The Modeling Life Cycle Explained

Often, a financial modeler will be engaged by a client to deliver a financial model by a given date. Typically, these are the kinds of assignments taken by freelance financial modelers. The modeler is responsible for not just delivering the model but managing the engagement with the client and also managing the project.

This chapter introduces the financial modeling life cycle to manage the client engagement and delivery of a financial model. Following the processes discussed in this chapter will ensure successful delivery of the model.

This modeling life cycle is a complete end-to-end process as in Figure 4-1 based on the following activities:

- Feasibility
- Scope
- Requirements
• Specifications
• Documentation and timelines
• Model design
• Model build
• Model testing
• Model review
• Handover and maintenance

Keep in mind that not all these activities are required for every financial modeling project (although 2–5 are the core activities that must be undertaken in every model) based on the size, the complexity, and the time constraints of the project.
The chapter discusses the lifecycle activities and addresses how each actively impacts the overall project delivery as in Figure 4-2.

**Figure 4-1.** The financial modeling life cycle
Feasibility

The feasibility is a focused analysis of the financial modeling project which will analyze and evaluate the model project in light of whether it is technically feasible and is within a budgeted cost. The activity should produce a physical output such as document or presentation summarizing the following.

The draft modeling project scope

This is used to define the business problem and/or opportunity to be addressed. The draft scope would define the parts of the client business affected either directly or indirectly by such a project, all project participants and end users, and the project sponsor.

Should the feasibility get an approval, then the draft scope will be used in the modeling life cycle and given a greater detail. Ironically too many financial modeling projects do not have a well-defined project scope. Consequently, this leads to projects that wander in and out of business boundaries and an expanded scope after the project has commenced.

The current analysis

This current analysis is an understanding of the current implementation, such of a system or financial model. The analysis will discover there is actually something wrong with the current system or model. Often, in modeling project, this analysis can suggest all that is required is a modification or update of the existing model. The strengths and weaknesses of the current system or model must also be identified (pros and cons).
A very common problem with this analysis is when the modeler discovers the issues of the current model, there is a temptation to stop the analysis and fix the model at this time. Always keep in mind that this is a feasibility; the broader project has yet to commence, so all that is required is to document your findings at this stage.

Current requirements

There is a conflict here because one of the tasks of financial modeling is the requirements gathering by the modeler, but the feasibility is prepared before. That said, it is vitally important that a top-level requirements gathering is performed specifically for the feasibility; this will not be as detailed or a deep as that taken during the requirements gathering stage.

The approach

The approach is a recommended course of action (solution) to satisfy the top-level requirements. Here, various alternatives are considered along with an explanation as to why the preferred course of action is recommended. The considerations though are

- Does the recommended approach satisfy the requirements?
- Is it also a practical and viable solution?

Evaluation

The evaluation should always examine the cost-effectiveness of the recommended course of action. In addition to the recommended solution, other alternatives are estimated in order to offer an economic comparison.

Review

All of the preceding elements put into a feasibility study document or presentation and a review is conducted with project sponsor and relevant personnel. The outcome of the review is to substantiate the thoroughness and accuracy of the feasibility study, and to either approve or reject the project, or to revise the feasibility before any decision.

For the modeler, an important activity is to make sure that approval of the project is signed by the sponsor and a commitment to support the project is given in writing. (Signatures carry a lot of weight later on as the project progresses.) Should the feasibility be rejected, the reasons for its rejection should be explained and attached to the document.
Conclusion

There is a very possibility that the modeler will enter into a client engagement after a feasibility has been completed and approved. Such a situation could be awkward as it means another person has provided the analysis of the solutions required and the requirements without input from the modeler.

I advise any modeler in this situation to be unduly concerned, because you have the engagement and there will be an opportunity to create the project scope and requirements at the project commencement. This will be your opportunity to address any concerns or issues.

There are some modelers who feel feasibilities are a waste of time and provide no tangible benefits. Just be aware that a feasibility study represents a commonsense approach to planning. On any modeling project, it is vitally important to have had a well-thought-out approach and plan to follow; otherwise, when things go wrong, the buck stops with you.

Scope

The financial modeling project scope pertains to the work necessary to deliver a model and the deliverables. Each financial modeling problem should have a scope put together in a document that has had a review and approval by the stakeholders (model users) and sponsor.

Key concepts of a project scope

The modeler should not be alarmed if the scope planning appears to be going back and forth. The best scope documents are almost always reiterative before a final agreed document is ready. The actual document should have a project scope baseline statement (what is in, what is out of scope) and a high-level breakdown of the work activities.

The important factor of the modeler at scope activity is to make sure all stakeholders must understand the scope baseline to minimize scope creep during project execution. It could be helpful when presenting a scope document to stakeholders and sponsors to have addressed the following points:

- Project justification (use the feasibility study if there is one)
- Project objectives
- Project scope description
- Project acceptance criteria
• Project constraints
• Project assumptions

Project justification
It is critical that you justify why the project came to be by describing the client’s business needs and how the model project addresses that need. Add to that the planned scope of work the modeler will perform. This is also an opportunity to add how the modeling project could be affected by any other activities you can foresee happening in the client business.

Project scope objectives
Objectives are statements describing what the project is trying to achieve. To assist with getting this objective right for the scope, think in terms of measurability. Write an objective in a way that can be measured so you can know when the project has been completed and delivered satisfactorily.

Product scope description
The scope descriptions are the features and functions the modeler foresees in the final model.

Model acceptance criteria
The acceptance criteria are a really critical part of the scope document. This section should state the standards required to satisfy the client quality expectations of the model to gain acceptance of the final model. Typically, this will come directly from the requirements gathering activities. Notice that the scope is before the requirements are gathered. In fact, you will find that because of the reiterative nature of the scoping; the requirements gathering tends to occur at the same time. From experience in modeling projects, the times of acceptance criteria tend to be

• Target dates
• Major functions
• Capacity, accuracy, and availability
• Repair times
• Development costs
• Running costs
Project constraints

A good advice is making sure the scope document clearly states project constraints. Constraints are anything that restricts the limit of the model, when the model is completed and how the model is produced. The most common constraints on modeling problems fall broadly into

- Technological: The sequence in which individual project activities must be completed
- Resource: Lack of necessary resources which may force parallel activities to be performed in sequence
- Physical: Contractual or environmental conditions

You want to make sure all possible causes of delay in the project's completion are highlighted in the scope. This is your indemnity, so make sure you put some real thought in it.

Project assumptions

Assumptions are statements that we believe to be true and how the modeler will address uncertain information during conception, planning, and performance of the project. Assumptions are identified to add potential risk to a project even though they may turn out to be false. Assumptions can impact any part of a project life cycle, so it is important to document and analyze them.

Requirements

Requirements gathering is absolutely critical to any project. Before reading further, I want you to think of when you have performed a structured requirements gathering exercise. Did you feel confident that you were doing it right?

I ask the question because, for many of us, requirements gathering just sounds like common sense. Go out there, listen to stakeholders, and write down the list of things they want, right! There lies the problem because requirements gathering in modeling projects needs to be performed correctly as it drives the model testing. If the requirements are not well understood, the testing will not be accurate, and the final model will not be appropriately tested.

The best way forward with requirements gathering is for modeler is:

**Step 1**

Create a statement of requirement. This can be a document template with the following:
• A succinct requirement specification for management purposes.
• A statement of key objectives.
• A description of the environment in which the model will work.
• Background information and references to other relevant materials.
• Information on major design constraints.
• The contents of the statement of requirements should be stable with very infrequent changes.

The statement is created by the modeler, but the details are filled in with the combination of stakeholders, project sponsor, the model end user, and the modeler.

**Step 2**

Make sure you have cross-referenced the requirements in the statement of requirements with those in the high-level requirements in the feasibility and the scope to ensure there is no mismatch.

**Step 3**

The last step is of greater importance. Once the statement of requirements is complete, the modeler must make sure all parties including stakeholders, project sponsor, and end user sign up to it. The modeler must then ensure all parties understand that this statement, and only this, is being delivered. There is a check that the modeler should make prior to getting the final sign-off.

**Pointers to modelers for a good requirements gathering**

Gathering requirements is not easy but is so critical to model projects. You get one chance to get it right, so failure is not really an option. Here are some pointers to any modeler on what to look for:

• Never assume you know what the stakeholder wants: There is nothing wrong with asking; always ask.
• Involve the model users from the start: The model users may not be clear at the outset of a project; do some investigation and locate them.
• You have a copy of the agreed scope of the project.
• Make all the requirements specific and measurable: This will help you to clarify when each requirement has been achieved in the model.
• Be realistic about requirements: Don’t allow stakeholders to push you past something that is just not possible.
• Make each requirement have a timeline: You can have two requirements statements, your master copy and the official approved. In your master copy, place a timescale against every requirement (start date, end date); you can use this in your overall project plan.
• If there is any doubt about anything in the statement of requirements, go back to the stakeholders and sponsors to get clarity.
• Always playback your understanding of the requirements to the stakeholders and sponsor.
• There will be a temptation from stakeholders to discuss the model solution and the technology during the requirements gathering. Avoid any such conversations until the requirements are fully understood; otherwise, you could yourself having stakeholders with implied requirements.
• Do not start the project until the statement of requirements is understood and agreed by the stakeholders, sponsors, and model users. You may be tempted, but should you start too early, your requirements will continue to change and move throughout the project.

Here are some pointers of mistakes that have arisen from financial model requirements in practice that you should always avoid:

• Basing a modeling solution on a cutting-edge technology unrelated to modeling only to discover it is just not possible to implement
• Not giving the priority requirements, such as “Mandatory,” “Optional,” and “Nice to have”
• Have not spent enough time consulting with the eventual model users
• Communicating you have the solution before knowing the requirements
• Lacking a clear understanding and making assumptions rather than clarifying
Conclusion of requirements

A requirements gathering is about creating a clear, concise, and agreed set of customer requirements that allow you to provide precisely what the customer wants.

Specifications

The specifications are the least understood stage of any modeling project because it tends to be more about the modeler’s experience. So, what is it?

A specification is the discussion of a specific point or issue. The specification in a nutshell is the part where the modeler discusses how they will achieve each aspect of the requirements.

When creating the requirements, the specification is developed alongside each requirement. The modeler will be looking at each requirement creating a template on how they will complete that requirement.

In the modeling life cycle, the specification and requirements are reiterative, and thus as each requirement is developed, then a specification is also produced. The longer the requirements take to complete, then so does the specification. This will be the modeler working document on achieving the technical challenges of the modeling project.

The specification document can end up being very detailed, defining each requirement implementation. For example, a specification document may list out all of the possible error states for a certain worksheet in the model, along with all of the error messages that should be displayed to the user. The specifications may describe the steps of any functional interaction, and the order in which they should be followed by the user. A requirements document, on the other hand, would state that the model must handle errors reasonably and effectively and provide explicit feedback to the users.

A specification may not always be a document. Some modelers prefer to create diagrams or schematics of functional relationships or flow logic. In some cases, its possible specifications can also be in the form of a prototype of the model.

Unlike the scope and requirements, the specification does not require approval or sign-off from the project client; it is purely a guide based on the modeler’s thinking for them to follow. Project specifications are much more important for determining the quality of the final model.
Documentation and timelines

Through the years, I have audited, reviewed, and updated several hundred financial models. The vast majority of those models have had no documentation, and of those that did have, the quality of the documentation has been poor.

Documenting the financial modeling should never be just an option; it’s a critical activity for so many reasons. The foremost reason is that it brings the financial model to life.

Documentation provides a means by which all aspects of the model development and its subsequent maintenance are available to whoever owns the model after the modeler has built it.

Often, the documentation is confused with a user manual. Well the user manual is a subset of the overall model documentation.

So, what is the documentation?

The documentation is used to define the way the project is managed by any governance that surrounds the project and consists of several parts that make it up:

- The documentation has a project plan, and yes, the project plan is part of the documentation.
- The documentation contains statement logs that demonstrate the status of the project when active from various views such as risk.
- The documentation is a guide of the technical functionality and also a user guide on using the model.

Financial modeling project plan

The project plan is the most important document that a modeler will create and maintain once the project begins. The plan is a forward-looking timeline that has a list of all activities during the project plan with the criticality and also dependencies on tasks and activities. It is used by the modeler to understand when the project is on track and when there have been slippages and to communicate to the project sponsor ahead of time.

Goal log

I find it amazing how many financial modeling projects have no defined goals. Perhaps, this is because there is an assumption that the goal is in the scope, for example, a goal “deliver an overhead allocation model.”
The statement may well be the deliverable of the project, but it conveys no information to the modeler. The goal needs more definition, such as “allow finance teams to price services” or even better “monitor 1,000 transactions through the portal per day.” See how such a defined goal portrays a stronger message. The reason why goals need to be stated and clearly defined is that they help to drive the design of the model and its outputs. They allow the modeler to understand the problems to be solved. They can then shape the specification the modeler creates. Goals are a key to the documentation.

Risk log

It is important that everyone is aware of the risks inherent in a project. Something simple like “the client does not have the resources to test the model” should be included. The risk log should be created as a single document with the risk, the agreed actions, and the impact if a problem occurs. Everyone should be aware of this document, and it should be maintained constantly. One line per risk with all the details is the best way to maintain the log.

Issue log

So, what is the difference between an issue and a risk? This is something every modeler should learn and keep in mind every time a model is being developed. The key difference is that a risk is always uncertain and could might or might not happen. An issue is something that has already happened and needs to be solved (if not, there may be a risk of something happening). An issue log should be maintained by the modeler alongside the risk log.

Model design

The design for the modeling project is a core element in the financial modeling life cycle. This design phase is about planning and carefully scripting how the model will look, feel, and work:

- To design a model end to end, all aspects of the model are covered in this stage including
- Decision on the type of model suitable for client scope and requirements (model types)
- The modeling environment (the support available)
- The modeling methodology that will be used (using own standards of one of the prevailing standards like FAST, OPERIS, BMP, or FMP)
Chapter 4 | The Modeling Life Cycle Explained

- The best practice level that will be implemented
- The financial standards to adhere (i.e., International Financial Reporting Standards (IFRS), US GAAP, Sarbanes-Oxley)
- The modeling styles to be implemented
- The in-model documentation
- The allocation of time and resource to the model build

Depending on the environment presented by the client and the model type chosen, the modeler might need to spend some of the time in design preparing external linking schedules such as

- An understanding of the risks and dynamics of the business
- Adequately reserve for guarantees or options in products and develop hedge strategies
- Key drivers behind the incentive of influencing regulatory capital levels
- Different reporting and profit measures such as risk-based capital requirements

The design of a financial model is discussed in more detail in Chapter 5.

Model build

The build phase is the most technical aspect of the entire financial modeling project. This stage is the actual physical creation of the model generally using Microsoft Excel application. The build is dominated by model scripting and model technical build practices and is the realization of the physical financial model.

There is a strong reliance on the skill set and experience of the modeler as to the quality of the model build. The aim for every financial modeler in the build stage is to look for simplicity with all the scripting.

Models designed by an experienced modeler will always show a high level of adherence to best practices and intuitive methods. Ironically, modelers with limited experiences tend to use the build stage as a way to showcase complex and badly thought-out scripting which reduces the case of a model review and audit. In Chapter 7, we take a closer look at how the model building would typically look like.
Model testing

Financial model testing is unfortunately rarely performed outside of the investment banking. This is perhaps one of the most overlooked phases of the modeling life cycle, and yet if performed properly, testing gives a model the strongest indicator of the model quality and that it performs as originally specified.

There are a number of types of testing that should be performed on the financial model, and not all of them are required for every model. The modeler should be able to advise any tester of the model as to which test would be suitable and which are not.

Model stress testing

All financial models bring an uncertain scenario, because there is no way to accurately produce the future with 100% accuracy all the time. Financial models are one method of getting the prediction, but because they rely on assumptions made by people, they are not infallible.

The power of models is that they are dynamic because change is reflected in assumptions which allows for a simplified viewing of reality. For financial modeling the simplification of reality means it’s impossible for the modeler to account for every possible event in the model. For example, the activation and spread of the COVID-19 virus in 2019–2020 and subsequent events surrounding that spread are unlikely to have been predicted accurately in a financial model prior to 2019. This is not a failure by financial modelers, but just goes to show that there are just too many variables to capture in any model.

With this understanding of simplicity of modeling to real life, we should now understand that financial models will have a risk factor built into them.

To address this shortcoming, risk managers developed stress testing as a tool to evaluate the potential impact on portfolio values of unlikely, although plausible, events or movements in a set of financial variables.

The techniques of stress testing are based on testing either sensitivities or scenarios within the financial model. Most financial models can benefit from the sensitivity tests because they assess the impact of large movements in variables on the outputs without specifying the reasons for such movements.

Sensitivity testing

An example of sensitive impact could be a 1% increase in annual inflation which brings a 2% decrease in annual EBITDA (earnings before interest, tax, depreciation, and amortization). If a financial model is built to best practices, tests like these can be run relatively quickly.
This sensitivity testing will highlight the inputs which have the greatest effect on results and identify them as key drivers. Sensitivity testing can also test if the model is realistic, because users will generally have some idea of what to expect when they make a change. Sensitivity tests are useful for exposing the accuracy of a model if accurate enough. If we know there is a linear relationship between input and variable, then we can gauge the expected change in the outputs. For instance, if we increase the cost of goods sold by 10%, our gross profit should decrease by the same level (10%).

Sensitivity testing has its limitations. You have to be careful how much you change each assumption, and how you compare the sensitivity of one assumption against that of another. Because it focuses on changing one element at once, to see the effect, you can’t get a feeling for worst or best cases, nor for the probability of good or bad results. So, it should be used together with other tests, and not on its own.

**Scenario testing**

Scenario testing is somewhat more challenging because it relies on the financial modeler building a model with option trees. Scenario testing looks at the impact on a set of alternative scenarios such best case, worse case, base case, or (pessimistic, expected, and optimistic), once they are run through a model. Each scenario will consist of a set of assumptions and inputs chosen by the user.

This approach is very useful for business planning and risk management, because it shows the impact of different conditions.

Scenario testing is quick, and it is easy to present the results to management, to show them best- and worst-case extremes. On the other hand, it can’t tell you how likely it is that each scenario will occur or whether there are other important scenarios that need to be tested. We would need to combine the scenario with a probability assessment such as Monte Carlo to get a better picture of the likelihood that the scenario will play out. In Chapter 13, there is a more detailed discussion of stress testing and other types of testing.

**Model review**

There is a likelihood you have had both the terms model audit and model reviews in reference to financial models and wondered what the difference is. To set you right, there is no difference; they mean the same thing. The difference comes down to liability implications and how the auditor qualifies a financial model. We will use the term model review going forward in the book to mean both audit and review.

The model review process is painstaking and very exhaustive by nature. At some level every cell in a financial model must be examined, and the
relationships to other cells and the impact of the final model must be exposed and documented. The time it takes to complete a full model review on financial model can be measured in hours, days, and weeks depending on the complexity and size of the model. Complex and large models will almost always take weeks to fully be reviewed.

The impact will be felt in the final bill. Model reviews are expensive firms that perform the review bill on the hours taken to review the model. The takeaway from this charging is if you want to reduce the model review, simplify the model, make it as light as possible, and build it to best practices.

In Chapter 14, model reviews are covered in detail; however, we can summarize the review process as being based on a number of checks:

- **Inputs checks**
  - Data
  - Assumptions
  - User inputs

- **Calculation checks**
  - Running tests
  - Reasonableness testing
  - Low-level review
  - High-level review
    - Sensitivities

- **Final report and review**

**Handover and maintenance**

The model handover is the phase where the modeler passes the responsibility of the model to the sponsor and ultimately the final user. This involves the activities that are performed prior to handover and after the handover pre- and posthandover, respectively.

**Prehandover preparations**

During the predelivery stage (running in parallel with model development), the modeler should consider how to prepare the maintainer to accept the model:
This may require holding some workshops for the sponsor and users to get acquainted with the model. Typically, the workshop will be including the following:

- A walk-through of the functional specification for the model
- A review and finalization of the user manual by the modeler with the maintainer
- A review of the technical manual by the modeler with the maintainer
- Providing the maintainer with the testing document and detailed brief on results and potential issues
- A review of the version control and version history detail with user details
- A review of the details of username, password, lockouts, and access controls
- Deployment diagram if applicable
- Model configuration and build
- Details of any support tools, databases, and applications that are required
- A thorough walk-through of the full capability of the model, including examining the structure and any VBA coding

**The model handover**

In the handover stage, the responsibilities of the model and accompanying materials are transferred from the modeler to the maintainer performing maintenance and support. Typically, these actions will be taken:

- Transfer of the model through an agreed method
- Transfer of data, assumptions, manuals, and other documents and materials reviewed in prehandover preparations and also that are required to operate and maintain the model
- Transfer of all previous builds and versions of the model. An acceptance from the maintainer confirming the items handed over, which should include versions and dates (It is sometimes preferable to also have an assurance from the maintainer that the items in prehandover have been covered.)
Posthandover and maintenance

The posthandover phase involves the modeler providing support and maintenance to the model and support for the maintainer for any activities that could not be provided for during stage 1. This phase would also involve a general maintenance of the model for a limited time only. This is potentially the most contentious aspect of the handover because it involves setting time limits on the length of time of the support. In addition, any possible handover activities that are required to be performed on the model should be clearly communicated and an assessment given of their impact on the model. Another aspect that would cause potential issues is the financial remuneration connected with the support. Does the modeler charge for this maintenance or is this part of the modeling brief and project? This situation should be based on a firm agreement between the receiving entity and the provider.

Conclusion

Now that you have awareness of financial modeling life cycle, it should bring some structure into the process of delivering a financial model for a client, or for internal stakeholders. As I mentioned earlier, not all the phases will be required in every model, but I do stress that every modeler should be aware and comfortable in managing every aspect of the life cycle irrespective of whether they use each phase.

Financial modeling is still seen as dark art pockets of industry; there is a mistrust as to the effectiveness of models and misunderstanding of what financial modelers practice on their trade. Using the life cycle as a guide will reduce so much of that mistrust, by bringing professionalism, ethics, and consistency to the financial modeling space.