

CHAPTER 1

Introduction

In my first book, *Modeling Structured Finance Cash Flows with Microsoft Excel: A Step-by-Step Guide*, I took readers through building a basic structured finance model from a blank worksheet. The text is a practical guide to transforming the concepts of a structured finance deal into an Excel-based model. However, in the finance industry, few people rely on a concept to close a deal. Instead, they rely on strict legal documentation that dictates the precise mechanics of the transaction. The difference between a deal based on general concepts and one based on well-defined rules can be substantial. This is why documentation exists for every concept in a deal. Attorneys spend hours writing terms sheets and indentures, banking associates review every word and integrate documents into a deal prospectus, and finally junior analysts lose sleep formatting and making charts to enhance the final prospectus.

Unfortunately, even with all this effort, reading through deal documentation can be arduous and difficult to interpret. However, well-written documentation provides a wealth of valuable information for those who want to know exactly how the deal works. Parties to the deal want to make sure every part of the transaction is well defined and published for understanding. Investors are the primary third-party readers who need to understand all the risks and rewards prior to investing in the deal. Savvy financial institutions read their competition's prospectuses to keep track of developments in structures. Auditors can use a public prospectus as a basis for evaluating a client's model. In general, anyone interested in understanding industries, asset classes, or even specific deals can gain valuable insights from deal documentation.

Reading through documentation allows for a strong understanding of the details, but the real value of documentation is that a reader can actually use the documents to reverse engineer a computer-based model of the transaction. Public prospectuses alone provide all or nearly all the information necessary to build a model that is representative of the deal. The resulting model will allow investors to see precise investment returns and the scenarios where their yields or durations are stressed. Financial institutions can model transactions to see the quantitative results of certain structures under differing stresses. They can also select a prospectus, use an internal model to reverse engineer the prospectus, hire an auditor to reverse engineer the same deal, and check both models' outputs to calibrate and audit the financial institution's model.

THE TRANSACTION

Many types of Wall Street deals can be reverse engineered. No doubt my background in structured finance influenced my decision to choose a mortgage-backed security as the example that will run through this book. However, structured finance transactions are ideal examples for reverse engineers to learn from because they are heavily documented. The deal chosen for this book is the Citigroup Mortgage Loan Trust 2006-WF2, serviced by Wells Fargo.

You do not need to be a structured finance professional to gain knowledge from this book. The focus will be on how legal documentation transforms into modeling. For those unfamiliar with structured finance transactions, specifically mortgage-backed securities, entire books are available that can help explain the concepts. For those who have an understanding of structured finance transactions, this book will reveal the inner workings of all the complex challenges presented in understanding and modeling a modern mortgage-backed security. For the benefit of the unfamiliar reader, and as a basic review for the seasoned professional, I will briefly explain the basics of this type of transaction.

Citigroup Mortgage Loan Trust 2006-WF2 is a mortgage-backed security issuance with seven senior tranches and five mezzanine tranches of debt. In general, a mortgage-backed transaction is composed of these *tranches* or “slices” of debt, which have each been funded by investors. The investors receive principal and interest that are primarily generated from assets. In this case, the assets are thousands of mortgages that have been pooled together. The interest and principal that mortgage obligors are paying are aggregated and passed through to the transaction. Depending on the tranche invested in, investors will receive certain allocations of this interest and principal. The basic transaction structure is shown in Figure 1.1.

THE DOCUMENTS

So where do you begin? First we should clarify what each document is, what it does, and what information we can procure from it. For this review, we turn to the Securities Act of 1933, the origin of securities registration. This act requires issuers to provide information about their transaction in the form of a registration statement. The exact information required by the Securities Act is detailed in Schedules A and B of the act. Other key documents that will be reviewed are shown in Figure 1.2.

Prospectus

The Securities Act requires a prospectus that discloses important facts regarding the company and the proposed transaction. The prospectus must provide full

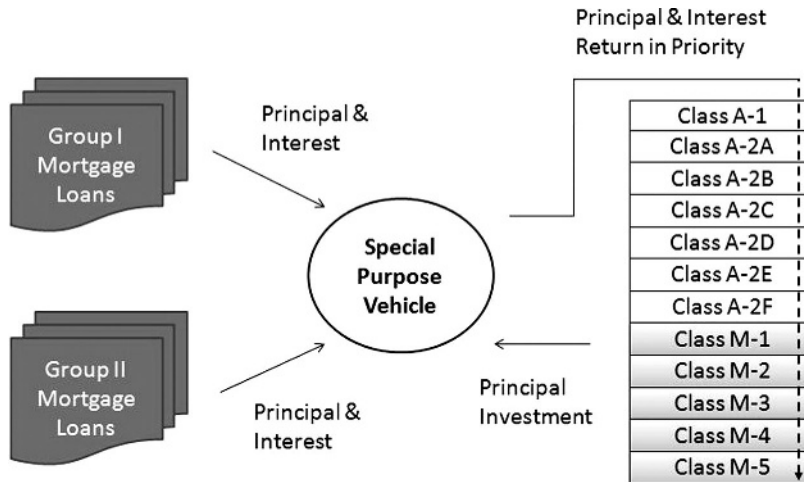


FIGURE 1.1 The flow of a standard structured finance transaction.

disclosure of all relevant facts about the securities being issued. The most important information includes:

- Information about the parties to the transaction such as the issuer, underwriter, and any entity owning greater than a 10% share.
- The amount of the issuer's securities owned by the parties to the transaction.
- The amount of debt created by the offered security, along with descriptions of the debt in terms of date, maturity, character, rate of interest, amortization style, and any terms of substitution.
- A balance sheet of the issuer.
- A profit-and-loss statement of the issuer.

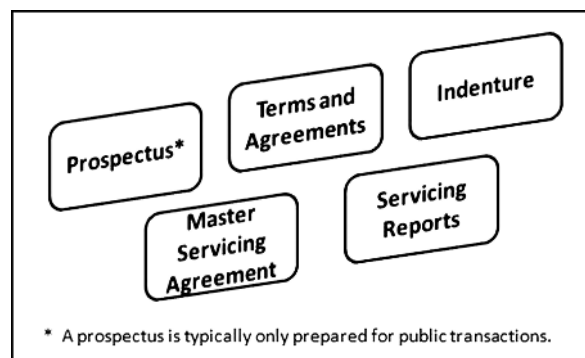


FIGURE 1.2 The core documents of a transaction.

Terms and Agreements Sheet

Prior to the creation of a prospectus, documentation begins with a terms and agreements sheet (*terms sheet* for short). Besides a few preliminary legal agreements such as confidentiality agreements, a terms sheet is one of the first major documents created for a transaction. As the name suggests, the terms sheet defines key terms for the transaction and is a precursor to a prospectus. Often a prospectus will even have a section called “Terms Sheet,” which provides important, selected information regarding the transaction.

For private transactions that are not sold into the public capital markets and therefore do not require a prospectus, a terms sheet is often the central document of the deal. Regardless of public or private intent, the terms sheet is often the document that is passed between parties and marked up as they come to agreement. A standard terms sheet for an asset-backed transaction includes:

Transaction Overview: A description of the deal, program, or facility.

Parties: The parties involved in the transaction including, but not limited to the borrower, seller, lender, servicer, trustee, collateral agent, custodian, liquidity provider, swap counterparties, rating agencies, and any other key parties to the transaction.

Program: A more detailed description of each aspect of the program that can be broken down into the following components:

- **Fees:** The percentage of variable and fixed fee amounts due to parties in the transaction.
- **Debt Description:** Important information regarding the debt size, pricing, tenor, call terms, and dates.
- **Collateral:** Eligibility criteria, concentration limits, and collateral definitions (e.g., default vs. delinquency).
- **Priority of Payments:** A description of all allocations of money during a collection period, which often changes depending on triggered events, such as an event of default.
- **Credit Enhancement:** Details of any reserve accounts or overcollateralization mechanisms/calculations.
- **Hedges:** Description of any derivatives or hedging mechanisms and parties in the transaction.
- **Events of Default:** Definitions for when the transaction is considered to be in default. These can be both qualitative and quantitative.
- **Representations and Warranties:** A detailed section that make a number of important items clear between parties, such as the financial soundness of the parties, authority to enter into the transaction, no conflicts of interest, no pending litigation, enforceability of documents, accuracy of information, government regulations, margin regulations, taxes, solvency, and many other statements that prevent ambiguities about each party and their role in the transaction.

Indenture

An *indenture* is essentially a contract between bondholders and a bond issuer. The indenture details the specifics of the bond issuance including rate, term, priority of payments, and so on. Its relation to structured finance is that an indenture exists between the trust, which is technically issuing the bonds, and the investors who are purchasing a share of the trust.

Master Servicing Agreement

A master servicing agreement (MSA) explains and defines the role of the servicer and how they will service the assets. This agreement should detail specifically how the servicer defines assets into certain states such as delinquent or defaulted. The MSA should also explain conditions for advancing missed interest or principal.

Servicing Reports

Most deals that require monthly servicing of assets are required to have some type of monthly report. This report often contains valuable data for reverse engineering, such as current asset characteristics. This is important because it allows the reverse engineer to see the current state of the assets with regard to balance, rates, periods, and so on. A reverse engineer can use this data to create expected cash flow into a transaction.

Limited performance data can also be obtained by carefully analyzing a series of servicing reports. Usually information on delinquent, defaulted, and prepaid assets is available. This information allows a reverse engineer to piece together transition rates and possibly default and prepayment curves depending how the data is reported. You should be very careful about the rates that are used in servicing reports. For instance, defaulted assets are often reported as a percentage of current balance, which is not the ideal way to capture and use default data.

Liability data is also usually included in servicing reports. The important data includes the current balance of the liabilities, the rates they are paying at, and maturity dates. This is information that a reverse engineer can use to get a current picture of the liability exposures and enhancement percentages (liability data shows the balances, so one can make a general guess on the liability priority to estimate credit enhancement).

A servicing report also shows the current status of tests and triggers. This will be seen in detail later in the text, but understanding the trigger and test calculations is important to engineering a model that correctly captures actual cash flow. Note though that a servicing report usually does not provide enough information to completely show the liability structure, namely the priority of payments.

Information Availability

One of the challenges that a reverse engineer has is obtaining information. If the reverse engineer is reversing a deal with the cooperation of the originator or a party close to the transaction, then obtaining many deal documents may be relatively easy. He or she may be able to access loan level data tapes, loss curves, or granular recovery information. However, the more distant the reverse engineer is from the source of the deal, the more difficult it will be to obtain such information. This is where the reverse engineer's ingenuity and analytical ability come into play.

THE PROCESS

Given the vast quantities of information available from documents, there needs to be a systematic approach to reverse engineering a transaction. The process can be generally encompassed by following these four steps:

1. Read
2. Conceptualize
3. Transform
4. Verify

The process, as a whole, is similar to creating a map of a hiking trail. The cartographer should first have a general idea of the area that he or she wants to map. This is akin to taking a survey of the area from above or from a vantage point that helps show a majority of the project area. Once the area is surveyed, the cartographer then begins the process of determining the relevant items on the landscape to incorporate on the map: the trail path, bodies of water, elevations, dangerous areas, roads, rails, and so on. He or she must conceptualize what objects to represent on the final map. Next the cartographer goes through the tedious process of transforming those objects to paper and digital media. Finally, the cartographer will want to test out the accuracy of the map, perhaps by trying to use it to navigate the trail.

My construction process is conceptually very similar in many respects. Let's take a look at each step in more detail.

1. Read

It may sound simple, but for first-time reverse engineers I suggest a page-through of the prospectus. A question I normally get at this point is: "Do I have to read the whole prospectus cover to cover?" I usually answer "Yes" to this question, because if a person is new enough to the process to ask the question, then they should probably read every page. As one's skill and experience increases over time, there will be sections that the reverse engineer can skip or pass through quickly. However, a new

reverse engineer might skip over important sections such as footnotes that directly contribute to a section of the complete model.

Eventually, as you get more familiar with documentation, you'll tend to jump to specific sections of the prospectus to be able to pull the necessary data. Although this may be appropriate in many situations, a quick scan of the entire document is preferable, as new deals are constantly evolving with subtle differences that have profound impacts on transactions.

2. Conceptualize

The conceptualization part of reverse engineering seems ambiguous, but one of the goals of this book is to develop a system for breaking down the information in the documentation so it can be applied to an Excel-based model. Conceptualization is the process of grouping related data points so they can be applied to all necessary sections of the model. The primary groups of data that are necessary to reverse engineer a deal include:

- Dates and Timing
- Asset Performance
- Asset Amortization
- Liabilities
- Structural Components
- Metrics

As you progress through the documentation you should take notes of each data point that can be grouped into one of these cohorts. This prepares the reverse engineer for the next step, transformation.

3. Transform

With all the data grouped accordingly, the next step is to transform the data into a working Excel model. This is done by incorporating each data point in its relevant group into a fully dynamic model. Each group on its own is of little value, but combined, they produce a working model where the deal is clearly visible and assumptions can be changed to see the impact on the transaction. Every conceptual section needs to be transformed.

Dates and Timing Dates and timing need to be incorporated as scalar and vector inputs. These will create the framework for the model, upon which all other sections are predicated. The date and timing inputs must be set up in a dynamic framework because of the unique timing challenges that reverse engineering presents. A deal can be reversed at any time after closure and can either be completely paid off, where the dates and timing are all historical or the deal can be in progress and the timing will be a combination of historical and projected. Also, the reverse engineer has the

option to model the deal from closure as a new transaction, where all the dates and timing are projected.

For the most part, the dates and timing used will be tied to the deal documentation, but if the reverse engineer envisions running any scenarios that require altering the dates or timing, then the computer setup must be done so that the user can quickly alter any of the inputs and have the resulting change flow through the model.

Asset Performance The next concept, Asset Performance, is incredibly important because the assumptions related to asset performance have the most profound impact on transaction performance. The two main performance factors for asset-backed securities are loss and prepayment. How loss and prepayments are modeled in a transaction can vary widely between deals, and also between the deal and how it is presented in the final documentation. Between deals, the loss calculations may use different bases such as defaults based off original balance or defaults based off current balance. Prepayments might be expressed as conditional prepayment rate (CPR), absolute prepayment speed (ABS), or a number of other prepayment methodologies.

Standardized rates for certain asset classes may also be used. For example, in mortgage modeling, the final prospectus might show the results of using various standard default assumption (SDA) curves in combination with various Public Securities Association prepayment (PSA) curves. Whereas these curves might be shown in the public prospectus, the originator or banker may have performed a more granular analysis on the historical loan level data.

With such variability of information, the best practice would be to first reverse the deal with available information that can be checked against outputs in the model. I discuss this process in the “Verify” section of this chapter. After the model is verified, the reverse engineer can try other performance assumptions that might not be explicitly stated in the available documentation, such as using a custom loss curve that might be more representative of the assets instead of an SDA assumption. Because this situation is similar to Dates and Timing, where there may be a number of possible performance inputs, flexibility should be kept in mind while building the model.

Asset Amortization Asset amortization itself is a straightforward process, however, it can get complicated when having to reverse engineer a deal. The complications arise because a number of methods can be employed for amortization. For term deals, where the assets that were sold into the deal are the only assets that will ever exist in the transaction, loan level amortization is often done by the original structurer. However, loan level information is never incorporated directly into the prospectus and it may not be readily available. In the prospectus, though, there is often data on loan groups, known as representative lines (*rep lines* for short) that “represent” all the assets in the deal. To initially verify a deal the rep lines can be used for reverse engineering.

Many private deals done in conduits or other financing vehicles are set to revolve where, as assets pay off or default, new ones can be added. This presents a challenge for amortization that is overcome by instituting rigorous eligibility criteria. As mentioned earlier, eligibility criteria are typically detailed in the deal documentation. Original deal structurers usually create rep lines for the assets based on the most adverse limits of the eligibility criteria. For instance, imagine an agricultural equipment deal where the eligibility criteria allowed for concentrations of no more than 40% cranes, 40% harvesters, 5% pivots, 10% dusters, and 5% tractors. Now from a static loss analysis the structurer sees that pivots have the worst loss rates. Even if the originator only plans on selling 1% of total assets as pivots into the pool, the structurer must assume that 5% of the assets are pivots. There is no guarantee except for the eligibility criteria on the concentration of assets.

This is important because when it is time to amortize the assets, the structurer should create a separate rep line for pivots and use the default rate for pivots to amortize the assets out. Likewise, unless the amortization assumptions are explicitly stated, the reverse engineer will want to amortize the assets using rep lines that represent the most adverse possible pool.

Amortizing loans or rep lines can be a challenge for a reverse engineer because there can be many, many amortization schedules to create and aggregate. Solutions do exist in Excel, but a powerful option is to use Visual Basic Applications (VBA) code to transform the amortization concept into the computer model. This book covers the code necessary to amortize a loan in VBA.

One additional complexity in reverse engineering a deal occurs if the assets pay interest on a floating-rate basis. This requires the reverse engineer to find the proper rates at the time of the deal. Also, depending on when the analysis is being completed vis-à-vis the inception of the deal, a combination of actual and expected rates might be necessary.

Liabilities The bonds, notes, or certificates that fund the deal are the easiest to reverse engineer because the Securities Act requires their characteristics and terms to be precisely laid out in the documentation. The two main types of data necessary to reverse engineer the liabilities can be grouped into loan characteristic data and deal level liability data. Loan characteristic data includes items such as original balance, current balance, fixed rate, index, margin, original term, remaining term, or any other data necessary to correctly amortize the loan. Deal level liability data is information pertaining to the ordering and style of liability payment.

Note that the style of liability payment classified as deal level liability data can be confusing to differentiate from the loan amortization created from loan characteristic data. The difference is caused by the potential of the transaction structure to alter the standard amortization of the liability. The loan characteristic data instructs the reverse engineer on how to amortize the loan as a separate entity, however, the deal level liability data refines the amortization by giving instructions on how the liability is paid when it's incorporated into the deal.

Structural Components I touched on structural components when I described deal level liability data, but given the number of possible structural features, they can be classified as a group of their own. Structural components include features such as triggers, swaps, reserve accounts, wraps, credit guarantees, and any other mechanisms that direct the flow of payments or attempt to mitigate the cash flow divergences or shortfalls caused by prepayment, default, and other risks, such as interest rate mismatches or foreign exchange rates.

Reverse engineering structural components can vary in difficulty. Reserve accounts, wraps, and credit guarantees are relatively easy to incorporate into a model. They are all mechanisms to cover shortfalls of cash. Each has inputs to determine how much loss the mechanism will cover, if and how the mechanism gets reimbursed, how much the mechanism costs, and a detailed description of which liabilities can use the mechanism and the ordering for all the fees and reimbursements resulting from the mechanism.

Triggers are tests in a deal that, if failed, can cause the liability structure to change. Quantitative triggers are easy to model because they are just conditional statements based on a test involving numeric comparisons. However, qualitative triggers exist in transactions that are sometimes accounted for during the original sizing, making it difficult for the reverse engineer to model. For example, in a private transaction, a structurer might run a scenario where they test for an event of default based on the bankruptcy of the originator a year from the start of the transaction. The structurer notices that the current structure would sustain a slight loss in such a case and suggests advancing the originator a smaller amount to mitigate against such a scenario. Without the knowledge of such steps, a reverse engineer would not know why the advance rate was set slightly lower than quantitatively necessary.

A more difficult component to reverse engineer can be derivatives in a transaction, such as a swap. Although swap rates can be assumed as the liability rates, a more precise model would want to use the swap rates in combination with market rates and the notional schedule to determine how much cash was flowing in and out of the transaction because of the derivative. For a reverse engineer this can require some research to transform a swap into a working model. The reverse engineer will have to find historic market rates and try to see how the swaps paid or took in money based on the prevailing rates.

Finally, there are a number of structural components that structurers create on their own. These are structures such as yield supplement over collateralization accounts (YSOA), net weighted average coupon (WAC) carryover, cross collateralization, and so on. Each of these changes the cash flow in very specific ways to parse and mitigate loss. For instance, YSOA accounts are typically used in auto transactions when there are auto loans in the pool that are under the deal funding rate. The extreme example is when an auto lender offers zero percent financing. The obligor is not paying any interest, but is making up for the difference in their principal amount. However, this would be a problem for a structured transaction that is a true pass-through security because there are fees and interest to be paid to debt holders. In a pass-through, the entire principal would be passed through to the debt holders and there would be nothing for transaction fees and interest. Structurers

have created methods for discounting the principal of the assets so that they create an implied interest amount to be used in the waterfall. Most structurers that create YSOA accounts include a schedule of the YSOA balance, however, newer structurers have been using dynamic accounts, which are difficult to reverse engineer.

The same is true of other substructures that are very complex in nature, but only vaguely described in the deal documentation. This forces the reverse engineer to look for auxiliary data that can allow him or her to back into the correct calculations. These substructures are usually transformed last as they can be the most difficult to verify the model with when incorporated.

Metrics Beyond analytical purposes, metrics such as yield, duration, and weighted average life allow reverse engineers to verify that their model is correct. Before we discuss verification as the final step, the transformation of metrics from a concept is very important. First, each metric needs to be calculated correctly with the right formula. This is the easiest part of transforming it from a concept to a workable system because the mathematical formulas translate easily into Excel formulas, often with preexisting functions.

Setting up a system to calculate the metrics for scenario combinations is the second step. This is an important part of the transformation, particularly for verification purposes, because a model may produce the correct results for one set of assumptions, but when the assumptions are switched there is a chance that a structural concept was transformed improperly. These errors can be detected by running combinations of different scenarios and checking the results. Often, prospectuses provide output tables for a given set of assumptions, such as prepayment and default.

4. Verify

Verifying that the model is correct is the final step of reverse engineering a transaction. This is largely an extension of the “Metrics” section where the reverse engineer verifies that the model produces results that can be matched up against the documentation. To do this, the reverse engineer needs documentation with results. Two excellent sets of documents and the results they contain include:

Prospectus: Typically contains yield and duration tables for different prepayment and default scenarios. Also includes decrement (often referred to as *dec*) tables, which provide the periodic balances as a percentage of original balance for the assets and all liabilities. Dec tables are usually created for various prepayment and default scenarios.

Servicer Report: Historic servicer reports include information regarding balances, prepayments, defaults, interest rates, and so on. A reverse engineer can piece together historic servicer reports and check if their model produces the same balances.

When should a reverse engineer use either set of documents? This goes back to the question of availability and purpose. With prospectuses, the precision of verification

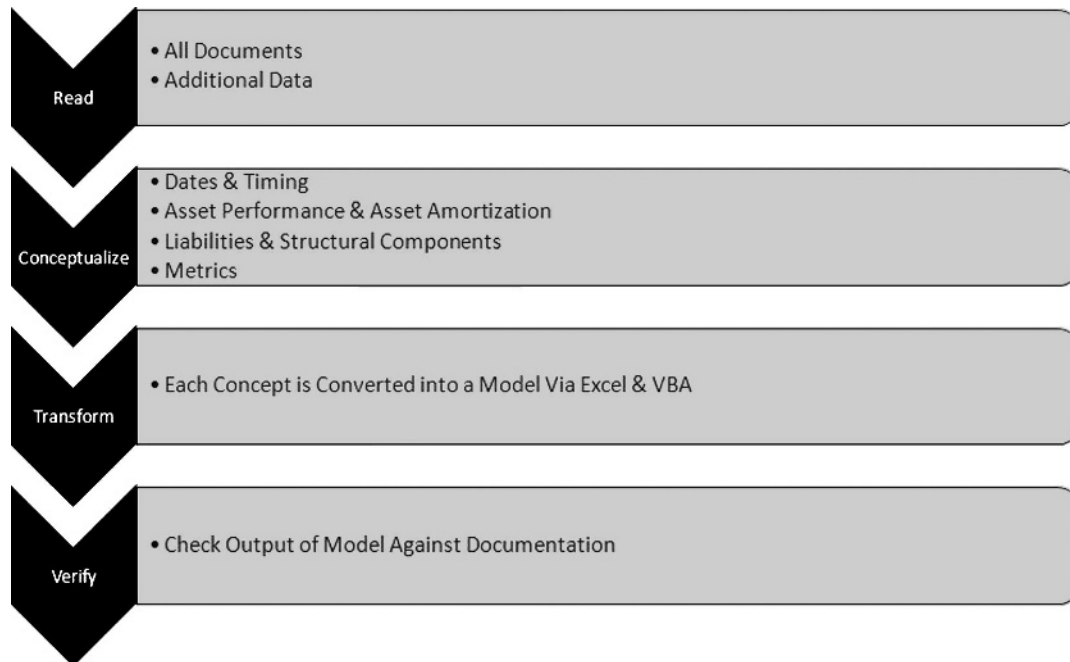


FIGURE 1.3 The general reverse engineering process should follow these four steps.

or “tying the model” to the documentation should be very good. This is because the outputs are based on a set of assumptions that are decided during the structuring. The prepayment and default curves used are either standard curves such as PSA or SDA or they are curves that have a specific assumption each period. Therefore, if the prospectus is available and the purpose is to calibrate a model, then this document should be used.

Servicer reports are based on actual performance and will have prepayment and default figures that can vary significantly from month to month. These can also be used to reverse engineer the deal with the goal of producing the most current results. Basic information about the assets though can also be garnered from servicer reports. More importantly, they are the best way to try to back into a deal’s current state. The general reverse engineering process is summarized in Figure 1.3.

HOW THIS BOOK WORKS

This is a complicated book. Although I always strive toward transparency and simple explanations and demonstrations in Excel, there are topics inherent in the complexity of the deal structure that will require complex formulas and the implementation of VBA coding at an intermediate level.

The chapters take you through each concept of the transaction by discussing the concept, showing how it is captured in written documentation, and then explaining and demonstrating how it is converted to an Excel-based model. To keep the conversion process as straightforward as possible, the model will be built off one primary document: the Citigroup Mortgage Loan Trust 2006-WF2 prospectus supplement. As I mentioned previously, this is a modern mortgage-backed security deal with seven senior tranches and five mezzanine tranches of debt. As excerpts from the prospectus supplement are examined, corresponding Model Builder sections will follow, that work through the conversion in a step-by-step format. All the information in the prospectus and referenced in the book is publicly available on the SEC Edgar web site, but a PDF version of the prospectus supplement has been provided on the CD-ROM. Readers will find the PDF version much easier to use as it has page numbers, whereas the HTML version from SEC Edgar does not. Also included on the CD-ROM is the completed model that corresponds to the final work from all the Model Builder exercises.

The Model Builder sections instruct you in a step-by-step format on how to reverse engineer the transaction and construct the model. You may notice some cosmetic differences between your final model and the example model included on the CD-ROM. This is largely because formatting instructions have not been included. You may want to use the same formatting style as the example model, but you have the option to implement your own formatting as it will have no effect on the model calculation. In addition, some unnecessary summary cells (cells that total up columns) have been left in the example model, but not instructed in the text. The ones that have not been detailed are useful to you as a reader, but not critical to the model calculation.

There is also an appendix for some VBA functionality that is not critical to the model, but greatly enhances the builder's ability to create and audit a model. This appendix provides explanation of the functionality in a method similar to the Model Builder exercises.

Finally, you will notice that from the screen shots, exercises, and model, everything was created in Excel 2007. There are many major differences between Excel 1997–2003 and Excel 2007 with regard to memory allocation and some VBA libraries; however, none of these differences should matter for this book. There are, however, some minor differences such as cell naming conventions and function problems that I will highlight in a section at the end of this book.

