 20 lines of code will contain an error. Given most financial models contain thousands of lines of code, it can be inferred that pretty much all spreadsheet models will contain mistakes.[9]

The risks do not just affect the model development process, the process of review is also susceptible to human error: a modeller can potentially make lots of mistakes and the review process may yet miss some of them.

These statistics are somewhat scary, and rightly so. It is important that the role of model assurance and review is taken seriously.

On a more positive note, the risks are manageable:

- Through prevention: building the model according to best-practice principles reduces the risk of errors being introduced and facilitates the review process.
- Through review: by a party independent of the model, ideally by a team of reviewers, and preferably by applying a combination of review techniques.

Properly carried out, the development and review processes ought to surface any material errors in the model itself; bearing in mind the role of the model as an approximation of reality, this ought to provide sufficient assurance for most purposes.

9 A frequently quoted survey by Raymond Panko of the University of Hawai'i in 2008 concluded that 88% of spreadsheets reviewed had significant errors.

16.3 Role and timing

As noted, we advocate a separate review role in the model development process (preferably a completely independent third role rather than a subset of the model sponsor role). This independence is important, developers can become snow-blind and are not generally best equipped to review their own work. From a timing perspective, while we have shown a simple, linear model development process with review and testing taking place at the end of development; in practice, model review can be run (at least partly) in parallel to save time and potentially assist the development process.

16.4 Third-party review

It is often helpful to employ the services of a third party to provide model assurance services. Obvious advantages are the independence, experience and technical expertise available through the employment of such specialists. The precise scope of work required is a matter for the sponsor to consider, according to their assessment of risk in a given context.

16.4.1 Types of review available

Health check – Designed to give high level insight into the underlying structure of a model. Produced using Excel audit tools to produce some key summary information on the architecture and structure of the model. The results are considered in view of best-practice guidance and benchmark data, and include comments on the findings in that context. Does not include commentary on the correctness of any formulae within the model, or specifically review the model for errors.

Code review – A full code review includes a review of formula, functions and, if required, macro code within a model or suite of models. This work is more detailed than a high level health check, since it involves looking at every unique formulae[10] within a model. Queries on individual formulae may be referred back to the modeller for explanation or amendment and as such this type of review is often an iterative process as issues are identified by reviewer and addressed by the modeller.

Model audit – These can be carried out either through a code review, or an exhaustive parallel model build (see section 16.5.3). A model audit generally goes further than pure code review exercise (for example incorporating enhanced testing and review of tax and accounting), and will typically involve providing an assurance opinion as part of the deliverable. Although frequently referred to as a model audit it is important to understand that this is distinct from a statutory audit of a set of accounts. Generally the fee charged for a model audit will be greater than the other types of review, due to the wider scope involved and additional testing and risk associated with providing a formal assurance opinion.

16.4.2 Limitations

It is important that buyers of third party model assurance services understand what they will and will not receive as part of that service. In general terms it is a review to confirm that a model, under a given set of assumptions and relationships, will calculate the outputs accurately.

Limitations of such a review include:

- **Technical limitations** – A financial model will usually contain a large number of inputs, calculations, and outputs. There will often be hundreds of thousands of formulae, with thousands of unique formula. As such, the review of a model will make extensive use of technical review tools to leverage any efficiencies available through consistency across

10 A unique formula is a formula in Excel which is distinct in construction from the formulae above it, and to its immediate left on the spreadsheet. Looked at another way, if all the formulae in a block of cells are identical, a reviewer can gain comfort over the entire block from reviewing just that cell. It is in effect a unique identifying piece of code for that block of cells, albeit that the code is repeated in that block. The concept of unique formulae underpins most code inspection-based model review approaches.

areas of the model. As noted earlier, faced with the significant volume of formulae that typify most financial models, it is all but impossible to test a model to such an extent as to verify it is free from material error. Additionally, while very rare, on occasion model size and complexity can constrain the audit process through creating timetable or computational strains.

- **Assumptions** – A model review will tend not to include a comment on the reasonableness of the assumptions. These factors generally fall outside of the scope of a financial model review, with the review instead focusing on providing assurance that the assumptions are accurately reflected in the calculations in the model.
- **Viability** – Usually no comment will be made on the overall achievability of the forecasts or the viability of the project being modelled. There are many external factors which will influence the outturn of a project, which are beyond the control of management, and are impossible to quantify, and as such any opinion given would be caveated beyond any practical value.

A model audit is usually designed to attain a high level of assurance over those matters within the scope of that review. However, a model audit would not typically include an assessment of subjective matters and a model auditor would not usually form an overall opinion as to whether a model is fit for purpose.

Despite the best endeavours by a diligent and experienced model review team there is no guarantee that a model suite will be free from errors. In particular, it is impractical to test all permutations of inputs, and as such a review will necessarily be based on testing an agreed subset of potential inputs (for example, the base case and specific agreed sensitivities), with conclusions caveated appropriately.

The time required to review a model will normally be a function of the number and complexity of formula within a model, and as such a review of a complex model suite can be a time-consuming, and potentially expensive, exercise. However, there are certain techniques which can be used to increase review efficiency, for example by identifying similar formula blocks to reduce the number of unique formulae to be manually reviewed.

16.5 Approaches

When it comes to reviewing financial models, there are three broad approaches. They are not mutually exclusive and we would recommend using a combination of techniques. As a practical point, we would generally recommend undertaking a review using a copied version of the given model under review, particularly where tests are invasive or involve adding to, or amending, assumptions or calculations.

16.5.1 Analytical review

Analytical review refers to the review of a model and its calculations through reviewing, analysing, and considering the results it generates. This technique is most powerfully applied by the model sponsors – an intuitive understanding of the subject matter of a deal should flush out most material errors in a model.

However, such techniques are not infallible, we know of models that have been given a clean bill of health by analytical review, only for material errors to emerge when subjected to technical code review. In part this reflects the risk that users/reviewers can sometimes convince themselves that the model must be right, and find ways to rationalise a set of results.

It is beyond the scope of this note to set out a full financial due diligence work programme for assessing model outputs and forecasts (which would normally consider underlying assumptions and commercial bases). Rather we suggest some key checks that will help to highlight basic modelling errors and mistakes. When approaching analytical review, we suggest printing the primary statements and input sheets and reviewing them by eye.

- Scan the primary statements for unusual or unexpected peaks, troughs and trends, or infinitely accumulating balances.
- Look for covenant breaches and unusual ratio and output values.
- Review and sense check the interrelationships between the main inputs and outputs (for example, when comparing capex inputs with forecast capex spend, any variations should be explainable by reference to inflationary or sensitivity assumptions).
- Review and compare related items in the primary statements for consistency (eg, comparing EBITDA and operating cash flow, or depreciation with balance sheet fixed assets).
- Compare forecast period results with historical results for unexplained or unexplainable variations (eg, jumps in sales/spikes in working capital).
- Review line item row totals for overall sense (eg, check net assets and share capital plus reserves are equal).
- Carry out simple proof in total checks (eg, check implied tax rate by dividing aggregate tax through the forecast period with aggregate profit before tax (PBT)).
- Recalculate simple accounting ratios (such as debtor and creditor days) and sense check them against inputs and historical data.
- Zero the inputs and check that the model outputs are in turn all zeroed (a good test for flushing out errors from hard-coded values in formulae).
- Zero indexation and check inputs equal outputs.
- Run some basic sensitivities (eg, sales +10%) and check:
  - that the changes in outputs are directionally and proportionally consistent; and
  - that no unexpected/unintended items change.
- Run the main scenarios and review the results and outputs, comparing them as for sensitivities.

If, despite best intentions, time is limited for reviewing a model, then if the sponsor does nothing else, they should carry out analytical review of the outputs ahead of any other testing.

16.5.2 Code inspection

The second technique is a review of the actual programming code or logic of the underlying model. In practice, an Excel worksheet is made up of numerous individual formulae. The challenge this creates is that reviewing each and every one can be extremely time-consuming. However, the nature of spreadsheet programming risk is such that sample testing simply cannot provide adequate assurance over the model's coding, inevitably the bit that wasn't reviewed will contain the error. So in practice, while a code inspection could focus on known risk areas, it really has to review all the code.

To resolve this problem practically, the review burden has to be reduced to make it tractable, without resorting to sampling. You may hear modellers and model auditors refer to 'unique formulae' or 'distinct formulae'. Rather than review each individual cell in a row, they review the starting formula in that particular row. Assuming the row is constructed consistently, checking that single formula gives comfort over the entire row (or block, where the formula is repeated in contiguous rows).

If one pictures a model with a 10-year timeline calculating monthly cash flows, you can see that reviewing only one cell in each row can reduce the review burden by a factor of 120. This time-saving effect is heavily reliant on the model being constructed according to best practice, and consistent from left to right within the calculation block. Applying this approach effectively also necessitates comprehensively identifying the unique formulae. Fortunately there are a number of tools commercially available to assist in this task (see section 16.6.4).

The main risks with code inspection are that the reviewer can become fatigued. They may be unable to see the wood for the trees or begin to suffer from a form of Stockholm Syndrome, believing they know what the modeller was thinking or intending, and failing to challenge it effectively. These risks can be mitigated through combining code inspection with other review disciplines, and by deploying a team of reviewers.

The big advantages with the code inspection approach include that it provides specific targeted insight into a model's coding, and can therefore assist with the development and debugging process. From a process perspective, the volume of work required can be reliably estimated, which is helpful for budgeting and project planning. Finally, it can be good at spotting contingent errors. These are coding errors that only become apparent under certain usage conditions/combinations of inputs. They are among the hardest errors to identify: sensitivity testing may also flush them out, and combined with code inspection, ought to assist in identifying problems in this area.

16.5.3 Reperformance

The third approach to model review is to build a parallel model (sometimes referred to as a shadow model) using the same input set. The output from the parallel model is then reviewed to see if the same results are generated. Building a separate model will highlight areas of difference, and the act of adjusting the models to bring them into sync by converging the results ought to identify major errors. While this approach can be very powerful, its shortcomings are that it can be time-consuming and expensive (possibly unpredictably so, unless the subject matter is generic); it may not detect contingent errors; it will not usually provide comprehensive specific cell-level feedback; and the shadow modeller runs the risk of going native, adopting similar approaches to the initial modeller, resulting in similar oversights.

This approach works best, where the subject matter is generic or repeated (for instance it has been widely used in the private finance initiative (PFI) sector) or where the model is well documented. For complex/voluminous subject matter models it is best restricted to specific targeted areas of the model under review as part of a wider program of analytical review.

Indeed, to reinforce the point made earlier – each of these techniques has its pros and cons and we advocate a combination approach. Reperformance testing can be applied to support proof in total testing carried out within an analytical review context – rebuilding specific workings to test and understand them. After all, what is a full parallel rebuild of a model if not an extended proof in total test?

16.6 Review tools

In the previous section we overviewed the main techniques available to the model reviewer. Here we consider the tools available to assist the reviewer, looking at two categories: native Excel tools, and third party spreadsheet add-in tools.

16.6.1 Native Excel review tools

Excel has a number of inbuilt tools to assist in reviewing spreadsheet models. It is not the purpose of this note to provide a detailed teach-in on their use; instead we focus on a handful of the most important tools and their key features:

Trace precedents/dependents

Under the formula menu on the ribbon, Excel provides tools to trace precedents (the cells on which a formula in the active cell depends), or dependents (the cells that depend on the active cell). Using the buttons under this section of the ribbon, Excel provides tracer arrows highlighting the flow and direction of logic. Double clicking on the blue arrows will take you to the cells at either end.

In addition to the buttons, Excel provides keyboard shortcuts (see Appendix C) to allow the user to jump to precedents/dependents on the same sheet as the active cell: 'ctrl+[' to jump to precedents; and 'ctrl+]' to jump to dependents.

As noted above (and illustrated in Figure 34) 'edit directly in cell' can be disabled, after which a simple double click on a cell will also jump to its precedents. These functions allow the user or reviewer to see what cells refer to or are referred to by the active cell.

Highlighting row and column differences

Excel provides the ability to quickly identify inconsistencies within a block of formulae. Selecting a calculation block,and applying row differences will reveal left-right inconsistencies: highlighting all formulae after the first break in consistency. Similarly, applying column differences reveals vertical breaks in consistency.

The buttons and menu for highlighting row/column differences can be found on the ribbon under 'find and select,' under the home tab. However, Excel's keyboard shortcuts are much quicker to access and use. The relevant shortcuts are: 'ctrl + \' to highlight row differences; and 'ctrl + shift + \' to highlight column differences.

The 'highlighting differences' functionality can also be used for non-contiguous formulae cells that are consistent. This can be a significant time saver for model reviewers. Most of the third-party review tools available will show each non-contiguous (but consistent) formula row as a unique formula – which means they each need to be individually inspected. Row and column differences can be used to quickly demonstrate that they are consistent, and to reduce the review burden to the first instance (though the reviewer still has to satisfy themselves that the repetition of the formulae is correctly implemented/appropriate).

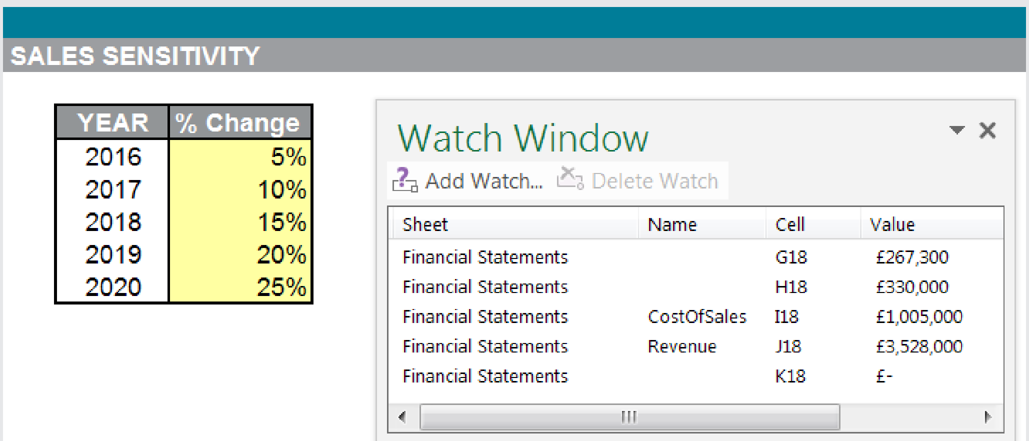
It is also straightforward to do. If you have consistent formulae separated by blank rows, select the entire range starting with one of the formulae rows and encompassing the subsequent consistent formulae rows and blank rows. Next run column differences and then apply a colour format to highlight the inconsistencies revealed by the column differences test. This should just show blank rows recoloured – if any of the formulae cells in the range are highlighted, there is a break in consistency.

Watch window

Excel allows the user to set up a 'watch window' containing key cell results to monitor. The 'watch window' remains visible whatever sheet the user/reviewer is on. It allows the user to monitor the changes in specific cells when changes are made or sensitivities run. It can be found under the formulas tab in the ribbon.

Excel provides other automated error checking tools, but a detailed consideration of these is beyond the scope of this guideline.

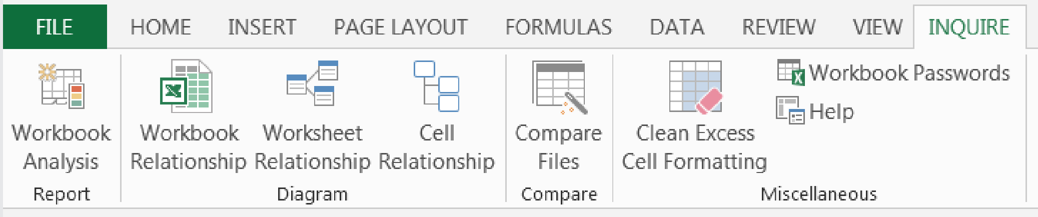
Figure 46: Using a watch window to monitor key outputs



16.6.2 Inquire add-in

Excel 2013 includes an add-in called 'inquire' which provides assistance in reviewing spreadsheets. Inquire includes many of the same functions provided by third-party spreadsheet auditing software.

Figure 47: Inquire add-in



Activating 'inquire'

Although 'inquire' is included within Excel 2013, it is not activated by default. To activate 'inquire' go to 'file > options > add-ins,' and in the 'manage' list, select 'COM add-ins,' and click 'go'. In the resulting pop-up window tick the 'inquire' check box followed by 'ok'.

Overview of 'inquire' functions

Workbook analysis report: This feature creates a detailed report for the active file, including: formulae used, formulae which evaluate to errors, sheet visibility, cell contents, named ranges, and details of hidden rows/columns, etc. These outputs are helpful in reviewing a workbook for common problems like formulae which evaluate to error, as well as adherence to best practice, for example by identifying cells which included hard-coded values in their formulae.

Relationship diagrams: The 'workbook relationship diagram' creates an interactive, graphical map of the workbook connections, including to other workbooks, databases, text files, HTML pages, SQL servers, and other data sources. The 'worksheet relationship diagram' works similarly, presenting a map, at a worksheet level, between each sheet in the active workbook and worksheets they are linked to (including sheets in other workbooks). The 'cell relationship diagram' creates a detailed graphical map of the links between the selected cells, with capabilities to show cell links through other worksheets, and workbooks. These functions can be useful in mapping a model, to show connections between worksheets in a workbook, or between files in a model suite.

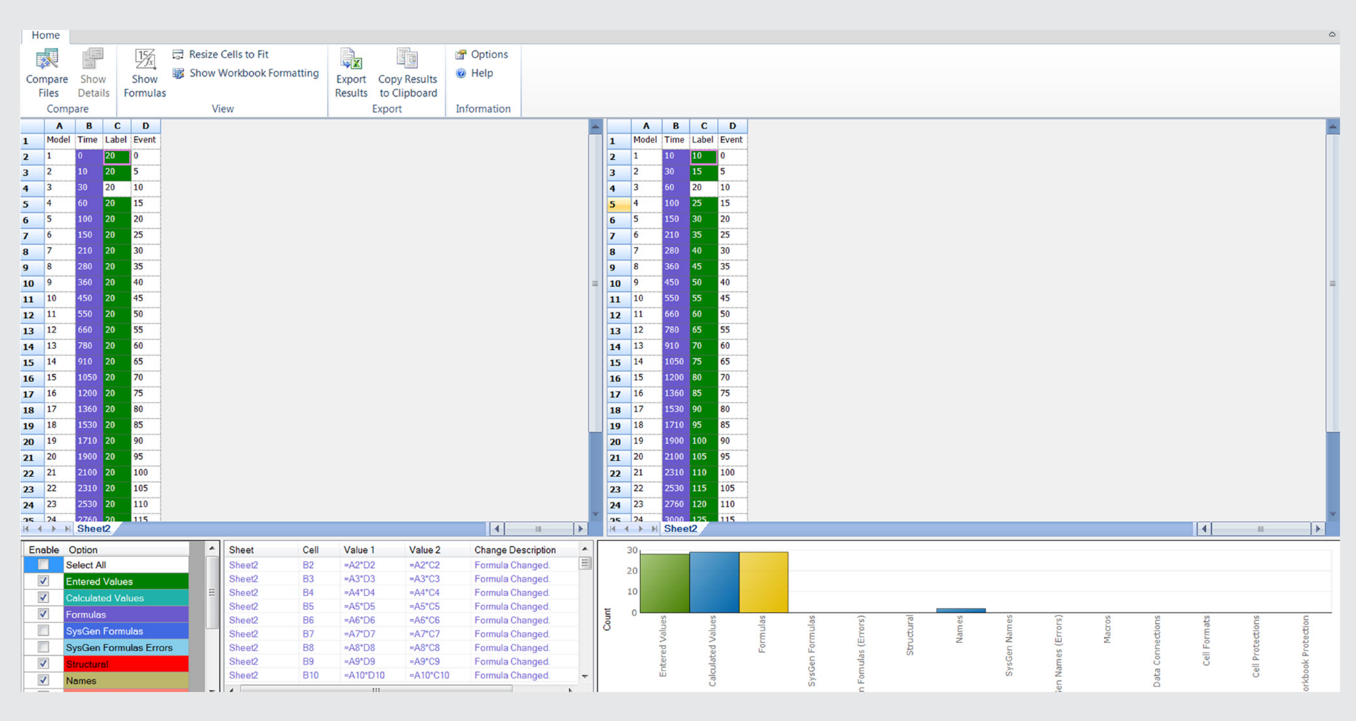
Compare files: This feature allows the user to compare two open files and to see, simultaneously, the differences, on a cell by cell basis, between the two files (see figure 48). Colour coding is used to identify the nature of the change. This interactive tool can be useful in identifying changes between versions of a model.

Clean excess cell formatting: This command removes excess formatting from beyond the actual 'used range' of a worksheet. The feature can help to reduce unnecessary size and calculation time in Excel files. Note, that it is not possible to undo the effects of this command.

Workbook passwords: The 'inquire workbook analysis' runs by opening the last saved version of your document in the background, and it therefore requires the passwords of workbooks saved with password protection enabled, in order to access their contents. Therefore, if running the 'inquire workbook analysis' on a password protected file you must first store the password into the password bank, so that the add-in can open and decrypt the file to analyse.

The 'inquire' tab also comes with a built-in help command, which provides further details on its functionality, and specific examples of all the commands included within 'inquire'.

Figure 48: Inquire add-in functionality



16.6.3 User-created review tools

The standard suite of Excel provides certain tools that are useful for reviewing models and their outputs, however, there are other techniques that can be applied to create your own review tools. In the following section we have included some examples of what can be created to assist in your review process.

Variance sheet

Although Excel provides similar functionality by way of the 'watch window,' and the 'scenario manager,' it can be useful to create your own 'variance sheet'. Consider the example of an integrated financial model for a potential project, whereby you wish to evaluate the impact of changes to the input data on your financial statements.

Create two blank sheets within the model. On the first, paste a copy of the financial statements first as values, and then paste over your formatting – this will be your 'base data'. On the second, do the same, but replace each value with a formula which subtracts the associated figure from your base financial statements from the equivalent figure in your live formula-driven financial statements. This variance sheet will illustrate the impact on the financial statements of modelling changes, and changes to input data, and can therefore be used to test both. The base data will need to be reset periodically, to enable review of new changes in isolation from the effects of previous changes, and the base and variance sheets will need to be updated if the format of the financial statements is changed.

Analytical review charts

It can often be easier to identify issues in a financial model by using visual aids, rather than reviewing a sea of numbers and hoping to spot a trend or relationship between figures in rows. It can therefore be much more powerful and intuitive to chart, for example, analytical ratios or the numbers in your income statement, and to identify and review trends. By doing so, it can be much easier to see whether debtor days are spiralling out of control, or that cash reserves are draining dry.

An efficient method of doing this is to create a 'chart data' sheet within the model which links to various financial statement outputs. Charts can then be added to illustrate common ratios or relationships between variables. (For example charting sales, trade

A war story

A factory was built approximately 50% too large due to end-market estimation errors arising from mistakes in coding foreign exchange. A thorough process of analytical review will help to mitigate the risks of undetected coding or estimation errors.

Source: RSM

debtors and the debtor days assumption will help to illustrate if these flows, balances and assumptions are moving in sync as expected.) By incorporating a set of charts which link to model outputs they can remain in the model for a quick and easy verification of these tests as the model is developed and used.

It may be possible for your organisation to develop a standard suite of analytical review tests to perform by way of ratio analysis and charts. This set of tests can be built into your standard base model to be available for all projects. Alternatively, a standard set of tests could be set up on a standalone sheet which can be dropped into any given model and wired up to the outputs of that model to provide a standard view which can be reproduced quickly.

16.6.4 Spreadsheet add-in tools

In addition to the inbuilt tools, there are a number of commercially available add-ins. We do not review and compare them in detail; instead we focus on describing the key generic capabilities such add-ins provide. This will help users to understand the available functionality and why it is useful, and then help to filter and identify the right package to meet their needs. Such tools can help by:

- **mapping.** Generating a visual representation of the structure and logic of a given model (for example, highlighting inputs and labels versus calculations, identifying unique formulae, showing calculation blocks, and highlighting inconsistencies).
- **listing.** Producing lists of the unique formulae in a model, of the inputs in that model, and of the named ranges in a model. This helps to identify and to categorise the guts of a model, and to monitor and focus review effort.
- **comparing.** Providing a comparison of two versions of a model, and identifying changes at the formula level. This can be helpful as models mature, allowing the user/reviewer to identify only those aspects that have changed, and to review incremental changes on an exceptions basis.
- **tracing.** Extending Excel's 'trace precedents/dependents' functionality to generate visual representations (tree diagrams), or interactive listings of a cell's parents and children. This facilitates the review of complex formulae and an understanding of the flow of logic.

This is not an exhaustive list and readers should review the available tools and select the package with the functionality most appropriate to their needs. Excel 2013 introduced an add-in named 'inquire' which contains some useful functions for model reviews (see section 16.6.1). We, and a number of other leading firms, also use proprietary model review tools.

16.7 Common issues and errors

Finally, in relation to model review, we have listed below a few of the classic mistakes to look out for:

- **sum insertions:** if you have a sum range and insert a new row beneath it to be summed, the SUM function will not automatically pick it up. It is usually worth looking closely at SUM ranges in formulae to check for orphan rows and columns.
- **IRR/NPV:** Excel's IRR and NPV functions are a frequent source of error or misstatement, the two most common being:
  - Forgetting that they treat the first cash flow as happening at t=1 not t=0 and failing to adjust accordingly if required.
  - Not being aware that blank cells or cells returning a blank result are skipped in the calculation.

Both problems can be avoided by enabling the 'analysis toolpak' add-in, which is shipped with Excel, and by using the XIRR and XNPV functions (see section 13.5.3).

- **hidden space/white text:** check for hidden worksheets/external links or white text on white background in the active worksheet area. It may be being used to conceal data. More benignly, it may just be presentational but runs the risk that it is accidentally deleted.
- **plug figures:** check the balance sheet for calculations that appear to compute a difference between net assets and reserves items, for static cash balance, and for balances that balloon disproportionately over the forecast period.
- **hard-coded results:** check for obviously pasted results (typically they have a long string of numbers after the decimal point). Are they intentional or accidental?
- **time period conversions:** swapping between years, months, and quarters can be a source of error. In particular with regard to treatment of interest (for example, decompounding), and working capital calculations. Furthermore, the treatment of part period adjustments (for example, to allow for a financial close date falling mid-way through a modelled time period) is also a frequent source of problems.

The other areas where we commonly see mistakes are:

- **depreciation:** over-depreciation is quite common where appropriate error checks/ stops have not been included in the fixed asset calculation. Check that aggregate depreciation is less than or equal to the aggregate fixed asset additions and brought forward assets over the forecast period.
- **indexation:** ie, inflation calculations. Are they stepping up correctly? Are they applied consistently? Are they properly adjusted for the model's timeline units (months, quarters, etc)?
- **taxation:** in particular, deferred taxation and treatment of rolled-up interest. Can you carry out a proof in total on the aggregate tax charges? Incorrect tax charges in loss-making periods (eg, tax shown as an inflow). Calculation for tax charge or payment being based on incorrect time periods.
- **distributions:** Profit available for distribution being calculated on earnings before tax, instead of on net profit.
- **interest:** in particular the translation of annual rates into quarterly or monthly rates. Is the method applied internally consistent with the quoted interest basis? Are interest conventions (such as ACT/360) correctly applied? Is the interest correctly applied to the period average balance?
- **incomplete lists:** as models evolve over time, it is not uncommon for new inputs to be added to validation lists. On occasion, named ranges are not updated to accommodate the growing list. It can be useful to implement 'dynamic named ranges' that extend as new list items are added to the list, creating a more future-proofed range.
- **sensitivities:** overlays not working consistently with each other, or not compounding correctly.

A war story

In 2005 an Australian bank was forced to halt trading on its shares and declare results a day early after accidentally releasing confidential results. Details of a $2.818 billion record profit were held within a spreadsheet and shared with brokerage analysts in a spreadsheet intended to hold historical results. Some news reports indicated an employee had thought that a black cell background fill would hide black text.

Source: www.cio.com

16.8 Key points

- Financial models are no less susceptible to human error than any other human activity.
- There is a high risk that any particular model contains errors (of varying severity).
- Errors can be reduced through prevention, applying best-practice techniques and framework, and by review, to screen and test for potential mistakes.
- There are a variety of model review techniques, each with advantages as well as disadvantages. We advocate a combination approach focusing on analytical review and code inspection.
- A variety of review tools (both native to Excel and developed by third parties) can be used to facilitate and to speed up the review process.
- There are a number of common mistakes that can usefully be incorporated into a programme of model review work.

Appendix A: Twenty principles for good spreadsheet practice

A1 The twenty principles in brief

The guidance given in this document is compatible, and consistent, with *Twenty Principles for Good Spreadsheet Practice*, published by ICAEW's IT Faculty in 2014, and summarised below. The full text of *Twenty Principles for Good Spreadsheet Practice* is available from ICAEW's IT Faculty website[11], and the publication's core principles are reproduced below under the three core headings.

The spreadsheet's business environment

1. Determine what role spreadsheets play in your business, and plan your spreadsheet standards and processes accordingly.
2. Adopt a standard for your organisation and stick to it.
3. Ensure that everyone involved in the creation or use of spreadsheets has an appropriate level of knowledge and competence.
4. Work collaboratively, share ownership, peer review.

Designing and building your spreadsheet

5. Before starting, satisfy yourself that a spreadsheet is the appropriate tool for the job.
6. Identify the audience. If a spreadsheet is intended to be understood and used by others, the design should facilitate this.
7. Include an 'about' or 'welcome' sheet to document the spreadsheet.
8. Design for longevity.
9. Focus on the required outputs.
10. Separate and clearly identify inputs, workings, and outputs.
11. Be consistent in structure.
12. Be consistent in the use of formulae.
13. Keep formulae as short and simple as practicable.
14. Never embed in a formula anything that might change or need to be changed.
15. Perform a calculation once and then refer back to that calculation.
16. Avoid using advanced features where simpler features could achieve the same result.

Spreadsheet risks and controls

17. Have a system of backup and version control, which should be applied consistently within an organisation.
18. Rigorously test the workbook.
19. Build in checks, controls, and alerts from the outset, and during the course of spreadsheet design.
20. Protect parts of the workbook that are not supposed to be changed by users.

11 http://www.icaew.com/technical/information-technology/excel/twenty-principles

Appendix B: New features in Excel 2013 and 2016

A number of readers will be familiar with the functionality of earlier versions of Excel. However, they may not be so familiar with more recent additions to the standard range of Excel functions. With this is mind, we want to bring to readers' attention some interesting new features that have been announced on the Microsoft support website[12], and made accessible in standard versions of Excel 2013 and Excel 2016, and which are relevant to a corporate finance audience.

B1 Excel 2013

- Start screen – Offers recent documents and templates when Excel starts.
- Quick analysis tool – Lets you convert your data into a chart or table in two steps or fewer. Preview your data with conditional formatting, sparklines, or charts, and apply your choice in one click.
- Recommended charts – Based on your data, Excel will offer chart recommendations, which you can click on to see a preview of that chart type using your data, allowing you a more informed choice before committing.
- One window per workbook – This feature replicates the toolbar and other menus at the top of each workbook window and allows them to be re-arranged on the desktop. This makes life a lot easier when working with multiple monitors.
- New functions – There are several new functions in the math and trigonometry, statistical, engineering, date and time, lookup and reference, logical, and text function categories.
- Share an Excel worksheet in an online meeting – You can connect to and share a workbook through Lync online meetings.
- Power map – In Excel 2013 and Office 365 Pro Plus, you can take advantage of 'power map,' which is a three-dimensional data visualisation tool that lets you overlay data onto a map using geographic and time-based data. 'Power map' is built into Office 365 Pro Plus, but for Office 2013 or Excel 2013 you can download a preview version.

New and improved add-ins and converters

- Inquire add-in – In Office Professional Plus 2013 or Office 365 Pro Plus, the 'inquire' add-in comes installed with Excel and helps you to analyse and review your workbooks, to understand their design, function, and data dependencies, and to uncover a variety of problems including formula errors or inconsistencies, hidden information, and broken links.
- Spreadsheet compare – From 'inquire,' you can start a new Microsoft Office tool, called 'spreadsheet compare,' to compare two versions of a workbook, clearly indicating where changes have occurred.

12 New features in Excel 2013: https://support.office.com/en-us/Article/What-s-new-in-Excel-2013-1cbc42cd-bfaf-43d7-9031-5688ef1392fd; and Excel 2016: https://support.office.com/en-us/article/What-s-new-in-Excel-2016-for-Windows-5fdb9208-ff33-45b6-9e08-1f5cdb3a6c73

B2 Excel 2016

- Improved autocomplete – Previously when typing the first few characters of a function, only functions whose names start with those letters are suggested in the formula bar. This improvement returns all functions which feature those letters. For example, typing =DAYS will bring up =NETWORKDAYS as a suggestion for autocomplete.
- TEXTJOIN – Combines text from multiple ranges, and each item is separated by a delimiter that you specify. Helpful in creating lookup values from multiple labels.
- CONCAT – Similar to the existing CONCATENATE function, but also supports ranges as inputs rather than just single cells.
- IFS – A function to replace nested IF functions.
- MINIFS / MAXIFS – Similar to SUMIFS in that they will return values based on a subset of data which meets certain criteria.
- Waterfall charts – Is now a native chart type within Excel.
- Other charts include treemap, sunburst, histogram, and box and whisker.
- Trend forecasting – Excel forecast values based on historical trends.
- Financial templates – Including cash flow and stock analysis templates.
- Quick Shape Formatting – Increases the number of default shape styles by introducing new 'preset' styles in Excel.

Appendix C: Excel shortcuts 2007 – 2016

NAVIGATION

Home	Beginning of row
Ctrl + Home	Beginning of worksheet
Page Up/Down	Moves vertically in worksheet
Alt + Page Up/Down	Moves horizontally in worksheet
Ctrl + Page Up/Down	Moves between workbooks
Alt + Tab	Cycle between open applications
Ctrl + Tab	Cycle between workbooks
Ctrl + W	Close the active window
Alt + F4	Exit Excel
Ctrl + F / Ctrl + H	Find/Replace
Ctrl + [	Jump to source of link
F5 + Enter	Return from last jump
Ctrl + F1	Toggle ribbon visibility

SELECTING CELLS

Ctrl + .	Cycle around corners of the selected range
Ctrl + Spacebar	Select the entire column
Shift + Spacebar	Select entire row
Ctrl + A or Ctrl + shift and spacebar	Select the current region
Ctrl + A twice or Ctrl + shift and spacebar twice	Select the entire worksheet
Shift + Arrows	Adjust the selection
Press Shift + any of the following keystrokes to select text/cells (where appropriate)	
Ctrl + Arrows	Edge of data region
Ctrl + End	Last used cell in worksheet

EDITING

Ctrl + R	Fill right
Ctrl + D	Fill down
F7	Spell checker
Shift + F7	Thesaurus
Ctrl + C, Enter	Copy & paste (clears clipboard)
Ctrl + X, Enter	Cut & paste (clears clipboard)
Ctrl + Z	Undo
Ctrl + Y or F4	Redo
Alt, H, V...	Paste special
+ S,T	Formats
+ F	Formulas
+ V	Values
+ B	All except borders
+ N	Link
F2	Toggle edit and point mode
F4	Toggle anchoring in cell
Shift F2	Edit cell comment
Alt + =	Sum adjoining range

AUDITING AND REVIEWING

F11	Quick chart
Alt, M...	Trace arrows
+ P	Precedents
+ D	Dependents
+ A,A	Remove arrows
Ctrl + ¬	Display formulae in worksheet
Ctrl + \	In a selected row, select the cells that don't match the formula or static value in the active cell
Ctrl + Shift + \	In a selected column, select the cells that don't match the formula or static value in the active cell
F3	Paste name list
Ctrl +]	Select cells that contain formulae that directly reference the active cell
Ctrl + Shift +]	Select cells that contain formulae that directly or indirectly reference the active cell
Ctrl + Shift + L	Creates an autofilter for selected cells
Ctrl + Shift + U	Expands and collapses the formula bar

STYLES AND FORMAT

Alt + '	Display the Style dialog box
Ctrl + Shift + ~	Apply the General number format
Ctrl + Shift + $	Apply the Currency format with two decimal places
Ctrl + Shift + %	Apply the Percentage format
Ctrl + #	Apply the Date format with the day, month and year
Ctrl + Shift + !	Apply the Number format with two decimal places, thousand separator, and minus sign (-) for negative values
Shift F10	Equivalent to a right click on the mouse

FORMAT OPERATIONS

Ctrl + 1	Format cells dialogue box
Ctrl + 5	Strikethrough (apply or remove)

Appendix D: Example statement of model requirements

D1 Overview

Note: this appendix is not intended to be used as a template engagement agreement.

Introduction

[•] have been appointed to develop a financial model of Project [•], and the first stage of the modelling process involved holding a requirements workshop on [DATE] to agree the key business issues for the project and their implications for the design of the financial model.

This paper represents the next stage of the modelling process; its purpose is to specify an agreed structure for the model, so as to allow a competent financial modeller to develop a model which will be recognised by the project team as appropriate to their requirements and objectives identified at the workshop.

The specification deals with the scope and structure rather than the data content of the model; consequently it will still be necessary for the modeller to hold conversations with appropriate members of the project team on specific matters of fact.

Objectives

[The Project involves the development and roll-out of a chain of themed restaurants funded by private equity backing].

[The business issues are therefore to determine and evaluate the cash flow profile for the business and the funding requirements; allowing management and the equity funders to assess a range of likely planning scenarios and to determine how best to fund the Project].

In our understanding, the purpose of the financial model is to achieve the following objectives, while using the simplest structure permitting inclusion of all significant factors affecting the Project:

1. [help determine the Project's funding requirements and the most appropriate financing structure];
2. [provide a best estimate model of the costs and returns of the Project];
3. [perform sensitivity studies, varying model assumptions by changing input data only];
4. [calculate discounted cash flow (DCF) and multiples-based valuations of the Project, which will form the main basis for assessing value and likely equity returns];
5. [prepare integrated financial projections requested in order to evaluate the business];
6. [allow monitoring and analysis of the company's financial situation and performance over the forecast period].

This specification describes a model which it is believed will meet these objectives as understood at the requirements workshop.

Structure

The model will consist of a single Excel [2013] workbook file, which will contain four sets of worksheets:

- information sheets, including model contents, map/layout and navigation, format key, and disclaimer;
- input data sheets, including scenario control inputs, time-based inputs, static (ie, standing data and/or non-time-based) inputs, and sensitivity factors, and historic data;

- calculation sheets, containing the formulas and coding required to generate the model outputs from the input data;
- output report sheets, containing the reports required to monitor and evaluate the results of the Project and in particular the project 'dashboard' summarising the key data points for a given scenario.

Each sheet will be constructed in a modular fashion to indicate clearly the separate blocks of inputs, calculations, and reports. A block diagram of the overall structure of the model is attached as Annex [•] (example in Figure 49).

Quality assurance and documentation

As well as separating inputs, calculations and outputs, the model will be constructed according to the principles of simplicity, consistency and integrity. It will include a sheet containing check totals for net assets/capital, cash, retained profit and other items as appropriate, to enable quick verification of the integrity of the model.

The model will be reviewed before release, using specialist auditing software, to confirm that the quality principles listed above have been appropriately applied.

In addition to the model itself, the following documentation will be provided:

- a data book, listing the assumptions made in the model, and the sources of all input data and assumptions; and
- a user guide containing instructions for operating the model and an explanation of the principles of its structure.

Scope and functionality

The periodicity of the model will be annual, with [2016] as the first year. Static historical data for the previous three years will be included.

The model will be laid out so that high quality reports can be printed directly from the 'print' command, and consequently there will be no requirement for separate print macros or reporting add-ins.

The model will be designed for clarity and ease of use, but staff operating the model will nevertheless require an understanding of the project and a basic working knowledge of Excel.

The model will contain flags and/or sensitivity drivers to enable some common sensitivities to be carried out very simply without losing the base data. It is intended that sensitivity runs and complex multi-factor scenarios should be managed and stored within a single file using the scenario inputs worksheet. Instructions for its use will be included within the user guide.

Rather than varying an input data item to ascertain the effect on the project valuation, it may be more useful to ascertain what change in the input data will produce a specified project valuation. In order to allow the use of Excel's 'goal seek' facility for this purpose, each input sheet of the model will contain a small 'dashboard' area showing the project value under the current scenario.

Sensitivities and scenarios

The key sensitivities and scenarios catered for by the facilities described above will be:

- [speed and volume of restaurant rollout
- mix of restaurant types;
- ....]

Some sensitivities and scenarios which will be outside the scope of the model are:

- [addition of overseas units;
- ....]

Next steps

This specification is being sent to all members of the project team for approval, and is expected to be finalised by [2016]. Development of the model will begin immediately after this, and is expected to be completed by [2016]. The project team will be involved with much of the development work, and will be kept informed of progress on a regular basis. Further details of scheduling are provided in the management plan attached as Appendix [•].

D2 Outputs

All outputs will be given in nominal terms. Financial statements will be presented in line with [IFRS].

Unless stated otherwise all outputs will be in £m. The general sign convention will be 'positive is normal'.

Income statement

- [Revenue from food, beverages
- Operating expenses:
  - payroll costs;
  - rent;
  - ....
- ....
- Retained profit for the year
- Cumulative retained profits]

Balance sheet

- [Fixed assets:
  - cost, accumulated depreciation and net book value;
  - categories as per management information available.
- Current assets:
  - cash;
  - ....
- ....
- Total net assets
- ....
- Total shareholders' funds]

Financing and investment appraisal

- [Summary of each debt tranche:
  - outstanding principal;
  - drawdowns and repayments;
  - ....
- Annual debt covenants
- Discounted cash flow valuation
- Exit multiples and IRR
- Graphical illustrations of key data trends
- ....]

Cash flow statement

- [Receipts:
  - ....
- Payments:
  - ....
- Cash balance]

Summary sheet

- [Key operating assumptions:
  - ....
- Key financing and investment appraisal values:
  - ....
- Key figures from financial statements:
  - ....]

D3 Inputs

Where annual inputs are referred to this means that an input is required for each year of the investment appraisal period.

Unless stated otherwise all inputs will be in £m. The general sign convention will be positive is normal.

Commercial/operational

Input data in this section is owned by ....

- [Number of covers
- Average spend per cover
- Direct cost of sales
- Employee costs
- Rental costs
- Fit-out costs
- Other costs
- ....]

Accounting and taxation

Input data in this section is owned by ....

- [Opening balance sheet data:
- as for balance sheet categories in outputs
- Debtor/Creditor/Stock days (no. of days)
- Capital expenditure
- Tax rates (%)
- ....]

Financing and investment appraisal

Input data in this section is owned by ….

- [Annual LIBOR rates (%)
- Total facility
- Interest rate margins over LIBOR (%)
- Repayment profile
- Associated fees
- Discount rates (%)
- ….]

Economic

Input data in this section is owned by ….

- [Exchange rates $/£
- ….]

D4 Calculations

Commercial/operational restaurant module

This will calculate:

- [the revenues, gross margin and EBITDA profile for each restaurant type over its lifecycle;
- the development of the portfolio over time and number of restaurants active in each period;
- the revenues, gross margin and EBITDA profile for the portfolio, classified by restaurant theme; and
- ….]

Inflation module

Annual inflation factors will be calculated for general inflation, wage inflation and capex inflation. These will assume that the uplift for input inflation rates is applied at the start of the relevant period.

Capex module

This module will aggregate all capital expenditure (excluding any categories switched out for a particular scenario) under appropriate fixed asset categories as set out in the management accounts.

Fixed asset module

Fixed assets are depreciated on a straight line basis at the appropriate input rates, and additions in each year are depreciated separately to avoid over-depreciation. Disposals are similarly identified as to the year of purchase of the assets, as are the opening balances of cost and accumulated depreciation. There is no facility for revaluation of assets.

Cash module

There are two parts to the cash calculation (which is separate from the cash flow statement included in the reports):

- A detailed list of cash flows in the year (including interest as worked out below) is used to calculate the cash balance at the end of each year;
- A parallel calculation using the indirect method is also included.

Starting from the same list of cash flows (but excluding interest), factors are applied taking into account the timing of the flows to give a time weighted average cash balance on which interest can then be calculated.

Financing and interest module

Interest will be calculated on an annual basis, using an average cash balance, consisting of the opening cash balance plus time-weighted cash flows for the year. It will not be grossed-up for interest on interest.

Investment and valuation module

The model will calculate valuations based on NPV and assumed exit multiples for a given range. These will be used to generate investor returns and cash on cash multiples.

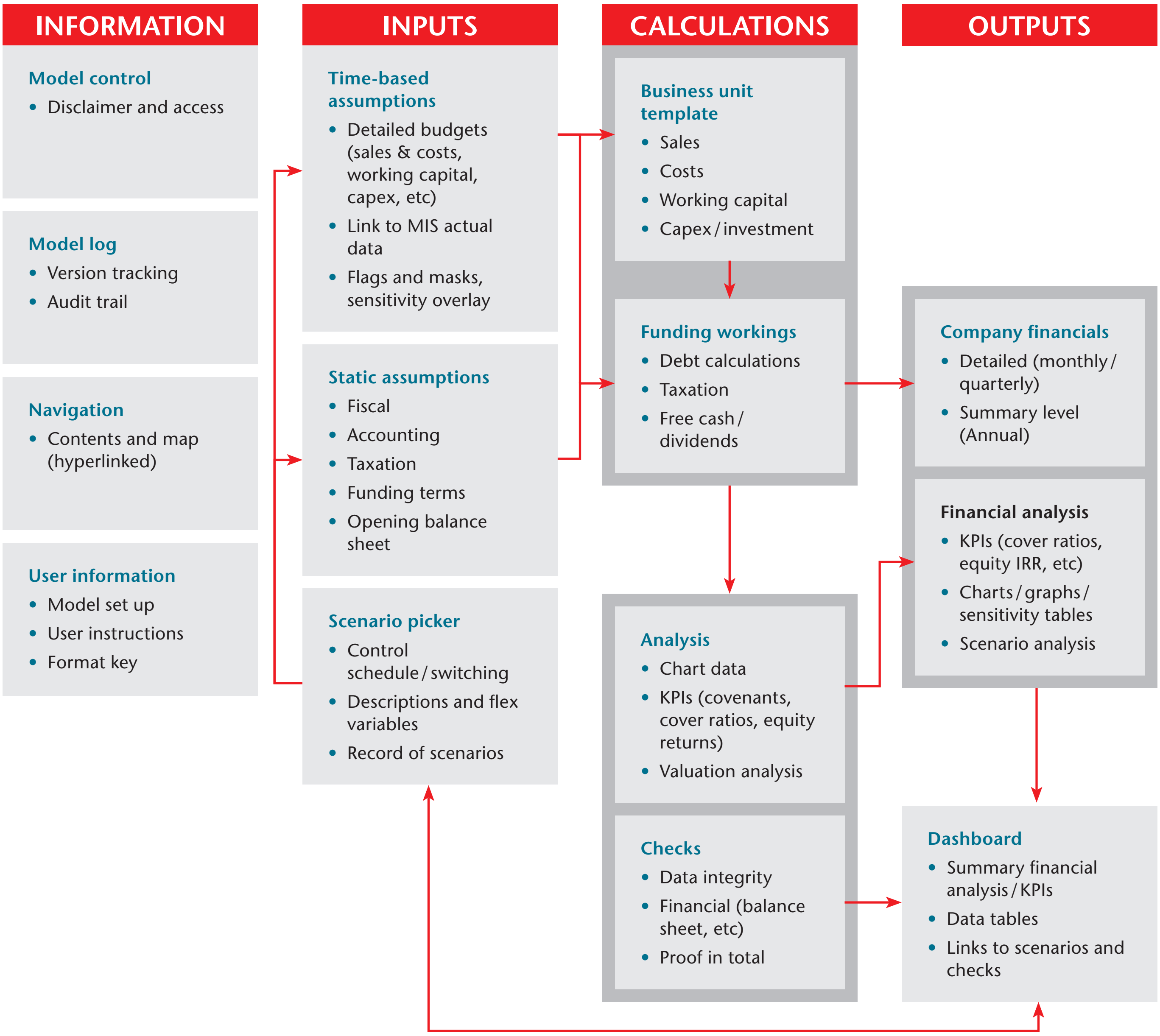
The calculation of net present value will assume period end cash flows. A terminal value will be calculated on the basis of the final year cash flow adjusted for ....

Tax, accounting and other modules

[…]

D5 Schematic of model structure

Figure 49: Schematic of a model structure



D6 Project plan

[*Includes details of key milestones and timings.*]

[....]

Appendix E: Glossary

Active cell
The cell which is selected on the current sheet, in which data is entered when you type.

Anchoring
The process of fixing cell references to specific cells, rows, or columns, preventing the reference from updating as formulae are copied down or across. This is achieved using the application of the '$' symbol before the column/row references.

For example, '$A1' would fix the reference to column A, but allow the row number to be updated as you copy the formula down. Updating this to $A$1 would then also fix the row number, resulting in formulae referring to the same specific cell, regardless of where in the workbook the formula was copied.

Call-ups, links
The use of simple links to reproduce data from input sheets in other locations within the model, typically for use in a calculation.

Circular reference
The circumstance where a calculation in a cell depends in turn on its own result (whether directly or indirectly).

Dashboard
A user-friendly executive summary of a model's outputs. Frequently, these include the functionality to change key inputs to determine the effect on the model.

Dependents
The dependents of a cell are those cells whose value depends on the value of the given cell. Ie, they reference the given cell directly, or else through one of its intermediate dependents.

Dynamic named range
Similar to a standard named range in Excel, however as new items are added to the list, a dynamic named range will automatically expand to include them.

Excel add-in
Modular software additions to Excel that can be installed in order to complement the standard features of Excel and to provide additional functionality for the user.

Hard-coded value
A hard-coded value in a cell is a number which is typed in, rather than the result of a link or calculation.

A hard-coded value in a formula is a number which is used within a calculation, but has not been referenced or identified as input data. For example, the use of a number such as 1,000 in the formula '=A1*1000'.

Hard-wired calculation
An implicit and/or fixed assumption regarding the mathematical relationship or computational treatment of a modelling item embedded within a model's logic.

For example, if a loan calculation bases interest on the opening balance plus any drawdown, there is a hard-wired assumption that all drawdowns take place at the start of the given time period. Conversely, if the interest is calculated on the basis of opening balances only, then the implicit assumption is that all other loan movements take place at the end of the time period.

Named range
Cell ranges, functions, constants and tables can all be assigned names by the modeller. Once assigned, these names can then be referred for use in calculations, which can make these calculations easier to understand.

For example, an input cell for a given interest rate could be assigned the name 'interest_rate' and could later be used in a calculation such as '=interest_rate*A1' to fetch the input value by name, instead of by cell reference.

Pivot table
Excel tools that may be used to quickly analyse large amounts of data and summarise the results by dimensions held within the dataset. These are rarely used in transaction modelling, but are common in a financial planning and analysis or financial reporting context to enable users to drill down into large date sets. For example, you could use a pivot table to calculate the total sales per region, per quarter, given a suitable data set.

Precedents
The precedents of a given cell are those cells whose value affects the value in the given cell. Ie, all cells to which the formula in a given cell refers.

Sometimes, where OFFSET or INDIRECT functions are used, the precedents can be opaque. For this reason we advise caution when using such functions.

Sliders
Sliders are controls which can be used to update the values of given input cells. These take the appearance of a scroll bar, allowing users to click the up arrow to increase the value, down arrow to decrease the value, or to slide the scroll bar to make larger changes.

Spinners
In a similar manner to sliders, spinners are another form of control which can be used to change an assigned input cell, however they simply consist of an up and a down arrow enabling small incremental movements.

Unique formulae
A 'unique formula' is a formula in Excel which is distinct in construction from the formulae above it, and to its immediate left on the spreadsheet. Looked at another way, if all the formulae in a block of cells are identical, a reviewer can gain comfort over the entire block from reviewing just that cell. It is in effect a unique identifying piece of code for that block of cells, albeit that the code is repeated in that block. The concept of unique formulae underpins most code inspection-based model review approaches.

Volatile function
A function which recalculates after any change to the model, regardless of whether the change occurs in the precedents of the volatile function.

Author

Alistair Hynd FCA, FCT

Partner
Head of Project Finance, Modelling and Valuations
RSM Corporate Finance LLP
T +44 (0)20 3201 8000
E alistair.hynd@rsmuk.com

Alistair Hynd is a partner in RSM's project finance, financial modelling and valuations service line. He is an experienced transactor, with more than 20 years' experience in corporate finance and investment banking at RSM (formerly Baker Tilly), KPMG, and Commerzbank. Before joining RSM in 2005 to establish its modelling team, Alistair was a senior transactor in the leveraged finance team at Commerzbank. Prior to that he was a founding member of KPMG Corporate Finance's PFI advisory group.

Alistair has led or delivered numerous modelling engagements, ranging from model builds for project finance transactions, merger and acquisition deals, and operational PFI projects, through to traditional model reviews, including the audit of complex multi-model rail franchise bids and securitisation models. He has acted as an expert witness on financial modelling matters for a number of complex litigation cases, including giving expert testimony in the commercial court under cross-examination on financial modelling and model audit techniques. Alistair was instrumental in developing and delivering modelling training and methodologies at KPMG, and Commerzbank, and has written a number of articles on modelling. He is a member of ICAEW's Excel Community Advisory Committee, part of its IT Faculty.

RSM is a leading independent firm of accountants and business advisers that specialises in providing an integrated range of services. It provides its growing and established business clients with audit, accountancy, personal and corporate taxation, VAT, management consultancy, corporate finance, IT advisory, restructuring and recovery and forensic services. The firm has national coverage through its network of offices and is represented internationally through its independent membership of RSM International.

RSM's modelling team provides the full spectrum of financial modelling services, from developing business models and providing assurance on third-party financial models, to delivering financial modelling training and litigation support. The team has acted on hundreds of assignments, for clients ranging from small and medium size enterprises and mid-cap businesses, through local and central government departments, to banks and blue chip multinationals.

RSM's modelling team has been a sponsor of EUSprIG – the European Spreadsheet Risks Interest Group – an academic and professional forum focused on improving spreadsheet modelling standards.

ICAEW CORPORATE FINANCE FACULTY

The faculty's professional network includes more than 80 member organisations and 7,000 members drawn from major professional services groups, specialist advisory firms, companies, banks, private equity, venture capital, law firms, brokers, consultants, policymakers and academic experts. More than 40% of the membership is from beyond ICAEW.

The faculty is ICAEW's centre of professional excellence in corporate finance. It contributes to policy development and public consultations and provides a wide range of services, events and media to its members, including its magazine *Corporate Financier.*

ICAEW is a world leading professional membership organisation that promotes, develops and supports over 145,000 chartered accountants worldwide. We provide qualifications and professional development, share our knowledge, insight and technical expertise, and protect the quality and integrity of the accountancy and finance profession.

As leaders in accountancy, finance and business our members have the knowledge, skills and commitment to maintain the highest professional standards and integrity. Together we contribute to the success of individuals, organisations, communities and economies around the world.

Because of us, people can do business with confidence.

ICAEW is a founder member of Chartered Accountants Worldwide and the Global Accounting Alliance.
www.charteredaccountantsworldwide.com
www.globalaccountingalliance.com

Chartered Accountants Worldwide

GAA Global Accounting Alliance

Corporate Finance Faculty ICAEW
Chartered Accountants' Hall Moorgate Place London EC2R 6EA UK

T +44 (0)20 7920 8685
E financingchange@icaew.com
icaew.com/financingchange

linkedin.com – find ICAEW
twitter@ICAEW
facebook.com/icaew