

Best Practices for Cash Flow Modeling

Spreadsheets are central to cash flow analysis, but poorly designed models introduce serious risk. This course addresses spreadsheet risk head-on and walks through best practices for building reliable, well-structured financial models. You'll also learn how to use assumptions effectively within spreadsheet models to support sound analysis and defensible conclusions. Any CPA or financial professional who builds or reviews cash flow projections and forecasting models will benefit from the practical guidance offered here.

CREDIT HOURS 2 | FIELD OF STUDY ACCOUNTING | EXAM QUESTIONS 10

[Start Course ↓](#)[Final Exam Copy ↓](#)

LEARNING ASSIGNMENTS AND OBJECTIVES

Assignment

- Study the full course materials included here.
- Complete the review questions after each chapter.
- Answer 10 final exam questions.
- Final exam is scored immediately, 70% or better passes.
- Certificate of completion is available in your account.
- Failed exams can be retaken free of charge.

Course Objectives

- Recognize the process and problems in creating spreadsheets.
- Recall some of the techniques recommended to reduce spreadsheet risk through good modeling practice.
- Identify what is meant by a real forecast.

Copyright Notice

© 2026 Pacific CPE, LP. All rights reserved.

This course and test have been adapted from supplemental materials and uses the materials entitled Cash Flow Analysis and Forecasting: The Definitive Guide to Understanding and Using Published Cash Flow Data by Timothy Jury © 2012 by John Wiley & Sons, Inc. Displayed by permission of the publisher, John Wiley & Sons, Inc., Hoboken, New Jersey. All rights reserved.

The materials in this course — including the text, illustrations, exhibits, examples, exam questions, and review questions — are the copyrighted property of Pacific CPE, LP. No portion of this course may be reproduced, stored in a retrieval system, distributed, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise) without the prior written permission of Pacific CPE, LP, except for the personal, non-commercial use of the registered learner.

Pacific CPE, LP grants the registered learner a single-seat, non-transferable license to access this course for the purpose of completing the associated continuing professional education credit. Account credentials may not be shared.

This course is provided "as is" for educational purposes. While Pacific CPE, LP makes a reasonable effort to ensure the content is current and accurate, the materials should not be relied upon as a substitute for professional advice. Pacific CPE, LP disclaims all liability for any error, omission, or loss arising from the use of, or reliance on, the course content.

Publication Date: August 2026

TABLE OF CONTENTS

CHAPTER

01

Spreadsheet Risk..... 6

1.01	Introduction	6
1.02	Spreadsheet Risk.....	6
1.03	Figure 19.1 The Information Flow Of A Spreadsheet Model	9
1.04	Ownership	11
1.05	Good Spreadsheet Technique.....	12
1.06	Conclusions About Spreadsheet Risk	17
	Test Your Knowledge.....	18

CHAPTER

02

Good Practice Spreadsheet Development 21

2.01	Data Entry, Formatting, And Spreadsheet Error	21
2.02	Table 20.1 Good Practice Layout Illustration	21
2.03	Types Of Cash Flow Spreadsheet Models	24
2.04	Table 20.5 Sample Of The Layout Of A Project Evaluation Model.....	28
2.05	Table 20.7 Sample Of A Part Model Illustrating The Identification Of The Net Present Value Of The Project	31
2.06	Conclusions About Good Practice Spreadsheet Development	38
	Test Your Knowledge.....	39

CHAPTER

03

The Use Of Assumptions In Spreadsheet Models. 43

3.01	Introduction	43
-------------	--------------------	----

3.02	Table 21.1 Illustration Of The Effect Of A Daisy Chain Assumption	44
3.03	The Selection Of Assumptions	44
3.04	Table 21.4 Illustration Of The Effect Of A Percentage Inflation Assumption In A Model	46
3.05	The Appropriate And Inappropriate Use Of Assumptions	48
3.06	Conclusions About Cash Flow Analysis	52
	Test Your Knowledge.....	53
Final Exam Copy		56
Index		

Chapter 1: SPREADSHEET RISK

Chapter Objective

After completing this chapter, you should be able to:

- Recognize the process and problems in creating spreadsheets.

INTRODUCTION

We could of course start developing our cash flow model by summarizing the required cash flow information on paper, perhaps using some sort of multi column pad. Indeed for some very simple tasks this may be more than adequate. However, whilst this may be satisfactory for capturing the source data, its shortcomings would soon become evident as soon as we start to try to manipulate the data.

In order to present the financial information required, in a way that is both organized and capable of easy manipulation, we need a software tool. The main tool of choice for most analysts is spreadsheet software. The first ever spreadsheet software, called Visicalc, was invented in 1979 and was initially released on the Apple platform. As this occurred before software patents were invented it was never patented. This allowed many others to develop their version of the spreadsheet, the most famous of these being Microsoft's Excel.

What made spreadsheets special, particularly for accountants and analysts, who, until then, had done everything on paper, was their ability to recalculate all values on a particular sheet after any individual cell was changed. Prior to the invention of spreadsheets making changes of this kind to forecast data could take hours with an eraser or correction fluid! The use of spreadsheets allowed for significant productivity gains. However, it also meant users became more ambitious, developing and using ever more complex spreadsheets to achieve their aims. This in turn led to the recognition of the problem of spreadsheet risk.

SPREADSHEET RISK

What then is spreadsheet risk? Put simply, this is the possibility that you are using the information in a spreadsheet to make economic decisions and the spreadsheet you are using contains errors *that you don't know about*. In extreme situations spreadsheet risk could (has) cost employees their jobs and their employers millions in losses.

The subject of spreadsheet risk has achieved some recognition in the academic world. I was first made aware of this many years ago when I was referred to a paper entitled "What We Know About

Spreadsheet Errors” published by Professor Raymond R. Panko at the University of Hawaii in 1998. This paper explained the problem of spreadsheet risk and pointed to a number of previous audits of spreadsheets in use in large businesses carried out by audit firms and others. At the time of writing this paper is still available on the Internet. These audits revealed that, without exception, in a given population of large spreadsheets in use there were a significant number of errors, with multiple errors often been present in individual spreadsheets.

Why is this error rate actually inevitable? There are a number of reasons. Firstly, unlike other commercial software, where inputs are controlled and validated by checksums and formatting limits, there are essentially *no controls* as to what a user can do in a spreadsheet. Any number (or label or formula) can be input to any cell at any time. For example the total at the bottom of a list of figures could be derived from anywhere else in the sheet. Even if the formula constituting a total is correctly coded to give the sum of the values above it may still not agree with the values displayed on the screen or printout (this being due to formatting issues which will be explained later).

Secondly, even though we have input the correct data we may make errors of logic in the structure of the model resulting in calculations that appear to result in valid output but in fact do not.

Thirdly, even if the information and logic in a spreadsheet is correct at one point in time, any cell can subsequently be overwritten. For example a cell may contain a formula, which, when the spreadsheet is presented to a user is overwritten by a value to illustrate some point. The rest of the sheet is still dynamic. However, the one cell overwritten is now fixed. Should the sheet be saved again this error becomes effectively permanent but is not visible when inspecting the output from the spreadsheet onscreen or in printed form.

Fourthly, there are risks associated with the misuse of a spreadsheet function called drag and drop. This enables the person constructing the sheet to quickly replicate whole groups of cells in adjacent parts of the spreadsheet. It is very easy to fail to drag the elements being written or overwritten to the precise location that was intended, so leaving parts of the sheet updated and others in their previous configuration.

Fifthly, we have known for many years prior to Professor Panko’s paper about keying errors. Keying errors are the errors made by keyboard operators when inputting data. Numerous studies were made of this phenomenon in the 1980s and 1990s revealing that this is a persistent problem. It is not possible to input hundreds of keystrokes of data error free. If you are a regular user of computer keyboards you will be familiar with the number of corrections necessary as a document is typed due to errors when keying. When inputting numeric values this type of error is more difficult to detect, as it doesn’t stand out as obviously wrong in the way a misspelled word does.

I hope by now you are sufficiently convinced by the above observations to accept that spreadsheet risk is a real problem. If you are not you should research the subject of spreadsheet risk further. There are many resources on the Internet to enable you to do so until you are.

MANAGING SPREADSHEET RISK

So, spreadsheet risk exists. What can we do about it?

There are a number of risk management approaches we can adopt to try to reduce, and ideally eliminate, the risk of unknown errors in our spreadsheets.

Size and Complexity

Our cash flow model can be as simple or as complicated as circumstances and the resources available to build it and manage it warrant. However, remember KISS!

KISS stands for “keep it simple, stupid!” It is a reminder that complexity *for the sake of it* is pointless. The optimum way to perform most tasks is usually the simplest.

There should always be an identifiable benefit when adding to the complexity in a model. If there is no benefit or payoff, don’t bother. Transparency and clarity should always be more important than complexity. Increasing complexity and size in a model brings hidden dangers, which are explained in more detail later in this chapter.

Planning and Specification

As a tool for the analyst or accountant the spreadsheet program is seductive. The ease with which we can start introducing information to a blank page simply by inserting a number of values and labels leads to a delusion.

The delusion being that we can construct all spreadsheets simply by trial and error. This works fine for small spreadsheets where we can see and observe most of what’s going on in a screen or so of information.

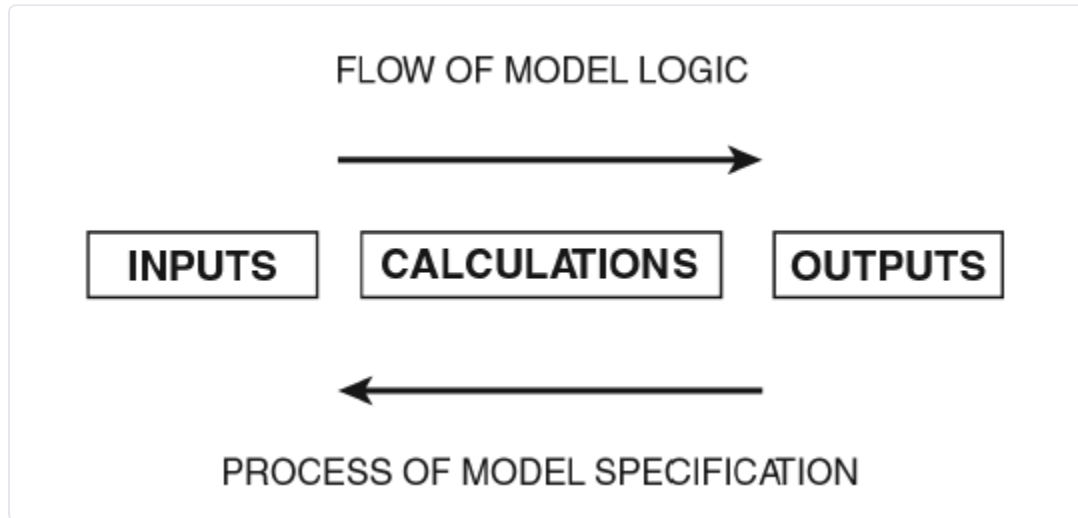
However, as soon as our spreadsheet is a couple of hundred rows down the page and 10 worksheets deep we can no longer observe the cause and effect relationships as they occur when we change a cell. Very rapidly the whole spreadsheet can become unmanageable and at some point we may lose our way and be obliged to start again, because we have lost sight of how the spreadsheet works and broken the logic somewhere. Why does this outcome occur?

The logic of a typical spreadsheet model works as shown in Figure 19.1. We have some input data. We perform some calculations on the input data. This results in output data. When we develop a spreadsheet using trial and error we follow this route in developing our spreadsheet. If we make early design errors or compromises it may turn out to be difficult or impossible to get the output we need. What we should be doing is as follows:

1. Decide *exactly* what outputs we want.

2. Identify the inputs that are required and the calculations and logic required to achieve our output.
3. Code the model accordingly.

FIGURE 19.1 THE INFORMATION FLOW OF A SPREADSHEET MODEL



For anything but the smallest spreadsheet it is wise to adopt a more formal approach to the whole process of construction. The process I recommend contains the following steps:

- A. Scope
- B. Specify
- C. Design
- D. Build
- E. Test
- F. Document and Use

Scope

- Consider the objectives of the model. What is the model for? Who will use it? How will it be used? What is its estimated working life?
- Decide what needs to be in the model and what can be omitted.
- Consider the level of detail required in the input and logical assumptions. Understand in outline how all the logic in the model will work.
- Estimate the time and resources required to develop and test the model.
- Agree the above with all the main stakeholders.

A valuable step in these initial negotiations is to mock up the proposed output schedules in a spreadsheet to show stakeholders what the output will look like. This reduces the possibility of challenge later on in the construction process as to what was required. Put in the title, labels and some sample outputs formatted as you intend to present them and make sure this is what users are expecting. Get them to sign this off if necessary. Remember, users are usually only interested in the output and its information quality. The inputs and logic required to get to the output are typically the sole province of the person responsible for the development of the model.

Specify

Specify the structure and logic sufficiently to provide an unambiguous statement as to how the results or output will be generated. (If you don't know this before you commence coding the model you will not have the skills to make the model work without outside assistance or training.) Essentially, the developer should understand the inputs and calculations necessary to complete the whole model. There should be no surprises during the actual coding process, everything should have been researched and thought out at this stage.

If there are new or unusual features in the model that the developer is not familiar with it may be wise to try them out by constructing a pilot model. This is a prototype for the part of the model where there is any uncertainty about how the logic of the model will be constructed. It is a check that the functionality of the spreadsheet works as anticipated.

Design

Having completed all the intellectual preparation required, develop a plan of the actual layout and structure of the model.

Build

Only now should you load the spreadsheet program and commence coding the model. There are a number of benefits in following this process. When you are coding you are only thinking about creating the layout structure and formula required, there is no need to constantly consider the direction and overall logic because this has already been specified.

If the trial and error approach is adopted rather than the process recommended above the spreadsheet model development is actually starting at this point. The person coding is burdened with the need to develop the model logic on the fly as well as inputting the necessary labels, input fields and formulae. This tends to result in a model that is more complex than necessary and which takes longer to build because assumptions are made and functionality is created that is not needed later in the process to generate the required output.

Test