

THE BUSINESS ANALYSIS LIFECYCLE: CREATE BETTER MODELS, ANALYSES, AND PRESENTATIONS WITH LESS EFFORT

Thomas A. Grossman
Masagung Graduate School of Management
University of San Francisco
2130 Fulton Street
San Francisco, CA 94117-1045

ABSTRACT

The risk analyst generally has a challenging task and limited time. He must 1) articulate the essence of a business situation in the form of a quantitative model, 2) construct a spreadsheet that implements that model, 3) perform analysis that generates useful results, and 4) craft a compelling business presentation. The wise analyst (and his manager) want to create high-quality models, analytic results, and presentations that are "right the first time" with minimal expenditure of time and effort. Unfortunately, most organizations—and business schools—provide no or minimal training. We discuss the business analysis lifecycle, and specific techniques to generate better models, smarter analyses, and more persuasive presentations in less time. We discuss the difference between Input-Oriented Modeling and Output-Oriented Modeling, the importance of separating Exploratory Modeling from Engineering a Spreadsheet, and the role of Prototyping to enhance productivity and increase quality. We discuss principles for performing probabilistic analysis, and thoughts on the art of crafting compelling presentations.

1 INTRODUCTION

Crystal Ball[®] is a powerful analytical tool for probabilistic risk analysis and range thinking. The measure of success of a Crystal Ball analysis is not the technical quality of the analysis itself, but the meaningfulness and cogency of the resulting managerial insights, and the impact of those insights on actions taken by the company.

We write from the perspective of the most demanding type of analysis: a one-off, unique analysis of a new situation, with a final audience of people with strong business knowledge but little or no knowledge of simulation or spreadsheet modeling. Readers who perform frequent simulation analyses of routine situations, directed at an audience that is accustomed to consuming simulation results, may notice that they have an effective solution to many of the issues raised in this paper; we hope that this paper will motivate them to reflect on their current practices.

The purpose of this paper is to expand your thinking about Monte Carlo simulation (e.g., with Crystal Ball) to consider a larger context of business analysis. Success depends critically not only on the Crystal Ball simulation, but on the intellectual and spreadsheet engineering activities that precede simulation, and on the extraction and presentation of managerial insight after the simulation. Thus, a successful analysis requires thoughtful consideration of the entire "business analysis lifecycle."

2 THE BUSINESS ANALYSIS LIFECYCLE

When talking about simulation using Crystal Ball, there is a natural tendency to focus on the analytical content of the work: How can one use Crystal Ball to describe risks, facilitate thinking in terms of ranges rather than point estimates, and generate probabilistic outputs? These are important questions, but they are limited to a narrow domain. It is important to broaden the focus to consider the spreadsheet model to which Crystal Ball is applied, and the business conversations subsequent to that application.

We use the **business analysis lifecycle** (Figure 1) to describe the context of a simulation analysis. An analysis starts with a *Business Situation* (top left box in Figure 1). The analyst creates a *Spreadsheet Model* (top right box in Figure 1), then works with the Spreadsheet Model (e.g., by running Crystal Ball) to create *Model Insights* (bottom right box in Figure 1) such as expected values and probabilities of important events. The analyst uses the Model Insights to articulate plain-English *Managerial Insights* (bottom left box) that can easily be communicated to others in order to influence thinking and decision-making. The lifecycle loops around, as the actions taken in response to the Managerial Insights influence the Business Situation, sometimes leading to follow-on modeling and analysis.

In Figure 1, each pair of boxes is connected by an arrow representing the activities performed by the analyst: The analyst must *model* the Business Situation to create the Spreadsheet Model, *analyze* the model to generate Model Insights, *interpret* the Model Insights to obtain Managerial Insights, and *communicate* using the Managerial Insights to influence the Business Situation.

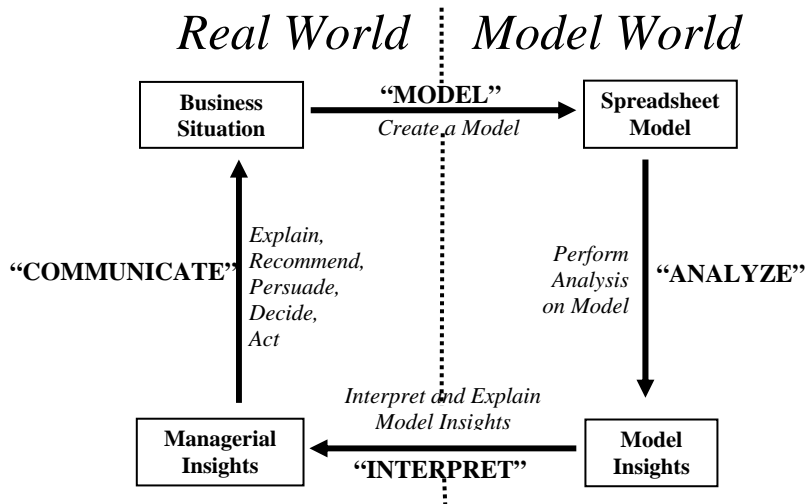


Figure 1: The Business Analysis Lifecycle

We examine in the following sections each of the four activities in a business analysis: Modeling (and the associated activity of Spreadsheet Engineering); Analyzing; Interpreting; and Communicating.

3 MODELING A BUSINESS SITUATION

The analyst’s work begins with the creation of a model that describes a Business Situation. A **model** is a mathematical representation of a business situation that use inputs (such as business decisions, and data regarding the business environment) to determine important outputs (such as sales and profit). Model outputs can be computed by hand, be programmed in a computer using a procedural programming language such as C++ or FORTRAN, and of course be programmed in a spreadsheet.

Although it is common to speak of a “spreadsheet model” as a single entity, it is wise to distinguish between the model, which is a set of ideas, and the spreadsheet that implements that model, which is a computer program. Conceptually, the model is independent of its computer implementation. Thus *modeling*—the creation of a model—is an intellectual act that is distinct from the act of programming a spreadsheet. There are many benefits to paying thoughtful attention to the act of modeling: One can develop better models, create a model in less time, and be less likely to waste resources analyzing a model that is fundamentally flawed and needs to be modified.

Modeling is generally considered an art or a craft, and we don’t know as much about it as we should. (Useful ideas about modeling can be found in Chapter 3 of Powell and Baker 2004.) Modeling seems to be an intensely personal activity. Despite these limitations, there are several guidelines that in our experience seem to be helpful.

3.1 Where to Begin Modeling?

There is no research that tells us how ordinary business people create models. From observations of business students, consulting clients, and conversations with business modelers, most people seem to start at the beginning. That is, they start modeling by entering some inputs into a spreadsheet, and work forwards towards whatever it is they think they want to compute. We call this **input-oriented modeling**. It is the equivalent of “bottom-up design” in traditional software engineering.

Input-oriented modeling is widely used. For beginners, this seems to be the “obvious” way to model. For small models, and routine models that have the same structure each time, input-oriented modeling is an efficient and effective approach. However, input-oriented modeling is analogous to building a house without blueprints. If the house is small, and similar to another house you recently built, and you genuinely know exactly what you want, you can get away with this approach. However, input-oriented modeling does not scale well to larger or harder models, and over-reliance on this technique can cause difficulties.

As the analyst works forward towards computing something of interest, he may discover late in the game that he doesn't actually know how to compute the thing that is desired. Indeed, he may discover that it is not at all clear just exactly what it is that he is trying to compute. Sometimes, the analyst reaches a "dead end" where he becomes stuck and is not sure how to proceed. The analyst can expend a lot of time entering data (and expensively gathering that data) only to discover that it is not needed after all. The results of using input-oriented modeling can include confusion, writing spreadsheet formulas that are later seen to be useless and discarded, frustration, and wasted time and effort.

We recommend the use of **output-oriented modeling**, where the analyst identifies precisely what it is that will be computed, and *works backwards* from this desired output. In the process of working backwards, the analyst creates the necessary intermediate computations, and eventually discovers the necessary data.

Working backwards from the desired output might seem strange. But upon reflection, we hope you will see that this is simply common sense; before embarking on a project, you should know exactly what you want that project to achieve. Why start programming a computer before you know exactly what it is that you want the spreadsheet program to compute? Output-oriented modeling imposes a discipline that forces the analyst to clearly articulate the goals of the spreadsheet before programming it. This focuses attention on the intermediate computations that are truly required, rather than the things the analyst "sorta-kind-a-maybe" thought would be required. As a result, the data required by the model is determined by the needs of the model, rather than by the initial hunch of the analyst, saving time on data acquisition and data entry.

3.2 Use Spreadsheets as a Tool for Exploratory Modeling

One of the great benefits of spreadsheets is that they are a wonderful environment for modeling. Have you ever considered how you would create a model if spreadsheets did not exist? It is much more difficult. In the ancient days before spreadsheets, when analysts needed models they had to write algebraic equations by hand and then ask software people to write computer code to implement those equations. Only after this lengthy process could the analyst see what the model actually did, and how useful it might be.

Spreadsheets changed all that. Because spreadsheets provide instant feedback during the modeling process, the analyst can see how the model is working, and evaluate in real time whether it makes sense, or does what the analyst wants. Thus, for most people, spreadsheets are a marvelous tool for expressing their ideas in the form of a model.

Thus, we recommend that one use spreadsheets for **exploratory modeling**, where the spreadsheet serves as a vehicle for expressing and fleshing out ideas. When using the spreadsheet as tool for modeling, it functions like a scratch pad, or the proverbial back-of-an-envelope, or an artist's sketchbook. The spreadsheet is a place to try ideas, to experiment, to explore. The spreadsheet is a vehicle for converting partially-formed ideas into fully-specified models.

The essence of exploratory modeling is attitude. While modeling, we recommend you avoid the attitude that you are creating a spreadsheet computer program. Instead, take the attitude that you are *exploring*. After you have completed a useful exploratory model, you can consider the distinct activity of engineering a spreadsheet computer program.

3.3 Separate Modeling from Spreadsheet Engineering

There is one big problem with using spreadsheets as a tool for modeling. When you finish the modeling process, you are left with two things. First, you have a deeper, clearer conceptual understanding of the problem you are working on. Second, you have a spreadsheet that seems to do something useful.

This is a very dangerous moment! That spreadsheet superficially resembles a purposeful computer program. But that is not what it is. This spreadsheet is the artifact—the residue—of your personal modeling process. It's the equivalent of an artist's sketchbook that is used to figure out the right composition for a painting. The artist would never start painting on top of a sketch; the artist will always begin with a clean canvas, on which he draws a thoughtful design to guide his painting. That design is deeply informed by the sketch or sketches; but it is always done on a clean canvas. And so it should be with spreadsheets.

After you have finished the exploratory modeling process, it is vital that you do not use the exploratory spreadsheet model as the basis for programming a spreadsheet model that will be used for analysis. Treat it like you would treat a sketch on a paper napkin; use it for reference and inspiration, but **start over from scratch**. Use a blank workbook and carefully engineer a new spreadsheet that will implement your model accurately and rapidly.

4 SPREADSHEET ENGINEERING TO IMPLEMENT A MODEL

Ideally, an analyst will quickly create an accurate spreadsheet model that can be efficiently analyzed, modified and (sometimes) transferred to other people. With spreadsheets, as with any computer software, programming in the way that seems

“natural” is a recipe for poor productivity and high error rate. Panko 2000 discusses the problems with spreadsheet errors and spreadsheet programmer overconfidence. Spreadsheet programmer productivity has never been studied, but if Panko’s assertion is correct that spreadsheet programming “seems to resemble programming practice in the 1950s and 1960s” then it is lower than it could be.

Spreadsheet engineering is the spreadsheet version of software engineering. It is concerned with techniques for creating high quality spreadsheets with high productivity. Because of the incredible diversity of spreadsheets and spreadsheet uses, it is difficult to give blanket recommendations at any level of detail.

We can provide one vital general recommendation for a spreadsheet programmer: Take seriously the task of engineering a spreadsheet! A useful framework for considering how to create a high-quality spreadsheet with high productivity is the nine elements of spreadsheet engineering summarized in Grossman 2004 and detailed in Grossman and Özlük 2004. Chapter 5 of Powell and Baker 2004 has some specific recommendations.

A common and serious mistake is to commence programming without careful consideration of the “development parameters” which identify how the spreadsheet will be used, by whom, how often, and for what purpose, and the resources and time frame in which the spreadsheet is to be created. It is wise to invest time up front so that the analyst knows what he is creating and why before he begins programming.

For many spreadsheets, it is wise to build one or more small *prototypes* before constructing the final spreadsheet. A **prototype** is a small version of the spreadsheet model, where it is easy to try out different designs, experiment with different cell formulas, and test small-scale analyses. This can save time in constructing the spreadsheet, reduce the risk of a programming error going undetected until after a significant investment in analytical work, and create a higher-quality product that is easier to use.

5 ANALYZING A SPREADSHEET MODEL

Some analysts find themselves in the comfortable situation of doing similar analyses over and over. For example, spreadsheet models of oil & gas production economics often have strong similarities, leading to similar models and analyses. In such situations, the model inputs that require Crystal Ball distributions might be known *a priori*, and the desired analytical outputs are largely known in advance. In these circumstances, the modeling, spreadsheet engineering and analysis tasks are relatively straightforward, and it is easy to get away with minimal reflection and planning.

However, there are many models where this is not the case. Upon creation of the spreadsheet model, the analyst must generate as much insight as possible in the limited time available. It is important to take a systematic approach to the analysis. In particular, even if the analyst is confident that a Crystal Ball simulation will be required, it is beneficial to defer simulation until after important preliminary work has been done.

Here are our recommendations on steps to take before running Crystal Ball. (Note: The only write-up of systematic model analysis is in chapter 6 of Powell and Baker; our recommendations are aligned with theirs.)

First, **establish a base case** by thoughtfully selecting base values for all inputs. Good base values should be “reasonable” estimates, which in turn generate “reasonable” base values for the outputs. The choice of base values is important, because they are the basis for comparison; all analysis will be comparisons to the base case. A powerful technique for misleading analysis is to use tendentious base values, so pick values that will be perceived by others as reasonable.

Second, **perform sensitivity analysis** on the inputs to identify those that are most sensitive. A “sensitive” input is one where a large change from base value results in a significant change in an important output. The sensitivity of the inputs can be rank-ordered using a tornado chart. The ranges through which the inputs are varied should reflect plausible “best case” and “worst case” values of the inputs. It is important to avoid the convenience of using a constant +/- percentage change for all inputs, because this provides insight into the “model sensitivity” rather than the “business sensitivity” of the inputs. (We note that the Crystal Ball Tornado Chart Tool is deficient in this respect.)

Third, **select the inputs to be modeled as distributions** using Crystal Ball. In analysis, as in many things, parsimony is wise. Only the sensitive inputs merit being modeled as Crystal Ball distributions. Adding distributions makes it harder to generate insight; it is easier to understand *how* uncertainty affects key outputs when there are only a few distributions. Adding distributions consumes resources: for each distribution, one must choose a distribution type and select appropriate parameters. Depending on how this choice is made, and the availability and form of data, this can consume significant resources. Adding distributions can also affect computer performance.

Fourth, **add Crystal Ball distributions methodically**. There is much to be learned from observing how a single uncertainty affects performance. It is beneficial to examine in isolation the effects of each distribution on key outputs. Multiple probability distributions can interact in a complex way. It is helpful to first understand the effect of individual distributions before introducing interaction effects. This process aids in identifying the key uncertainties to be watched or managed, and provides richer insight into the nature of the risks being faced.

6 INTERPRETING MODEL INSIGHTS

Model insights summarize analytical results in technical language. They are expressed in terms of the “Model World” of Figure 1. Model insights are numeric, precise, and use technical terminology. The generation of clear model insights is a central goal of analysis. However, for those model insights to be meaningful they need to be restated in terms of the business. Model insights need to be interpreted into managerial insights.

Managerial insights summarize analytical results in plain English. They are expressed in terms of the “Real World” of Figure 1. Managerial insights can be verbal, using business terminology rather than technical terminology in the form of short sentences or aphorisms, or they can be visual, using simple graphs to make a key point. Managerial insights are directed at people who are familiar with the business but unfamiliar with the model. You can think of managerial insights as bullet points or graphics in a managerial PowerPoint presentation.

6.1 Articulating Managerial Insight is a Distinct Skill

The crafting of accurate, coherent, compelling managerial insights is a vital skill. Unfortunately, this is a skill that often receives insufficient attention in analytical circles. This is strange, since one of the attributes that distinguishes effective senior analysts from their juniors is the ability to articulate cogent managerial insights.

Many analysts do not have the knack of explaining complex analytical results in simple, plain-English statements. They need to excel at this skill, or partner with someone who has does. We hypothesize that the failure to articulate meaningful managerial insights is frequently the root cause of failures of analytic work to influence business actions.

6.2 Work Hard at Articulating Managerial Insight

The task of writing the final report, or pulling together the end-of-project PowerPoint presentation is usually left until the end of the analysis. This seems to work well in most situations. However, the key intellectual contributions of the analysis are the managerial insights, and these should not be left until the end.

Here is our most important recommendation: Engage with managerial insights throughout the entire analytic process. Before and after every Crystal Ball run, the analyst should ask questions such as “What does this mean?”, or “How can I explain this in simple terms?” This stimulates the analyst to anticipate managerial questions, and to perform additional analysis that he might not otherwise have considered. In some cases, this additional work generates truly compelling results. Most important, the habit of focusing on managerial insight during the analysis keeps the analyst focused on the primary goal, which is communication, and lays the foundation for success in this vital activity.

7 COMMUNICATING MANAGERIAL INSIGHTS

There is a tendency among analysts—particularly those analysts who most enjoy technical work—to lump together everything that happens after the technical analysis into a vague category called “implement solution” (for example see Figure 1.4 in the best selling Ragsdale 2004), “report” or some other ill-defined term. We think this is a mistake; implementing a solution requires getting people to take action, which requires persuasive communication.

Communication, especially persuasive communication, is the ultimate goal of analytic work. Most experienced analysts and their managers can cite examples of good analytical work that went nowhere because the results could not be effectively conveyed to non-technical people. The purpose of communicating analytical results is to influence the company to do (or not do) something important. This is very different than merely reporting results, although results do need to be reported.

7.1 The Analysis Does Not Speak for Itself

Effective analytical projects are those that help stimulate and focus the myriad conversations that ultimately lead to decisions in human organizations. Avoid the attitude that “the analysis speaks for itself.” The analyst speaks for the analysis. It is essential that the analyst speak about the business, and not about the model. The core of the final PowerPoint® presentation (or other managerial communication regarding the analytical work) is a set of cogent managerial insights.

In our experience, the task of communicating persuasively is completely different than the task of doing the analysis. It requires a challenging mental transition in “thinking style,” from analytical thinking to communication thinking. My MBA students constantly struggle with this, and regularly present technical tours of their models rather than present insights into the business. In some sense, it is necessary to unplug one’s “analysis brain” and replace it with one’s “business brain.”

This helps explain why junior analysts and technically-oriented individuals often struggle or fail at the task of communicating analytic results to a general business audience. These individuals need coaching and guidance to be effective. It may be necessary for the task of communication to be done by someone more senior, or by someone who was minimally involved in the analysis.

8 CONCLUSIONS

By considering the entire Business Analysis Lifecycle and thoughtfully executing each step in the lifecycle, you can achieve better results in less time. The fundamental principle is to reflect frequently on what you are doing and why you are doing it, with the goal of efficiently achieving the desired end result.

Here is a summary of our recommendations:

- The ultimate goal of analysis is to stimulate and guide a managerial conversation that will lead to wise decisions.
- Think about what you are doing before you do it.
- Be precise about what you need your model to do before you create it.
- Distinguish between exploratory modeling and engineering a high quality spreadsheet.
- Build a spreadsheet thoughtfully, considering the nine elements of spreadsheet engineering.
- Perform analysis systematically.
- Introduce Crystal Ball distributions methodically.
- Create model insights.
- Distinguish between technical model insights and plain English managerial insights.
- Work constantly to create and refine managerial insights.
- Treat communication as a distinct and very different task.

REFERENCES

- Grossman, T. Spreadsheet Engineering of Crystal Ball Simulation Models, *Proceedings of the 2004 Crystal Ball User Conference*, Denver, 2004
- Grossman, T. and Özlük, Ö. A Paradigm for Spreadsheet Engineering Methodologies. *Proceedings of the European Spreadsheet Risks Interest Group 5th Annual Symposium*, Klagenfurt, Austria, 2004.
- Panko, R. R. What We Know About Spreadsheet Errors. *Journal of End User Computing* 10, 2, (1998) 15-21. Available at <http://panko.cba.hawaii.edu/SSR/Mypapers/whatknow.htm>.
- Powell, S.G. and Baker, K.R. *The Art of Modeling with Spreadsheets*, John Wiley & Sons, Inc., 2004.
- Ragsdale, C. T. *Spreadsheet Modeling and Decision Analysis Fourth Edition*, Thomson, 2004.

BIOGRAPHIES

Thomas A. Grossman (tagrossman@usfca.edu) is Associate Professor at the Masagung Graduate School of Management at the University of San Francisco, and has taught at the Tuck School at Dartmouth College and the Haskayne School at the University of Calgary. He did his Ph.D. in Management Science and Engineering at Stanford. He spent several years as a consultant doing analytical work, presentations, and software development for clients in the airline and electric utility industries. His research examines how people use spreadsheets, models, and business analytics to add value to organizations, including spreadsheet engineering, business analyst productivity, and spreadsheet modeling. He is President of the Spreadsheet Productivity Research Interest Group <http://sprig.section.informs.org/>, and Editor of the INFORMS Case and Teaching Materials Collection, which is soliciting Crystal Ball simulation cases. <http://www.usfca.edu/sobam/Faculty/grossman.htm>.