Simulation Modeling to Predict Drug Pipeline Throughput in Early Pharmaceutical R&D

By

Jeffrey B. Heyman

B.S.E. Biomedical Engineering and Electrical Engineering, Duke University, 2002

Submitted to the MIT Sloan School of Management and the Engineering Systems Division in Partial Fulfillment of the Requirements for the Degrees of

Master of Business Administration
AND
Master of Science in Engineering Systems

In conjunction with the Leaders for Global Operations Program at the

Massachusetts Institute of Technology
June 2010

Signature of Author _____
May 7, 2010
Department of Engineering Systems, MIT Sloan School of Management

Certified by _____
Roy Welsch, Thesis Supervisor
Professor of Statistics and Management Science and Engineering Systems

Certified by _____
Deborah Nightingale, Thesis Supervisor
Professor of the Practice of Aeronautics and Astronautics and Engineering Systems

Accepted by _____
Nancy Leveson
Chair, Engineering Systems Division Education Committee
Professor of Aeronautics and Astronautics, Engineering Systems at MIT

Accepted by _____
Debbie Berechman
Executive Director of MBA Program, MIT Sloan School of Management

This page intentionally left blank.

Simulation Modeling to Predict Drug Pipeline Throughput in Early Pharmaceutical R&D

By

Jeffrey B. Heyman

Submitted to the MIT Sloan School of Management and the Engineering Systems Division on May 7, 2010 in partial fulfillment of the requirements for the degrees of

Master of Business Administration
and
Master of Science in Engineering Systems

Abstract

With high costs and growing concern about research and development (R&D) productivity, the pharmaceutical industry is under pressure to efficiently allocate R&D funds. Nonetheless, pharmaceutical R&D involves considerable uncertainty, including high project attrition, high project-to-project variability in required time and resources, and long time for a project to progress from a biological concept to commercial drug. Despite this uncertainty, senior leaders must make decisions today about R&D portfolio size and balance, the impact of which will not be observable for many years.

This thesis investigates the effectiveness of simulation modeling to add clarity in this uncertain environment. Specifically, performing research at Novartis Institutes for Biomedical Research, we aim to design a process for developing a portfolio forecasting model, develop the model itself, and evaluate its utility in aiding R&D portfolio decision-making. The model will serve as a tool to bridge strategy and execution by anticipating whether future goals for drug pipeline throughput are likely to be achievable given the current project portfolio, or whether adjustments to the portfolio are warranted.

The modeling process has successfully delivered a pipeline model that outputs probabilistic forecasts of key portfolio metrics, including portfolio size, positive clinical readouts, and research phase transitions. The model utilizes historical data to construct probability distributions to stochastically represent key input parameters, and Monte Carlo simulation to capture the uncertainty of these parameters in pipeline forecasts. Model validation shows good accuracy for aggregate metrics, and preliminary user feedback suggests strong initial buy-in within the organization.

Thesis Supervisor: Roy Welsch
Title: Professor of Statistics and Management Science and Engineering Systems

Thesis Supervisor: Deborah Nightingale
Title: Professor of the Practice of Aeronautics and Astronautics and Engineering Systems

This page intentionally left blank.

Acknowledgments

I would like to thank the many people who played a role in creating this research project and making it a success. Through all of their support, I feel that the project provided a rewarding experience both for NIBR and for me personally.

Within NIBR, I was fortunate to be able to interact with knowledgeable researchers and managers across the organization. Everyone was extremely generous with their time, and continually offered invaluable insight towards my research efforts. I would specifically like to thank the entire Portfolio Management team for their expertise, guidance, and patience. Thank you to Tim Smith, my project supervisor, and Detlev Biniszkiewicz, project champion, for not only helping me define the project and align it to NIBR's needs, but for being personally invested in the project's ultimate success. Thanks also to Henry Wang and Charles Snow for fielding my frequent questions and supporting the project in general.

Thank you to my MIT faculty advisors, Roy Welsch and Debbie Nightingale, who have contributed to my growth both in the classroom and in this project. I greatly appreciate the expertise and enthusiasm they brought to this project.

Thank you to the LGO program staff and my LGO 2010 classmates, who have made the past two years a truly life-changing experience.

Finally, thank you to my wife, Rachel, my parents, my sister, my grandmother, and my in-laws for your love and support through this and other milestones in my life.

This page intentionally left blank.

Table of Contents

Table of Figures

1 Introduction

1.1 Thesis Motivation and Project Objective

Pharmaceutical research and development (R&D) involves considerable uncertainty, including high attrition (i.e., project termination prior to drug commercialization), high project-to-project variability in required time and resources, and long time for a project to progress from a biological concept to commercial drug. Despite this uncertainty, senior leaders must make decisions today about R&D portfolio size and balance, the impact of which will not be observable for many years.

Novartis Institutes for Biomedical Research (NIBR) is the global research organization of Novartis AG. Its Portfolio Management Group (PMG) aims to create tools that aid senior management in understanding pipeline status and progress, ultimately improving strategic portfolio decision-making capabilities. While a current-state project tracking tool is currently being developed, the PMG cites forward projection as a desired capability not currently achieved in a systematic, aggregated fashion. A significant challenge to successful forecasting is the high project uncertainty described above. Key questions include:

- Given the current pipeline, what will our pipeline look like in x-years?
- To achieve the desired pipeline in x-years, what should our pipeline look like today?
- How can various pipeline levers affect pipeline productivity?

The proposed solution is a forward-looking model of Novartis' research portfolio that will aid senior management in forecasting pipeline status, throughput, and productivity. The model incorporates information about the current research portfolio and historical project performance, and outputs probabilistic tallies of (1) the number of projects in each portfolio phase in the future, and (2) the number of key pipeline milestones achieved. This model will serve as a tool to bridge strategy and execution by anticipating whether future goals for pipeline throughput are likely to be achievable given the current project portfolio, or whether adjustments to the portfolio are warranted. Furthermore, by accurately forecasting the number of projects per research phase, this tool will aid resource planning both within NIBR and for downstream development activities. The research aims to develop the described model and evaluate its utility in aiding R&D portfolio decision-making.

1.2 Research Methodology

The challenge of R&D portfolio forecasting is encountered by all organizations in any R&D-intensive industry. As such, we treat the work at NIBR as a case study focusing not only the results of the modeling process, but also on the approach to developing an R&D pipeline model in general. The lessons learned from this study can be extended to organizations facing similar challenges.

Portfolio modeling has been used extensively in many industries. Therefore, we strive to leverage any existing knowledge in the field to guide the modeling effort, while at the same time recognizing any unique challenges within the studied organization that require novel methods and approaches. The research included the following key activities:

- **Literature review**: We strived to leverage existing knowledge and experience with R&D portfolio modeling by investigating methods employed within and outside the pharmaceutical industry.
- **Stakeholder interviews and process mapping:** To define the ultimate structure and logic of the portfolio model, we interviewed various stakeholders involved in research and portfolio management to elucidate real-world project flow and portfolio dynamics.
- **Data collection and analysis:** As discussed herein, the model relies extensively on historical project performance to forecast future performance. Therefore, we put significant effort into collecting and analyzing historical project data.
- **Model design and development:** Using input from stakeholder interviews and process mapping, we defined the logic and structure of the forecasting model. We employed an iterative, incremental approach to building the model, adding successive features along the way, and refining as necessary based on team feedback.
- **Organizational implementation of the model:** Implementation efforts included presentation to leadership, end-user training, technical training, and model-driven analysis of the current pipeline.
- **Model validation and assessment:** We validated the model to judge accuracy, and assessed its utility in supporting strategic portfolio decision-making.

1.3 Industry Context

The costs of pharmaceutical R&D are significant, and appear to be rising quickly over time, as shown in Figure 1. The rise in costs has been accompanied by a reduction in the rate of new drug approvals.(1) This disparity has raised some concerns of a productivity-crisis in the industry as a whole.

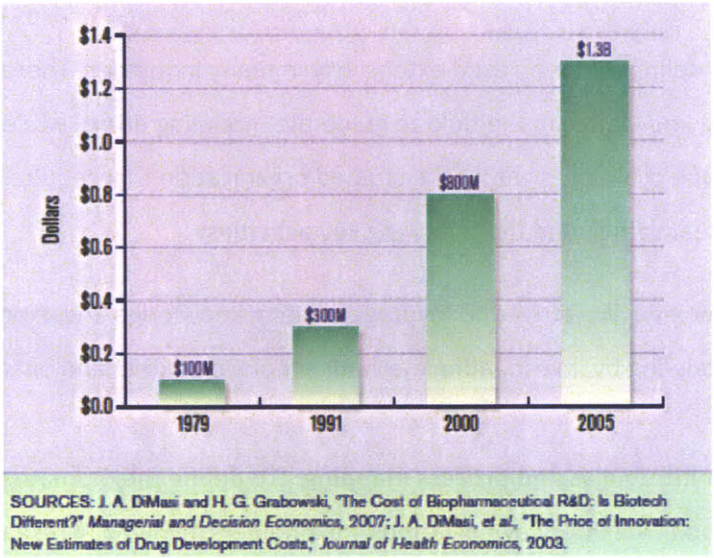


Figure 1: R&D Costs to Develop a New Drug Over Time (2)

Many researchers believe that the productivity concerns may be overstated. Berndt et al. found that productivity observations are "...neglecting to account for the contribution of incremental or follow-on innovation in the form of approvals obtained for new indications and formulations of drugs that are already in use probably results in a substantial underestimate of innovative output in biopharmaceuticals."(3) Nonetheless, the high industry costs and public concerns about productivity stress the importance of efficient use of R&D resources within pharmaceutical companies.

1.4 Company Background

Novartis AG is a global leader in pharmaceuticals, vaccines, generics, and consumer health products. Its mission is to discover, develop and successfully market innovative products to prevent and cure diseases, to ease suffering and to enhance the quality of life.(4) Headquartered in Basel, Switzerland, it consists of more than 100,000 employees in 140 countries worldwide.

Novartis Institutes for Biomedical Research (NIBR) is the global research organization of Novartis. Its research approach prioritizes patient need and disease understanding, and emphasizes

proof-of-concept trials – small-scale studies used to get an early read on the safety and efficacy of drug candidates.

NIBR's research activities are organized within various Disease Areas (DAs) and Platforms. NIBR's nine DAs focus on developing therapies targeting specific classes of diseases, and include departments such as Cardiovascular & Metabolic Disease and Oncology. Platforms groups provide research support to all DAs in a variety of specialized, cross-cutting areas such as Global Discovery Chemistry and Metabolism and Pharmacokinetics. In addition to DAs and Platforms, a Translational Sciences department supports activities in the later stages of NIBR's pipeline, helping to transition projects from research to clinical development.

Various support organizations provide services to research groups; these include functions such as Finance and Human Resources. One of these support groups, the Program Office, support other NIBR departments by providing "skills, analyses, tools, processes, and integrated information to enable effective decisions-making and management of our science."(5) NIBR's Portfolio Management Group resides within the Program Office and works to gain "an understanding of how the overall pipeline is developing, records project metrics, and analyzes lessons learned from these metrics."

NIBR's R&D focus is guided by two principles: does it understand the underlying cause of the disease, and does the disease represent a significant unmet need in patients(6). Noticeably absent is a stated focus on commercial and financial prospects of research efforts. This attitude was stressed in a recent *BusinessWeek* article stating that NIBR CEO Mark Fishman "banned running commercial analyses of new drug candidates until the company had sufficient clinical data. This approach, backed by Vasella, was heresy in an industry that spends vast sums trying to assign a hypothetical value to each potential drug at every stage of the R&D process."(7)

Novartis' decreased emphasis on commercial prospects during the early R&D process has not only strategic, but tactical implications. The creation of NIBR as a stand-alone research organization placed research portfolio decisions at an arm's length from the corporate sales and marketing functions. Headquartering NIBR in the United States, rather than co-locating with Novartis Pharma headquarters in Basel, Switzerland, further bolstered this separation. The BusinessWeek article acknowledged significant resistance from sales and marketing executives, who historically had held much of the decision-making clout.

NIBR's focus greatly impacted the research approach. Early in the project definition, it was made explicitly clear that efforts to model the R&D portfolio should not incorporate financial forecasts. As discussed in Section 2.3, this differs significantly from the typical applications of portfolio modeling in the pharmaceutical industry. As a result, we steered the research efforts towards understanding pipeline throughput, rather than quantifying expected financial returns.

1.5 Thesis Outline

This document is organized as follows:

- **Chapter 1** describes the motivation for thesis and introduces the research methodology.
- **Chapter 2** provides background on portfolio management in the pharmaceutical industry, and introduces the portfolio management challenges that motivate the proposed modeling solution.
- **Chapter 3** discusses the research methodology in detail by walking through each major component of the modeling process, including problem clarification, model scoping, model formulation, and model utilization.
- **Chapter 4** discusses the approach to validating the portfolio forecasting model and presents the validation results.
- **Chapter 5** evaluates the model's effectiveness in terms of accuracy and usability. In addition, it discusses organizational issues surrounding implementation within the organization, and includes recommendations for future work.

2 Pharmaceutical Portfolio Management

2.1 Drug Discovery and Development

Novartis' drug development process is typical of the pharmaceutical industry in general, and is represented in Figure 2. (6) Like R&D processes in most industries, it operates as a Stage-Gate™ process.(8) In such a process, projects progress sequentially through various stages of work; gates serve as checkpoints between stages where a decision-making body determines whether or not to permit passage of a project to the subsequent R&D stage based on various criteria.

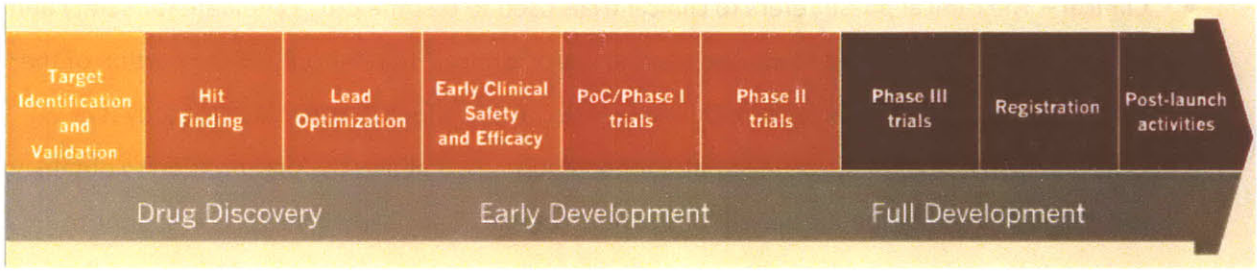


Figure 2: The Pharmaceutical R&D Process (6)

NIBR is primarily responsible for drug discovery and early development, with the ultimate goal of achieving positive readouts from proof-of-concept (POC) clinical trials. Once this milestone is reached, primary responsibility is transferred away from NIBR to downstream development functions. Therefore, we focus on understanding only the portion of the pipeline in which NIBR is involved, consisting of the following distinct R&D phases:

- **Target Identification (TI)** – At this phase, research activities seek to identify a biological target that may contribute to a particular disease. The intention is to develop a drug that interacts with this target to produce a therapeutic effect.
- **Target Validation (TV)** – At this phase, further activities validate the target's role in the disease.
- **Hit Finding (HF)** – At this phase, research begins searching for therapies to affect the validated target. The nature of the research activities depends on the *modality* of the therapy, or the molecule-type. For low molecular weight (LMW) therapies, also referred to as "small-molecule", assays are developed to test the effect of a large number of chemical compounds using high-throughput screening (most commonly). For biologics, large-molecule therapies are developed to bind to the target.

- **Lead Optimization (LO)** – Once a lead therapy is identified through HF, further work may be necessary to improve its behavior as a drug. An LMW compound may be chemically-altered to improve its properties, while antibodies may be altered to better bind to the target.
- **Candidate Selection Point (CSP)** – Although not an explicit milestone in all organizations, NIBR defines CSP as the point in which a drug candidate has been selected among other leads to enter preclinical testing.
- **Preclinical**[1] – Performed prior to commencing clinical trials in humans, this phase involves various testing of the candidate for safety and efficacy. Test platforms may include animal models and computer simulations.
- **Clinical** – The clinical phase refers to clinical trials used to test the drug candidate for safety and efficacy in humans. For this study, we use the word "clinical" to refer only to the portion of the clinical phase under NIBR's primary purview—specifically, the POC study. A POC study is a clinical trial in a small, well-defined patient population.(9) It allows the organization to obtain an indication of the therapeutic benefit or possible complications early in the process, prior to more substantial investment in Phase I or Phase II clinical trials. A positive POC study represents the endpoint of NIBR's primary pipeline responsibility, after which a project enters full scale clinical development.

In this report, we discuss two types of clinical phase projects that exist within NIBR. The first is a new molecular entity POC study, or NME POC, which is a study of a novel compound. In this report, we refer to these simply as POC studies. The second is a parallel indication extension POC, or PIE POC, which involves application of an existing compound to a new disease. In this report, we refer to these simply as PIE studies.

2.2 Challenges in Pharmaceutical Portfolio Management

2.2.1 Managing uncertainty

Pharmaceutical R&D is an uncertain business. As shown in Figure 3, research on 5,000 to 10,000 compounds is performed for every one drug that reaches the market. (2) Thus, the $1.3B cost often attributed to bringing a single drug to market in fact accounts for subsidization of the many failed efforts that a pharmaceutical company must absorb along the way. The industry as a whole spent $65.3B on R&D in 2009.(2)

[1] Screenshots included in this document may use the term "sPOC" when referring to the preclinical phase.

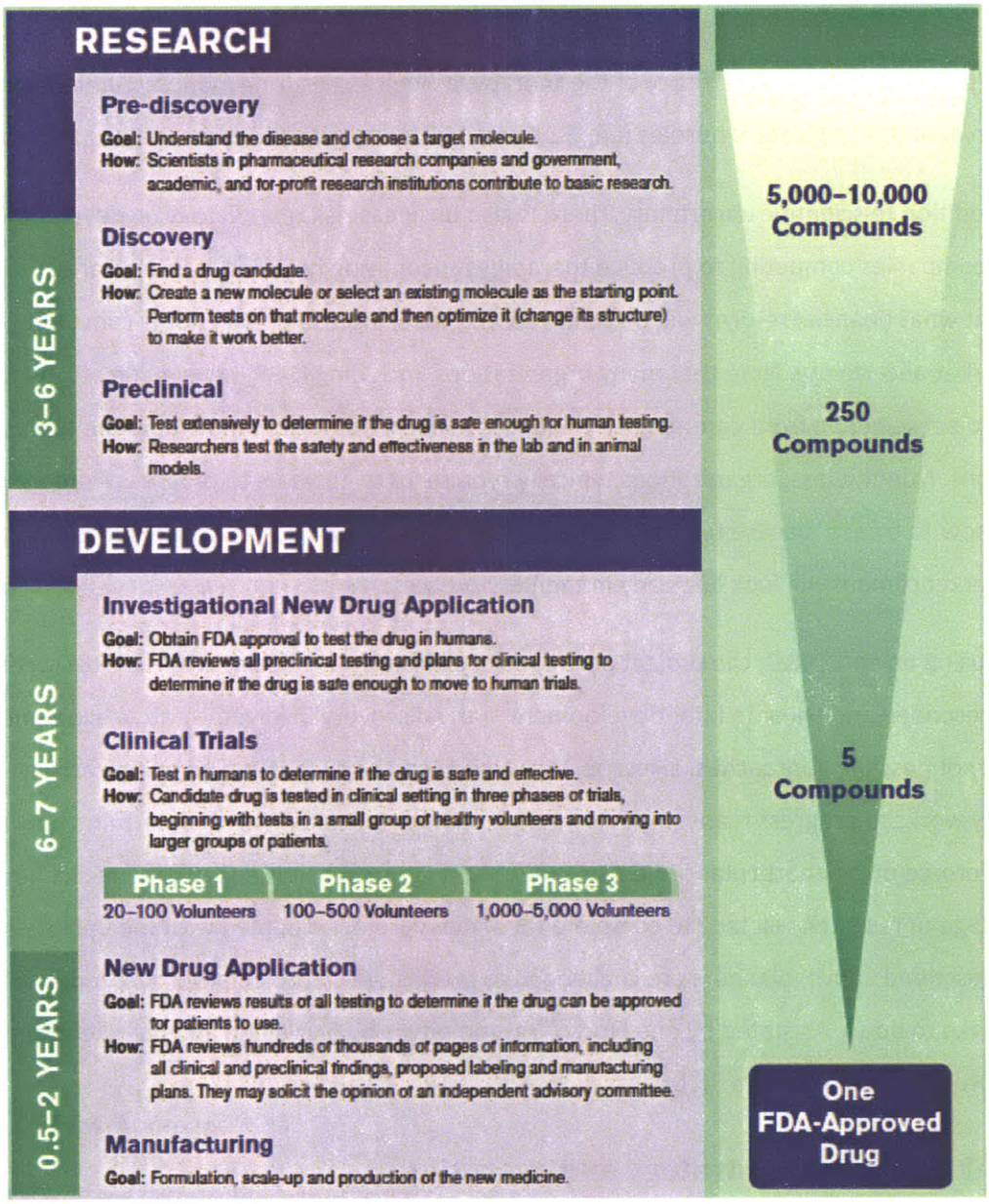


Figure 3: Timing and attrition in pharmaceutical R&D (2)

The high project attrition found in the industry results from the many hurdles to developing a successful drug. The most significant hurdle is the overall scientific challenge of developing a novel therapy to a disease. Drug development is scientifically unknown territory requiring discovery of new knowledge. Even as organizational competencies mature, such as in low molecular weight discovery, the industry continues to seek novel therapies in areas such as biologics and personalized medicine.(2) Because of this uncertainty, high attrition occurs at each phase of R&D. Early in the pipeline, attrition can occur due to inability to identify an effective target to treat a disease, or to find promising leads for

influencing a particular target. As projects progress through the pipeline, other hurdles include establishing sufficient safety and efficacy of the therapy, as well as ensuring manufacturability and scalability. In addition, regulatory hurdles (i.e., FDA requirements) gate the process along the way.

In addition to scientific uncertainty, there is also business risk related to drug development. With many companies competing to produce therapies for common diseases, it is unclear early in development what financial returns will be achieved. Likewise, disease prevalence is constantly in flux, creating market uncertainty. Note that many organizations, including NIBR, have made efforts to decouple project selection from commercial influences to avoid stifling of innovation due to business considerations. Nonetheless, organizations typically require 10 to 15 years to progress from a biological concept of how to treat a disease to commercialization of a therapy. Thus, it is quite difficult to predict what the market climate will look like early in the R&D process.

Focusing on early R&D, the project portfolio is particularly uncertain. Whereas R&D activities are relatively prescribed and linear in later development, early discovery involves iterative experimentation and a variety of possible approaches. Likewise, resourcing may be such that a given researcher concurrently works to progress many projects at once, making it challenging to determine how much work is performed on any particular project in a unit of time. As such, it is difficult to predict how long any given stage of research will take to complete for a given project. Coupled with the uncertain outcomes described above, planners are challenged to predict which projects will succeed, and how long it will take to do so. As stated by one study, "serendipity is highly valued and has likely hampered a rational approach to the problem."(10)

2.2.2 Linking portfolio strategy and execution

Pharmaceutical portfolio strategy in early R&D consists of various elements. As discussed, NIBR's primary goal is achievement of positive readouts from POC clinical trials, and thus, its portfolio strategy is to align the project portfolio to achieve this goal. The research revealed various strategically-relevant project attributes that can influence the balance of project types that the research organization outputs to development, as well as overall pipeline productivity. These strategic elements are not specific to NIBR, but are relevant to early drug R&D in general, and include:

- Project modality strategy – the balance of LMW and biologics therapies in the pipeline
- Clinical novelty strategy – the balance of NME POC and PIE POC projects in the pipeline

- Backup project strategy – the prevalence of backup projects in the pipeline, which address the same target/indication as another related project
- In-licensing strategy – the use of strategic partnerships or project/company acquisition to bring external projects into the organization at various stages of development

Portfolio management serves as a link between an organization's portfolio strategy and the actions that execute that strategy. As suggested above, portfolio strategy at NIBR includes goals for the numbers and types of projects that must reach particular R&D stages to achieve the desired portfolio output at the desired time. Thus, portfolio management defines the actions that are to execute this strategy by allocating resources to maximize the ability to do so.

The field of system dynamics refers to activities that serve to bring a current state in line with a desired state as a goal-seeking structure, represented Figure 4.(11)

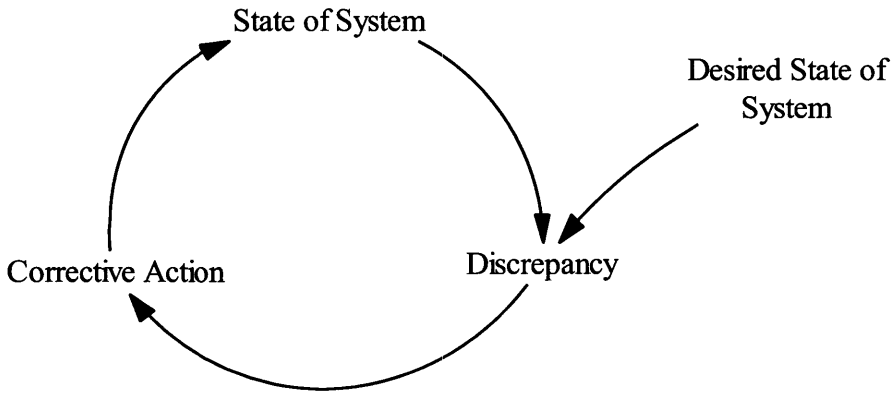


Figure 4: Goal-Seeking Structure

We would expect R&D portfolio management to behave like a goal-seeking system. That is, the process typically involves understanding the current state of the R&D pipeline, comparing the pipeline's output to the organization's goals, and resolving any discrepancy with corrective action such as reallocation of resources.

However, upon closer inspection, we observe some additional requirements for effective goal-seeking behavior that challenge the ability to label R&D portfolio management as such. First, the current and desired states of the system must be easily comparable to determine if a discrepancy exists. With an R&D pipeline, the discrepancy in question is whether or not the current state will output the desired

numbers and types of projects in the future. With this time delay between current and desired state observations, meeting this requirement would necessitate a deterministic system in which the current-state portfolio creates a known project output at a known time in the future. A second requirement for effective goal-seeking behavior is that the corrective action must be capable of modifying the current state of the system. With the R&D pipeline, this would require an ability to modify the current-state portfolio such that that any deviations from the expected project output could be remedied.

Unfortunately, these requirements are not easily satisfied in pharmaceutical R&D. As stated above, the drug pipeline is not deterministic, but rather, involves considerable uncertainty, including high attrition and high project-to-project variability in required time and resources. Thus, we cannot observe the current-state portfolio and predict exactly what it will output in the future. Similarly, when discrepancies between project goals and actual pipeline output are observed, it is not always possible to implement corrective actions to resolve these discrepancies in real-time. That is, since projects take significant time to progress through each stage of R&D, changes in the early project portfolio take years to progress through to changes in pipeline output. Failing to meet these requirements, portfolio management fails to function as an effective goal-seeking system, as represented in Figure 5. Rather, we are unable to compare today's portfolio with the desired future portfolio, and therefore, are unsure which actions to take to close any discrepancy that may exist.

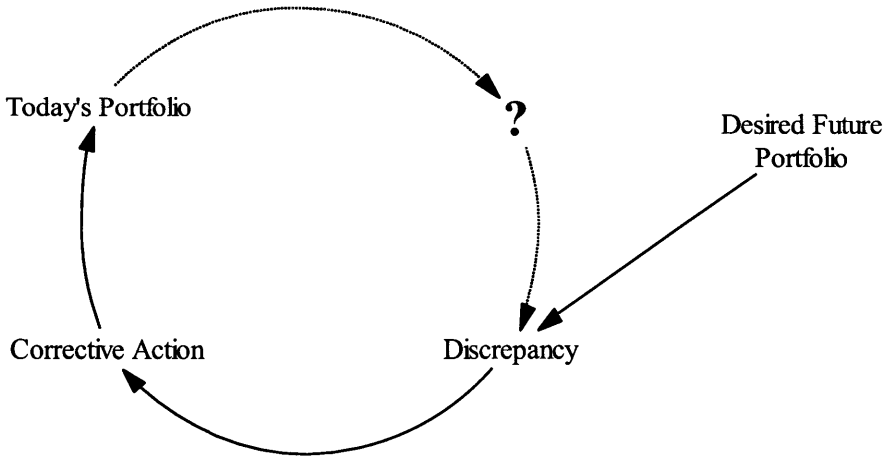


Figure 5: Portfolio Management as a Broken Goal-Seeking System

To create a working goal-seeking system, we must solve the two stated challenges—i.e., the inability to compare the current and desired portfolio states due to their separation in time, and the inability to make corrective actions once discrepancies are observed. We propose that creating a

computer-based forecasting model of the portfolio offers one approach to addressing these challenges, as suggested in Figure 6. If a model can create a forecast based on the current portfolio, this forecast can then be compared to the desired future portfolio. Any observed discrepancies can then be remedied with corrective action to today's portfolio. Managers can experiment with different corrective actions in the model to determine the most appropriate action to remedy the discrepancy.

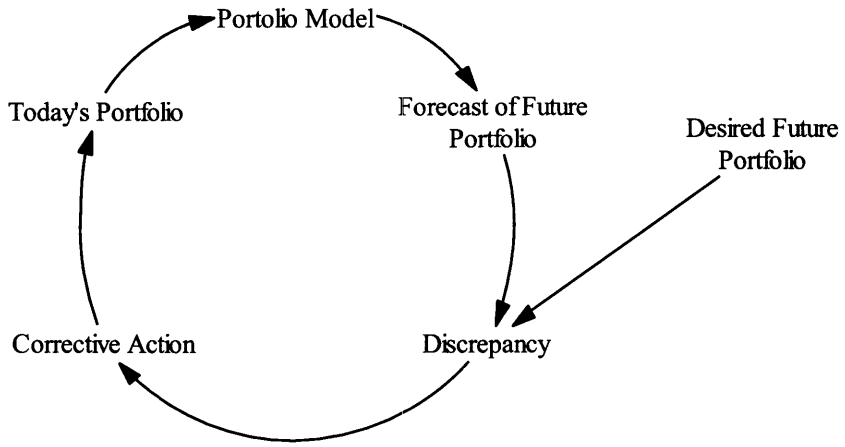


Figure 6: Model-Assisted Portfolio Management as a Goal-Seeking System

2.3 Historical Use of Modeling in Pharmaceutical Portfolio Management

Quantitative modeling techniques have been used extensively to support portfolio management in the pharmaceutical industry, although the literature base largely addresses clinical development forward; early drug discovery is left to more qualitative techniques. In discussing probabilistic pipeline modeling, Blau et al. explains that "complexity, creativity and iterative nature of the discovery process make it difficult to lay out a time course flow of events. However, once a new candidate molecule is found by frequently quite circuitous routes...the subsequent steps in the development and commercialization are more well-defined."(12) Similarly, Bode-Greuel and Nickisch explain that there is "general agreement in the pharmaceutical industry that it does not make sense to do detailed financial analyses in the research stage. The main argument is that at this point efficacy and differentiability might not have been established and that the effective does in man and the COGS are not known."(13) Rather, portfolio management efforts in research organizations tend to focus on the number of project at different phases in the pipeline to ensure productivity targets are met, and on the portfolio risk structure with respect to balance across different types of projects.

In drug development, we observe various modeling approaches, often combining probabilistic methods with financial reward projections to forecast portfolio returns. Tang et al. describes use of Monte Carlo simulation over a Bayesian Network to model risk associated with bringing new compounds to market.(14) Similarly, Blau et al. employs probabilistic network models and financial information to forecast portfolio risk and reward.(12) Subramanian et al. integrates both simulation and optimization under constraints into a Simulation-Optimization Framework.(15) With this approach, simulation approaches account for task duration and success uncertainty while optimization approaches account for project task scheduling under resource constraints.

As has been discussed, the costs of bringing a drug to market are substantial. Much of the cost is often attributed to costly clinical trials that must occur before FDA approval of a new drug. These costs likely drive the rigorous portfolio analysis techniques at the clinical stages and beyond, grasping for an NPV value to support the rapid escalation of cost per project. However, Mohr et al. explains that "while individual project budgets/investments in the research phase are relatively small compared to late development projects, the overall budget spent on research is nearly comparable."(10) The article suggests that while results of individual research activities may be difficult to predict, it is likely possible to effectively model the R&D process as a whole.

We propose that, given historical information about performance of the R&D pipeline as a whole, modeling techniques such as stochastic parameter representation and Monte Carlo simulation are applicable and useful to earlier R&D phases as well. After all, a study on the effectiveness of portfolio management across companies and industries found that the top 20% of businesses with respect to portfolio performance place more importance on portfolio management.(8) The two competencies that most greatly differ between top and bottom performers are "portfolio balance—achieving the right balance of projects" and "the right number of projects for the resources available". These attributes are not only relevant in later drug development, but can be readily understood through application of modeling techniques in early drug R&D.

3 Research Methodology

This chapter presents the research methodology used to develop the forecasting model. Generally, the approach moves sequentially from understanding the end-application, through model scoping and development, to utilization of the model to drive decision-making. Specifically, Section 3.1 describes the structure and dynamics of the R&D pipeline gathered through stakeholder interviews and internal company research. Section 3.2 explores how the structure and dynamics of the pipeline are converted into model specifications; we refer to this exercise as Model Scoping. Section 3.3 provides detail on the model structure and logic that implements the desired specifications. Finally, Section 3.4 presents two case scenarios that illustrate use of the model to drive strategic portfolio decisions, and discusses the model's sensitivity to changes in input parameter values.

3.1 Problem Clarification: Understanding the Real-World R&D Pipeline

Both to aid model accuracy and management buy-in, we sought to develop an intuitive model that mirrors the structure and dynamics of the real-world R&D pipeline as closely as possible. We therefore spent significant effort understanding the mechanics of the pipeline. Key aspects of this research are discussed below.

3.1.1 Pipeline Phase Dynamics

As discussed in Section 2.1, the R&D pipeline consists of a various sequential phases. To model how projects progress through this pipeline, we must characterize project activity within each phase, as well as project movement between phases. These dynamics are illustrated using the "bathtub analogy" in Figure 7. The level of water in the bathtub is equated to the quantity of projects in-process at a given time. This level is regulated by various project flows—specifically, the rate of project transitions from the previous phase, the rate of project transitions to the next phase, and the rate of project terminations, which remove projects from the pipeline.

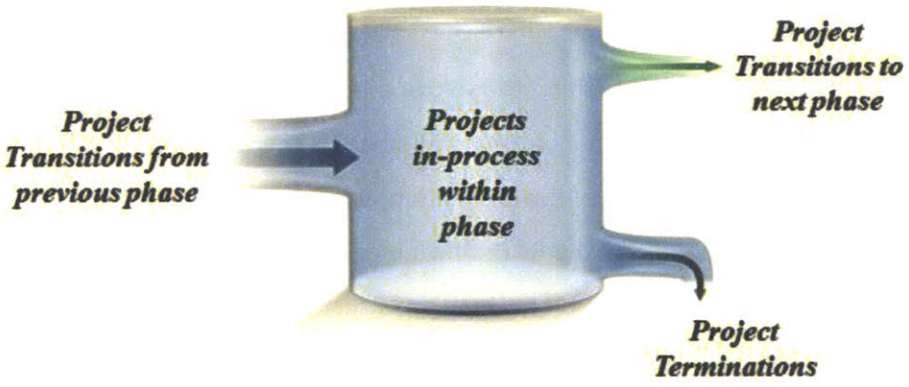


Figure 7: Pipeline Phase Dynamics

3.1.2 Project Progression

Understanding the project flow dynamics of each R&D phase individually, we must then understand the forces impacting how projects progress through the pipeline, from phase to phase. Research reveals two primary drivers: phase duration and phase transition rate. The phase duration defines the amount of time that a particular project spends in a particular R&D phase, spanning from the date that it transitions from the previous phase to the date that it transitions to the subsequent phase. The phase transition rate defines the percentage of all projects entering a particular phase that will ultimately transition to the subsequent phase. Together, these two parameters define the quantity of projects that flow through the pipeline, and the speed at which they do so. It is the uncertainty in these two variables that challenge the portfolio management process, and therefore, are the variables we seek to accurately represent in the model to capture overall pipeline progression dynamics.

Phase duration differs for every project. As such, to capture the real-world nature of project durations, we must define a model structure and behavior that accounts for the stochastic nature of this variable. Likewise, although transition rates are calculated for an aggregated set of projects, they too may vary from year to year; thus, they too should be represented stochastically. The uncertainty in these two progression variables drive the decision to rely on simulation modeling, since each instance of a forecast will yield unique results. Subramanian et al. refers to "multiple Monte Carlo time lines" to represent the outcomes of stochastic project progression. (15) Borrowing from this framework, Figure 8 illustrates the stochastic progression of projects from a hypothetical launch phase P0 through a 5-year planning horizon. Each project follows a unique pace of progression from phase P1 to P4, and some will terminate out of the portfolio (i.e., they will fail to transition) during the planning period. Effective

portfolio decision-making in these real-world conditions requires an understanding of the distribution of possible portfolio outcomes sometime in the future.

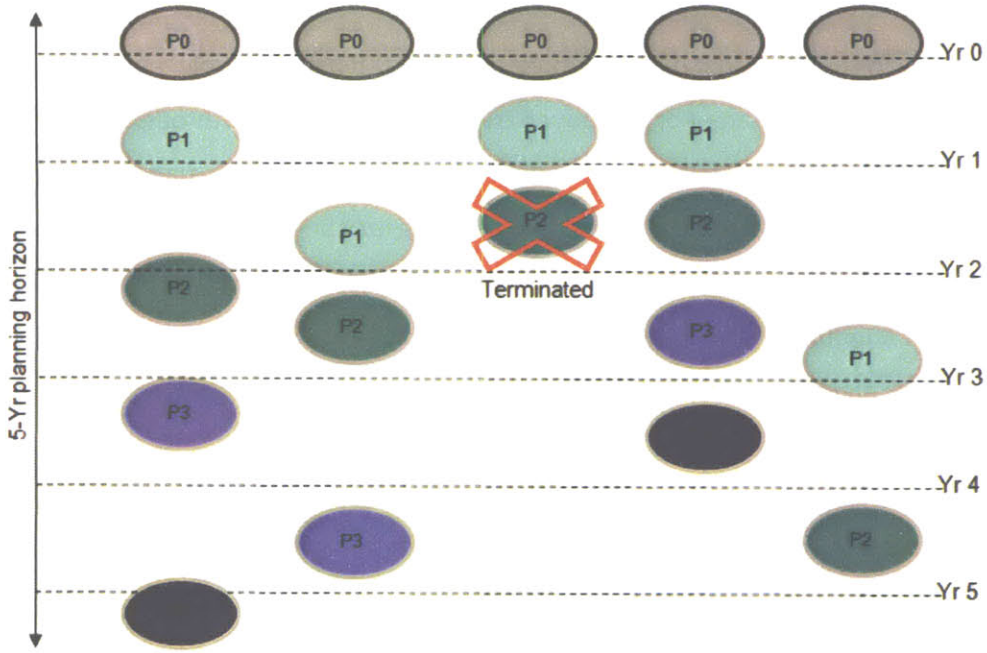


Figure 8: Multiple Stochastic Time Lines

3.1.3 Pipeline expansion points

Research reveals that the pharmaceutical R&D pipeline is not a strict funnel, with projects beginning at some starting phase and only progressing through the pipeline or terminating. In fact, there are also instances in which the number of projects can effectively amplify at "mid-pipeline" phases, which we refer to as "expansion points".

There are three expansion points that contribute significantly to the total number of projects in the portfolio. The first—backup project creation—typically occurs at the CSP phase. To increase the likelihood of eventual development of therapy for a specific indication, R&D managers may choose to "backup" a candidate compound with one or more projects to treat the same target and indication. The resulting backup projects are launched at the CSP phase and progress through the pipeline as would any other project.

A second expansion point—PIE study creation—occurs at the clinical phase. A clinical study for the first therapy to treat a novel target and indication is referred to within NIBR as the NME POC study,

or simply, POC. If researchers determine potential applicability of the drug candidate to an alternate disease, they may launch a PIE study to investigate this parallel indication. The new PIE project will typically launch at the clinical phase and progress independently from the POC study.

A final expansion point that can contribute to the R&D pipeline is project in-licensing. In-licensing refers to accessing a target or compound from an external source through a strategic partnership or outright acquisition of a company or technology. In these cases, the new project can be treated as entering the internal project pipeline. In-licensing may be utilized for various reasons, such as to gain access to specific intellectual property, external capacity or resources, or to fill in gaps in the internal portfolio. Although in-licensing can theoretically occur at any R&D phase, it is most common at LO, CSP, preclinical, and clinical phases.

3.1.4 The model pipeline

The project team reached consensus on the representation of the real-world R&D pipeline process flow shown in Figure 9. This depiction captures the most important attributes of the real-world pipeline, while intentionally omitting those attributes deemed insignificant to avoid unneeded model complexity. Each phase is represented as a stock of projects, with various project flows regulating these stocks. Feeding the first phase is a yearly *incoming projects* assumption. Projects flow through the pipeline based on distinct phase transition rates, which define the project transitions and project terminations for each phase, and phase durations, which define how long projects remain in each phase. Finally, projects are added to the portfolio mid-pipeline via three mechanisms: in-licensing projects from other organizations, which typically can occur at any point between LO and Clinical; launching of backup projects at the CSP phase; and launching of PIE projects at the Clinical phase.

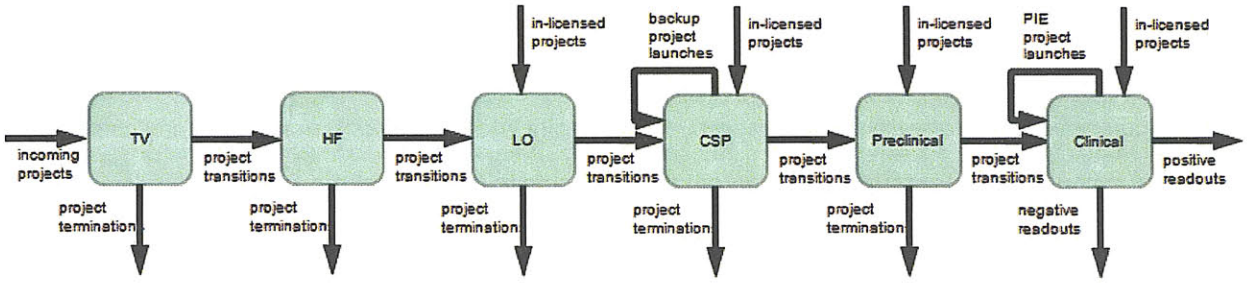


Figure 9: Simplified R&D Pipeline Flow for Modeling

One additional portfolio attribute, project-type differentiation, is not depicted in the figure but is relevant to the model. Projects can be differentiated along many dimensions, such as modality (small-molecule versus biologic), disease area (cardiovascular versus oncology), and clinical project type (NME POC study versus PIE POC study). Projects of a particular type might categorically have different phase durations and transition rates than those of another type. To avoid unnecessary complexity, we worked with the project team to identify the most important dimensions of differentiation to capture in the model, and selected modality and clinical project type. Projects of a particular type are tracked and quantified separately as they progress through the pipeline.

3.2 Model Scoping

Prior to model development, we developed a simple framework to specify the model's key inputs and outputs, depicted in Figure 10. The purpose of this framework is to organize the information gathered during the problem clarification step into actionable specifications that can drive model development. To draw an analogy to product development, the problem clarification step gathers the marketing specifications, whereas model scoping defines the engineering specifications for the model. The results of the model scoping efforts are discussed below.

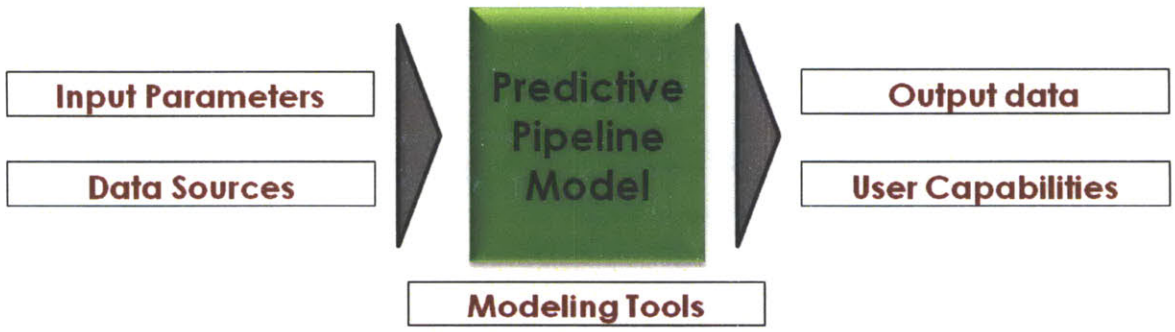


Figure 10: Framework for Model Scoping

3.2.1 Input parameters and assumptions

The model's input parameters are those project and portfolio performance attributes needed to recreate the pipeline structure and dynamics described in Section 3.1. By defining the input parameters needed, we can then gather historical data to quantify them. The required input parameters are:

- **Phase duration**: Coupled with the phase transition rate, phase duration is a key parameter describing how projects progress through the pipeline. To accurately progress projects in time,

the model requires definition of how long projects remain in each pipeline phase prior to transitioning to the subsequent phase. As described in Section 3.1.2, differences in duration from project-to-project are a significant source of pipeline uncertainty. As such, duration must be represented stochastically in the model. Ideally, we aim to understand the frequency distribution of all possible project durations, and progress projects assuming a random sample from this distribution.

- **Phase transition rate:** Coupled with the phase duration, phase transition rate is a key parameter describing how projects progress through the pipeline. For all projects entering a particular R&D phase, only a portion will eventually transition to the subsequent phase. The model requires definition of what portion will successfully transition from each phase. Whereas durations are project-specific, transition rates pertain to a group of projects. Transition rate uncertainty relates to year-to-year variation in overall transition rates.
- **Current projects:** The model requires a current state of the portfolio from which to forecast. The current state is the quantity of projects in each pipeline phase, delineated by project type (i.e., modality, PIE/POC). A challenge in creating this portfolio snapshot is that progress *within* a phase differs for each project at any moment in time—i.e., one preclinical project may just have begun the phase, whereas another may be nearing completion. Resolution of this issue is closely linked to the choice of model logic itself, and therefore, will be discussed in detail in Section 3.3.3.
- **Incoming projects:** The forecast focuses on a particular portion of the R&D pipeline—specifically, TV to Clinical. Each year, as projects progress, new projects will enter the beginning of drug pipeline at TV[2]. Thus, forecasting pipeline throughput requires an assumption of how many projects (delineated by project type) will enter each year. That said, the actual number of incoming projects is likely to differ from this assumption, and therefore, this parameter too should be defined in the model as uncertain.
- **Expansion points:** The model requires a mechanism to incorporate pipeline expansion points, described in Section 3.1.3. As such, we must define the prevalence and timing of backup project launched during CSP, the prevalence and timing of PIE studies launched during clinical, and the prevalence of in-licensed project launched at LO, CSP, preclinical, and clinical phases.

[2] The project team identified use cases requiring the ability to start the model at either TV or LO, based on user selection. As such, the model must be flexible to allow either scenario, requiring an *incoming projects* assumption for both phases.

3.2.2 Data Sources and Data Collection

A centerpiece of our hypothesis is that past portfolio performance can be used to predict future performance. Thus, equally important to the mechanics of the model is the input data used to drive it. Through the research we identified three available sources of historical project performance information: internal information systems, industry benchmarking organizations, and interviews with research management personnel. For each input parameter, we determined the best available source(s) among these options, and collected appropriate data accordingly.

Like many organizations, NIBR captures various types of project information in an array of information systems. Objective, empirical project data is potentially the most valuable data source as it directly reflects past organizational performance. As such, significant effort went into collecting historical project data to serve as the basis for the model's pipeline forecasting. Nonetheless, with the promise of this data comes challenges related to data integrity (i.e., was project information accurately recorded), data availability (i.e., is data easily accessible or scattered among many disparate systems), and data interpretability (i.e., can the needed information be extracted from the raw data).

Access to internally-captured project data is particularly vital to the model for determining past project performance—specifically, phase durations and transition rates. As discussed in Section 3.1, effective and useful pipeline forecasting depends on accurate capture of uncertainty in these parameters. By looking at phase duration data for a large quantity of past projects within NIBR, it is possible to create a distribution describing the prevalence of different phase durations across the entire portfolio. Likewise, project data can be used to determine not only mean transition rates from phase to phase, but also the year-to-year variability of these transition rates. The other two data sources discussed below provide mean statistics for these performance characteristics, but do not adequately capture uncertainty.

Although we aimed to base the analysis on internal historical data whenever possible, such data was not available for every data element needed for the model. Industry benchmarking data provided an additional source of historical data for the pharmaceutical industry as a whole. In particular, we relied on a proprietary industry analysis from KMR Group (16), which provided various aggregated metrics on R&D performance for eight leading pharmaceutical companies, as well as Novartis' individual performance.

Interviews with process experts provided a third data input. These interviews complemented the empirical data set by providing a means to verify accuracy of historical data and identifying organizational dynamics that influence how historical data should be treated. (For example, changes in milestone definitions over time required us to limit the data set to specific years.) Within NIBR, we specifically targeted research project management personnel within the Disease Areas and Platforms, who held knowledge about their departments' past performance.

For confidentiality purposes, we have omitted Novartis' actual data from this document. However, we discuss below the general approach to data collection and analysis for the various input parameters in the model.

3.2.2.1 Phase Durations

The model is designed to assume that each future project will perform similarly to some project in the past. Based on real-world pipeline progression mechanics, we sought to collect the empirical frequency distribution of phase durations for historical projects. Our initial intention was to then model the historical data by fitting an appropriate known distribution to the empirical data. As will be discussed, we were not able to rely entirely on historical data due to limitations in data set size and historical data quality. As such, the Portfolio Management team collectively determined that a combination of historical project data, industry benchmarking data, and interview-based data would be used to reconstruct the needed data set.

At the time of the project, the best available source of historical phase duration data was an internal portfolio data captured by NIBR's Portfolio Management Group. This report included milestone dates for a large number of historical projects, allowing determination of relevant phase durations for each project, LO through Clinical. The data did not include project data for TV or HF. For these phases, we chose to utilize industry benchmarking data taken from the 2009 KMR report.

In assessing the raw data, we observed that some projects were reported to spend unrealistic amounts of time in a given phase (e.g., 0 years). Extremely short durations suggest improper data entry, such as a user entering the same milestone data for multiple phases, while extremely long durations suggest that a project perhaps remained dormant, without active work for a significant amount of time. Initial statistical analysis of the data set revealed that inclusion of this apparent outlier data could greatly impact the statistical model of the data set, which would in turn impact the forecasting accuracy

of the model. Thus, we chose to exclude unrealistic data from the data set that would ultimately drive the model.

We used interview-based input to select reasonable cut-off points for the data set, prior to attempting to fit an appropriate distribution to the data. Conversations with research management indicated that capturing projects with durations between 0.3 and 2.5 years for each phase would capture most of the relevant data for the purposes of the model. As such, we treated projects with durations outside this range as outliers, and excluded them from the data set prior to statistical analysis.

We used a statistical software package, JMP, to analyze the truncated duration data with the goal of determining appropriate probability distributions to represent the historical duration data in the portfolio model. When multiple "good fits" were available, we selected those which could be easily modeled with Excel's built-in distribution functions—e.g., lognormal and Weibull distributions.

Figure 11 illustrates the approach to distribution fitting. Due to data confidentiality, we have obscured all numerical data indicating actual phase durations. The JMP output provides a visual depiction of the data set and estimated distribution, selected moments of the data set (e.g., mean, standard deviation), parameter estimates for the fitted distribution, and a p- value for the goodness-of-fit test.

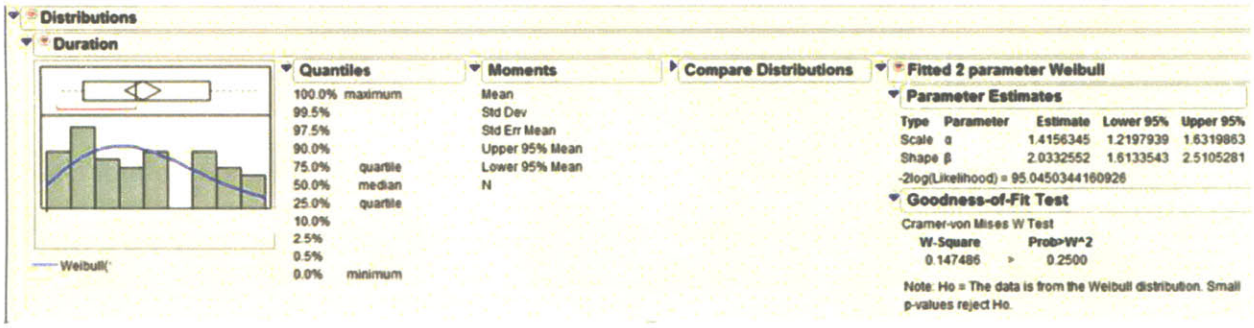


Figure 11: Duration Distribution Fitting with Moments and P-test

From the statistical analysis, we were able to fit either lognormal or Weibull distributions to duration data for LO, CSP, and preclinical, both LMW and biologics projects, as illustrated in Figure 12. That is, for the goodness-of-fit tests, the p-values fail to reject the null hypothesis that the empirical data comes from the respective distributions in all cases (i.e., p-value>0.05). However, for the clinical phase, we were not able to fit a distribution that passed the goodness-of-fit test.

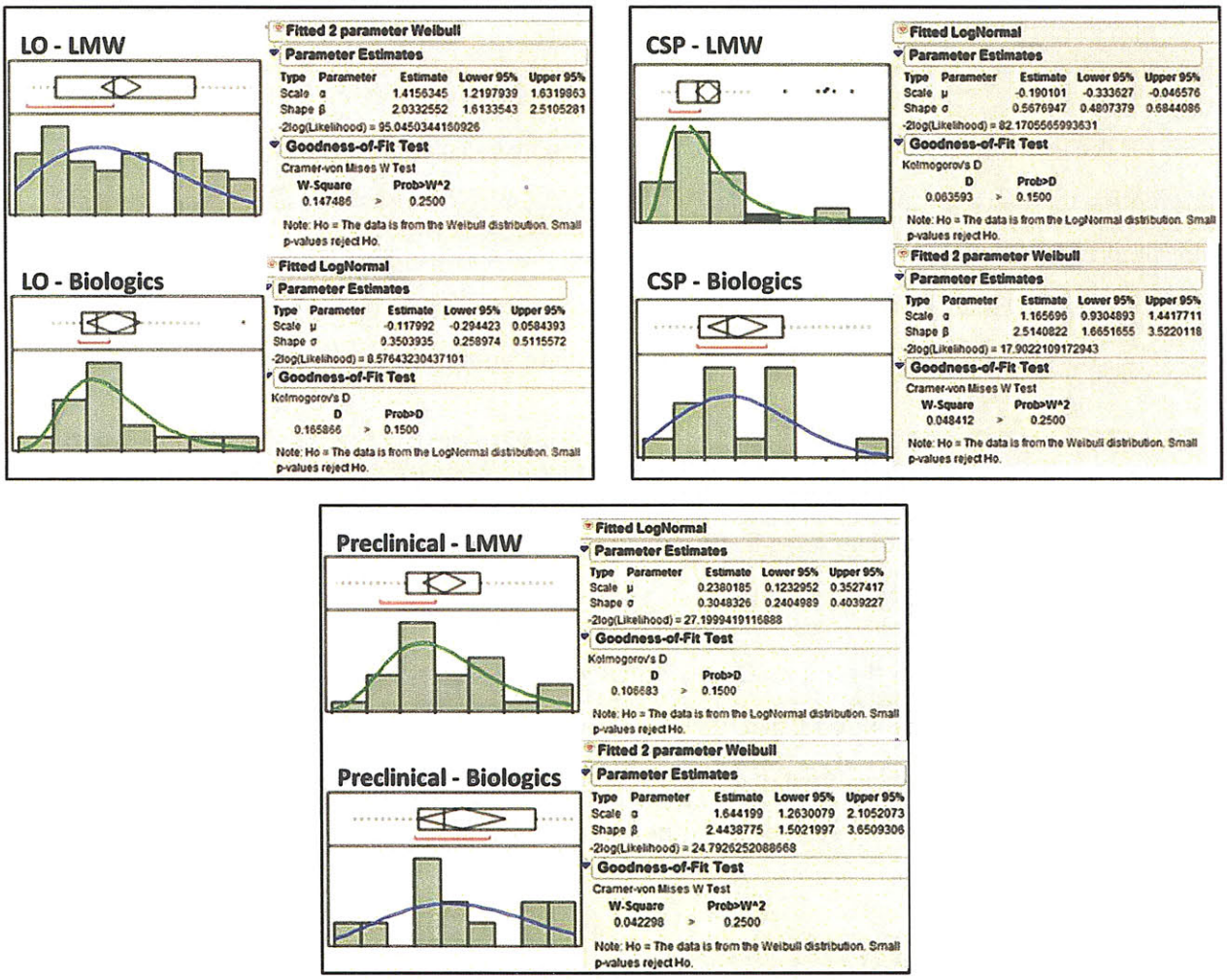


Figure 12: Fitted Distributions for LO, CSP, and Preclinical Phases

Despite determining fitted distributions that pass the goodness-of-fit hypothesis test for three phases, the JMP analysis revealed some concerns regarding our approach to modeling the historical data set in general. Specifically, the numbers of projects analyzed decreases from LO to clinical, decreasing our confidence in the results at the later portfolio phases. With small sample sizes, we were not confident that the fitted distributions truly reflected the expected durations of future projects.

In parallel with the data analysis, we also determined that it would be difficult to utilize a distinct type of distribution for each R&D phase in the forecasting model (e.g., lognormal for CSP and Weibull for LO). This would greatly add to model complexity when compared to relying on a single type of distribution for all phases.

With both data set limitations and modeling challenges, we chose to alter our approach to modeling the historical duration data for inclusion in the model. Based on our research and discussions with research management, we chose to use the Beta distribution to model each research phase. The Beta distribution was selected based on various benefits:

- The beta distribution is a flexible distribution defined by two shape parameters, α and β. As such, depending on the choices of these parameters, the distribution can appear symmetrical (i.e., similar to a normal distribution), or skewed to one side or another. For modeling purposes, this flexibility allows us to dynamically adjust the distribution to different phase assumptions.
- The model assumes stochastic phase durations defined by the PERT (Program Evaluation and Review Technique) methodology. With this technique, durations are defined by three parameters: optimistic duration **a**, likeliest duration **m**, and pessimistic duration **b**. These parameters, can be easily captured through both historical data analysis and expert interviews, and can be easily transformed into α and β parameters of a beta distribution for implementation in Excel. When used together, a beta distribution defined by the PERT parameters is referred to as a BetaPERT distribution.
- Excel's built-in functions not only offer the BETA function, but also the BETAINV function, which simplifies modeling of "random draws" from the beta distribution. (Similar built-in inverse functions are not available for the Weibull and Lognormal functions.)
- The BetaPERT distribution is a well-accepted approach to modeling R&D task durations.

To define the parameters of the BetaPERT distribution, we utilized information from both the empirical historical data and interviews with research management. Specifically, for LO through clinical, the mean value of the empirical data for each phase was selected as the "likeliest" phase duration. For TV and HF, the 2009 KMR report was used to determine Novartis' mean phase durations. To determine the optimistic and pessimistic durations for all phases, we interviewed research project managers from four DAs. For each, we asked them to identify upper- and lower-bounds[3] for the duration of each phase TV through clinical, differentiating by project type. We used the average of responses across the four DAs to define the distribution parameters.

[3] Upper-bound is the longest amount of time that a project would spend in a given phase under particularly bad circumstances (e.g., unexpected delays, the longest or most difficult type of project, etc). Projects should take this long no more than 1% of the time. Lower-bound is the minimum about of time that a project would spend in a given phase under ideal circumstances, with everything going right for the easiest/fastest types of projects.

3.2.2.2 *Phase Transition Rates*

For the purposes of the study, we define transition rate as follows: For all projects that enter a particular R&D phase, the transition rate is the percentage of projects that ultimately transition to the subsequent phase. Transition rate calculation is irrespective of the phase duration—i.e., it captures if a project will ever transition, regardless of how long it takes.

As with the duration analysis, we utilized internal portfolio data to determine historical transition rates for CSP, preclinical, and clinical phases. For TV, HF, and LO, we utilized industry benchmarking data from the 2009 KMR report.

To describe the approach for determining a particular phase's transition rate, we will use CSP as an example. We first determine the number of projects in the data set that include a CSP date. Of those, we tally those with a preclinical date as well. Of those *without* a preclinical date, we must determine which were terminated, and which remained active in the portfolio. For the latter group, we conclude that these project essentially had not yet reached a "final state"—i.e., transition or termination. As such, we must back this quantity out of the original CSP project tally. With these tallies, we can now perform the following simple calculation.

$$Transition\ rate_{CSP} = \frac{(CSP\ date\ with\ preclinical\ date)}{(\ CSP\ date_{all}\ ) - (CSP\ date\ without\ preclinical\ date_{active}\ )}$$

For modeling purposes, the project team requested an ability to control variability in the transition rate from year to year. Although insufficient data was available to empirically determine the yearly variation in transition rates at each phase, we included in the model a mechanism to add a variability assumption. Specifically, the model uses the entered transition rate as the mean of a normal distribution. The user can also enter a standard deviation for this distribution, which defaults to a value of 5% for all phases. While the model initializes with these distribution parameters, the model randomly calculates an "actual" transition rate from this distribution for each year within each simulation trial. (That is, the user may assume a mean LO transition rate of 50%, but in any given year, the "actual" transition rate used in the model may be slightly lower or higher, based on the random draw from the normal distribution.)

3.2.2.3 *Current Projects and Incoming Projects*

Generally, the number of current projects should be entered by the model user at the start of each simulation. Model defaults were set based on the current portfolio at the time of the study.

Forecasting with the model requires an assumption of projects entering TV or LO (depending on the chosen starting phase) each year. To determine default values to include in the model, we interviewed various stakeholders within NIBR to determine historical projects inflow rates to these phases. Discussing this data with research management, we agreed to choose model defaults the roughly represented 2009 project inflows. The model user can then adjust from these defaults as needed.

As with transition rates, we implemented variability in actual incoming projects by employing a normal distribution with mean as the user-entered average project assumption, and standard deviation as the user-entered "+/-" value.

3.2.2.4 Backup and PIE Projects

The project team collectively determined that it was vital for the model to capture two real-world portfolio attributes in which the project count can effectively amplify, which we refer to as "expansion points" in the model. Specifically, a given CSP project may be followed by one or more backup projects, and an NME POC clinical study may be followed by one or more PIE POC studies.

We employed a similar data collection strategy for each of these portfolio expansion points. We first considered the model logic to determine exactly what data was needed. For modeling purposes, we assume that some proportion of projects that enter CSP will result in 0 to 5 backup projects; likewise, we assume that some proportion of projects that reach NME POC will result in 0 to 5 PIE POC projects. Thus, we treat CSP entry and NME POC entry as "trigger points" that launch new projects in the model.

To work within this model logic, we needed historical data describing the prevalence of expansion projects. For each trigger point reached historically, we determined the prevalence of launching 0, 1, 2, 3, 4, or 5 projects, as well as the mean time to launch each expansion project. Together, the calculated proportions and mean time-to-launch served as the default values for expansion point prevalence and timing employed in the model.

3.2.3 Modeling Tool Selection

Modeling tools must be suitable for capturing the structure and behavior of the real-world system, and acceptable to the end-user to ensure adoption of the model. Guided by these criteria, we evaluated available platforms on which to build the pipeline model.

Microsoft Excel is a logical choice for input and output of numeric data. Managers are typically knowledgeable of the operation and features of the software, and it is widely available in many organizations, including NIBR. To accommodate the uncertainty noted in the input parameters, we chose Oracle Crystal Ball for Monte Carlo simulation, which integrates seamlessly with Excel. Finally, we employ Excel's built-in Visual Basic for Applications (VBA) programming platform for data manipulation. Compared to relying on worksheet functions within Excel, VBA programming provides greater flexibility and control over the data processing that underlies the forecasting model.

Many other simulation platforms are available that can potentially support portfolio forecasting. However, for the purposes of prototype modeling, we gravitate towards an Excel-based solution based on its minimal learning curve and financial investment. Over time, if growth in model complexity due to feature expansion exceeds feasibility within Excel, additional modeling platforms can be evaluated.

3.2.4 Output Data

The model's output should provide a forecast for those metrics of strategic significance for portfolio decision-making. As such, we relied on input from NIBR's Portfolio Management Group in selecting relevant output metrics, as well as review of existing portfolio reports and dashboards. Specified metrics fall into three categories: portfolio size, clinical readout, and phase transition metrics.

Portfolio size metrics illustrate the numbers of projects in relevant portions of the R&D pipeline in a given forecast year. These metrics provide management with information as to the health of the portfolio; that is, the number of projects currently in R&D should correlate to the number of successful clinical studies at a later date. Those metrics of strategic significance to NIBR, and thus, those relevant to the model are the sizes of the CSP, preclinical, and clinical portfolios individually, and the size of the NME portfolio (i.e., the sum of the CSP portfolio, preclinical portfolio, and POC clinical portfolio). In addition, NIBR wishes to forecast the percentage of biologics projects in the portfolio.

Clinical readout metrics provide the tallies of the clinical study readouts during a given forecast year. These are arguably the most important output metrics, since the ultimate goal of the research organization is to deliver positive studies to downstream development. The relevant metrics include total positive readouts, total negative readouts, positive POC readouts, and positive PIE readouts.

Finally, phase transition metrics tally the number of phase-to-phase transitions that occur in a given forecast year. They offer a sense of the productivity of each phase individually.

3.2.5 User Capabilities

User capabilities define how the end-user wishes to interact with the model. Questions include:

- How much technical aptitude is expected of the end-user to utilize the model?
- How much time can exist between beginning a model scenario and achieving a usable output?
- How should the output data appear?
- What level of user control of assumptions is required?

Portfolio assessment and decision-making is made at a fairly high level in most organizations, including NIBR. This audience is primarily interested in dashboard-type model outputs, and likely prefers not to undergo any significant learning curve in order to obtain these outputs. This audience suggests two primary options for model design. The first alternative is a technically-complex model that would be managed by an analyst within the organization; leadership would request dashboard reports based on specified assumptions, and model output would be delivered at a later time. The second alternative is a simple, intuitive model that can be used by leadership for real-time decision-making. Given a relatively large number of portfolio levers, the first, transactional approach would limit the usability of the model for quick-turnaround scenario analysis. We therefore gravitated towards a simple model design that supports real-time adjustment of assumptions and observation of resulting output. Technical complexity should be hidden from the user. Furthermore, the model requires sufficient speed such that various changes to portfolio levers can be manipulated to investigate different scenarios.

Given the described audience and use case, another important consideration is output data display. The problem space suggests a potentially overwhelming number of data points in a given output—i.e., 14 relevant output metrics for each of 11 forecast years yield 154 forecast metrics. Furthermore, when we consider the need to quantify uncertainty in each metric over a given forecast simulation, we realize the need to present the most relevant metrics in an intuitive, digestible manner through customized output reports. We can model these reports on existing portfolio reporting methods to aid adoption and again minimize the learning curve.

Finally, we consider how the user interacts with the model with respect to input parameter adjustments. As discussed in Section 3.2.2, significant effort went into data collection and analysis that defines default input parameter values. However, scenario analysis may require adjustment to those default values. For example, a manager may wish to observe the impact of improved efficiency by decreasing phase durations, but must be able to return to the default historical values as needed. This

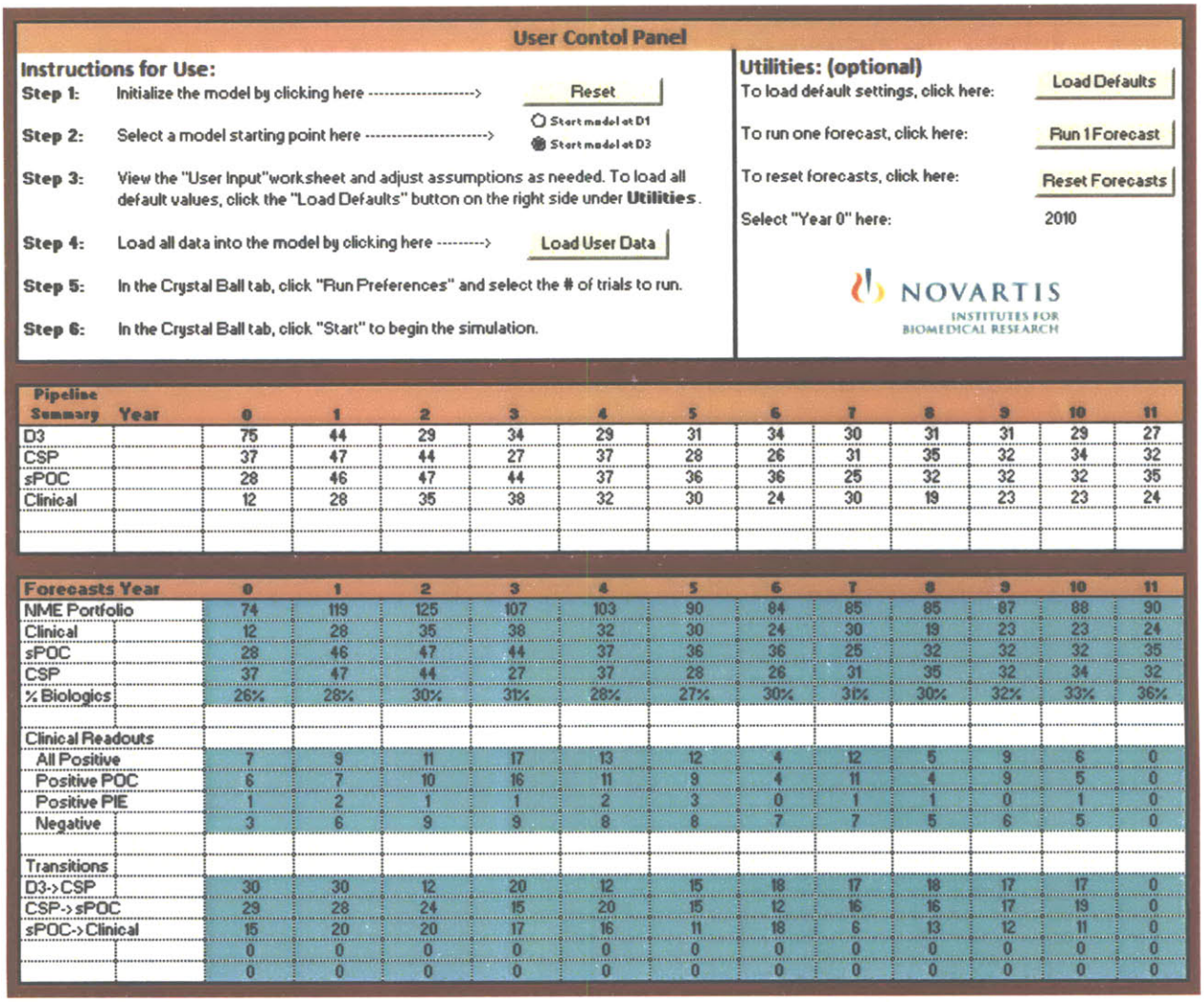
need for both default values and adjustment-capability defines an additional use case which must be easily accommodated in the model.

3.3 Model Formulation

This section describes the model formulation that implements the pipeline structure, dynamics, and user-specifications described above. Data visible in model screenshots does not represent the true NIBR dataset.

3.3.1 User Interface

The forecasting model resides in a Microsoft Excel workbook, and includes multiple worksheets. The user navigates the model from the *User Dashboard* worksheet, shown in Figure 13. This page includes integrated "Instructions for Use", which dictate the user's workflow, and optional "Utilities" to manipulate model settings. To run the model, the user sequentially follows the steps under "Instructions for Use". This worksheet also includes a summary tally of projects in the pipeline for a given simulation trial, as well as yearly forecast values for that trial.



User Contol Panel

Instructions for Use:

Step 1: Initialize the model by clicking here ----------> Reset

Step 2: Select a model starting point here ----------> ○ Start model at D1 ◉ Start model at D3

Step 3: View the "User Input" worksheet and adjust assumptions as needed. To load all default values, click the "Load Defaults" button on the right side under **Utilities**.

Step 4: Load all data into the model by clicking here ---------> Load User Data

Step 5: In the Crystal Ball tab, click "Run Preferences" and select the # of trials to run.

Step 6: In the Crystal Ball tab, click "Start" to begin the simulation.

Utilities: (optional)

To load default settings, click here: Load Defaults

To run one forecast, click here: Run 1 Forecast

To reset forecasts, click here: Reset Forecasts

Select "Year 0" here: 2010

NOVARTIS INSTITUTES FOR BIOMEDICAL RESEARCH

Pipeline Summary	Year	0	1	2	3	4	5	6	7	8	9	10	11
D3		75	44	29	34	29	31	34	30	31	31	29	27
CSP		37	47	44	27	37	28	26	31	35	32	34	32
sPOC		28	46	47	44	37	36	36	25	32	32	32	35
Clinical		12	28	35	38	32	30	24	30	19	23	23	24

Forecasts Year		0	1	2	3	4	5	6	7	8	9	10	11
NME Portfolio		74	119	125	107	103	90	84	85	85	87	88	90
Clinical		12	28	35	38	32	30	24	30	19	23	23	24
sPOC		28	46	47	44	37	36	36	25	32	32	32	35
CSP		37	47	44	27	37	28	26	31	35	32	34	32
% Biologics		26%	28%	30%	31%	28%	27%	30%	31%	30%	32%	33%	36%
Clinical Readouts													
All Positive		7	9	11	17	13	12	4	12	5	9	6	0
Positive POC		6	7	10	16	11	9	4	11	4	9	5	0
Positive PIE		1	2	1	1	2	3	0	1	1	0	1	0
Negative		3	6	9	9	8	8	7	7	5	6	5	0
Transitions													
D3->CSP		30	30	12	20	12	15	18	17	18	17	17	0
CSP-> sPOC		29	28	24	15	20	15	12	16	16	17	13	0
sPOC-> Clinical		15	20	20	17	16	11	18	6	13	12	11	0
		0	0	0	0	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0	0	0	0	0

Figure 13: Model UI - User Dashboard

The second worksheet in the model is the *User Input* page, shown in Figure 14. This page allows the user to manage the values of all input parameters used in the model—i.e., current projects, incoming projects, in-licensed projects, backup project prevalence and timing, PIE project prevalence and timing, phase duration, and phase transition rate. An identical worksheet, labeled *Default Values*, stores the default values determined as part of the historical data collection and analysis phase of the project. The user may replace all *User Input* values with the stored defaults by clicking the appropriate utility button on the *User Dashboard*. To avoid data corruption, the *Default Values* worksheet is password-protected such that only designated super-users can permanently change the defaults.

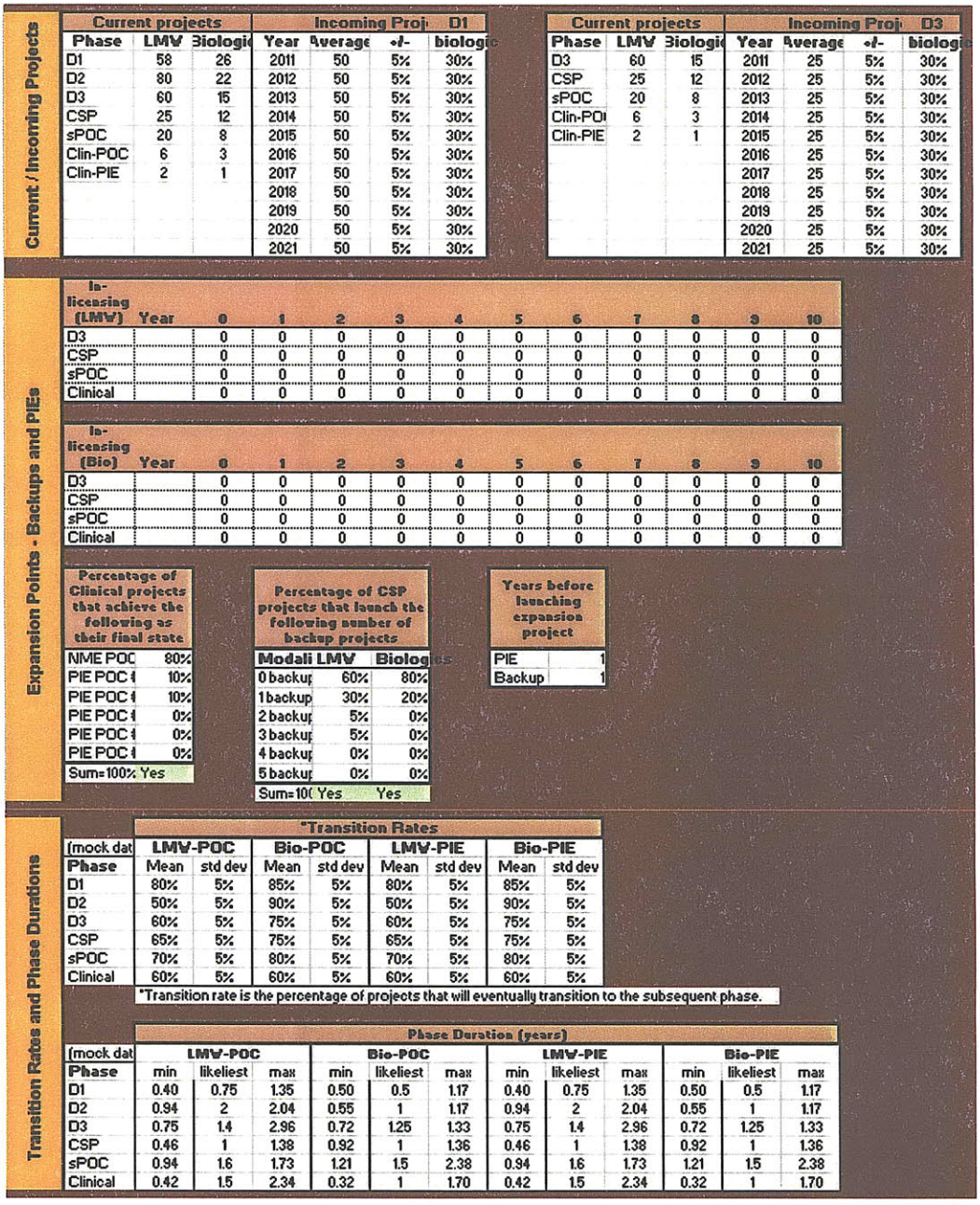


Figure 14: Model UI - User Input

The remaining four worksheets pertain to customized leadership reports. One worksheet, labeled *Leadership Report Data*, utilizes Crystal Ball worksheet functions to extract relevant statistics from the forecast cells on the *User Dashboard*. Three additional worksheets automatically display bar graphs following each simulation for portfolio size, readout, and transition metrics, respectively. These graphs synthesize the raw, per-year output data into portfolio trend information over time, as requested

by NIBR leadership to support decision-making. Sample graphs are provided in Figure 15, and the worksheets in their entirety appear in Appendix A.

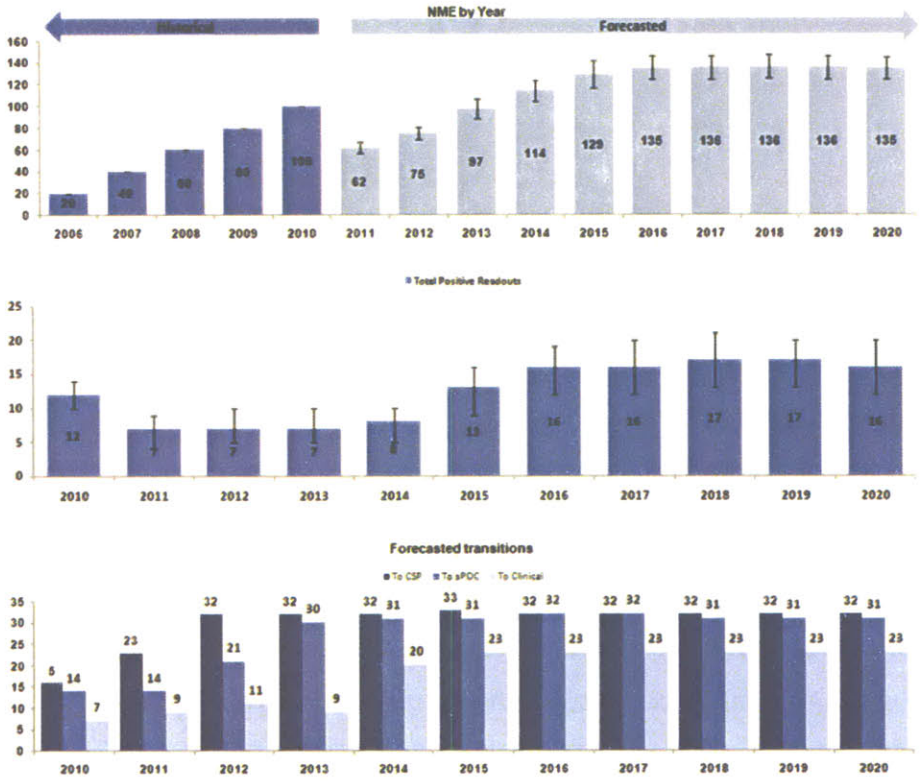


Figure 15: Sample Leadership Reports

In addition to the customized leadership reports, the end-user has access to reports created by the Crystal Ball software. For each simulation, Crystal Ball tracks statistics for every "forecast cell", shown in light blue in Figure 13. The benefit is that the user can investigate various data elements for any particular metric forecasted for any particular year. An example is the forecast chart in Figure 16, showing the frequency distribution for forecasted positive POC studies during year 3. Although the forecast charts provided a useful view of forecast variability, we believe that the customized reports provide a more useful view to portfolio managers. It is not clear that the predictive accuracy of the model justifies heavy emphasis on the percent likelihood of a particular forecast outcome. Rather, we believe the best use of the output data is to observe the mean and range of uncertainty for a given forecast, which is provided in the customized reports. Furthermore, the customized graphs allow trend comparison across multiple years, which the Crystal Ball forecast charts do not.

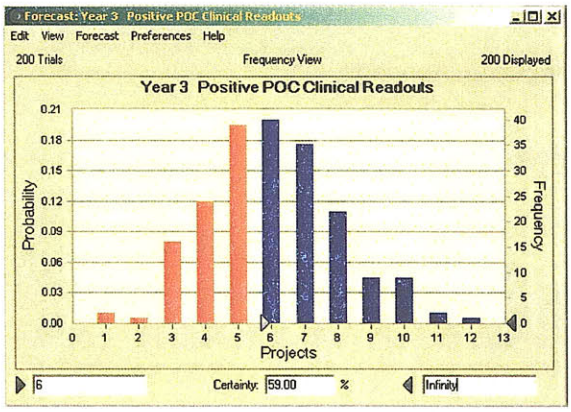


Figure 16: Sample Crystal Ball Forecast Chart

3.3.2 Model Architecture

Prior to discussing the logic that enables pipeline forecasting, it is useful to describe the high-level architecture of the model. The model is built using three key software components: Microsoft Excel, Visual Basic for Applications (VBA), and Oracle Crystal Ball. Whereas Excel provides the primary user interface as a platform for data input and output, VBA is used to process the data once it has been entered. Although VBA can interface directly with data residing in Excel worksheets, it is typically faster to perform calculations and other processing in memory using VBA. Thus, the general workflow of the model is that user-entered data is automatically loaded from an Excel worksheet into *variables* in memory, custom VBA scripts process the data in memory, and the results of the processing are loaded back into the Excel worksheets for output display. Data that appears in Excel as multiple, related cells are typically stored in an *array* in VBA, which is essentially a group of elements having a common name. For example, the values representing the number of current projects in each of six phases of R&D are stored in memory in a 6-element array called *current_projects*. This array terminology will be used extensively below as we describe the model.

Crystal Ball acts as an Excel add-on program, interfacing directly with the workbook in which the model resides. It is the simulation engine that manages uncertainty in the input variables by tracking simulation outcomes as inputs vary. A given simulation consists of a user-specified number of trials. In the model design, a trial is a single, 11-year forecast of the portfolio with a specific set of input assumptions; it is one possible outcome. Because many of the input assumptions are stochastic in nature, each trial will produce a different picture of what the pipeline will look like over the 11-year

period. Crystal Ball captures the results of each forecast metric for each trial, allowing us to create reports that comment on the simulation results—i.e., the aggregate of all trials—at once.

The model relies on Crystal Ball's ability to track forecasts for each simulation trial, but does not make use of its ability to alter input assumptions per trial. Rather, input assumptions are controlled directly through the model's custom VBA scripts. This choice stems from the model logic requiring two levels of uncertainty, which we refer to as the "inner and outer loops", as illustrated in Figure 17. The inner loop creates the single, 11-year forecast; in other words, it implements one simulation trial. As described in detail in Section 3.3.3 below, this forecast involves capturing project-to-project uncertainty in phase durations, as well as yearly variation in phase transition rates and other input parameters. While input assumptions are changed thousands of times during the inner loop, we are not interested in the forecast values resulting from each change (which Crystal Ball would record if driving the input assumption changes). Rather, we are interested only in the complete 11-year forecast obtained after this inner-loop has fully processed, illustrating the need for the outer loop. That is, the model, via Crystal Ball, records the forecast values *after each trial*, or 11-year forecast period. This captures not only the project-to-project and year-to-year variability in parameter values, but uncertainty across different possible 11-year forecast outcomes. If each inner-loop trial represents one possible picture of what the future will hold, the outer loop captures the values and uncertainty for all of these possible outcomes.

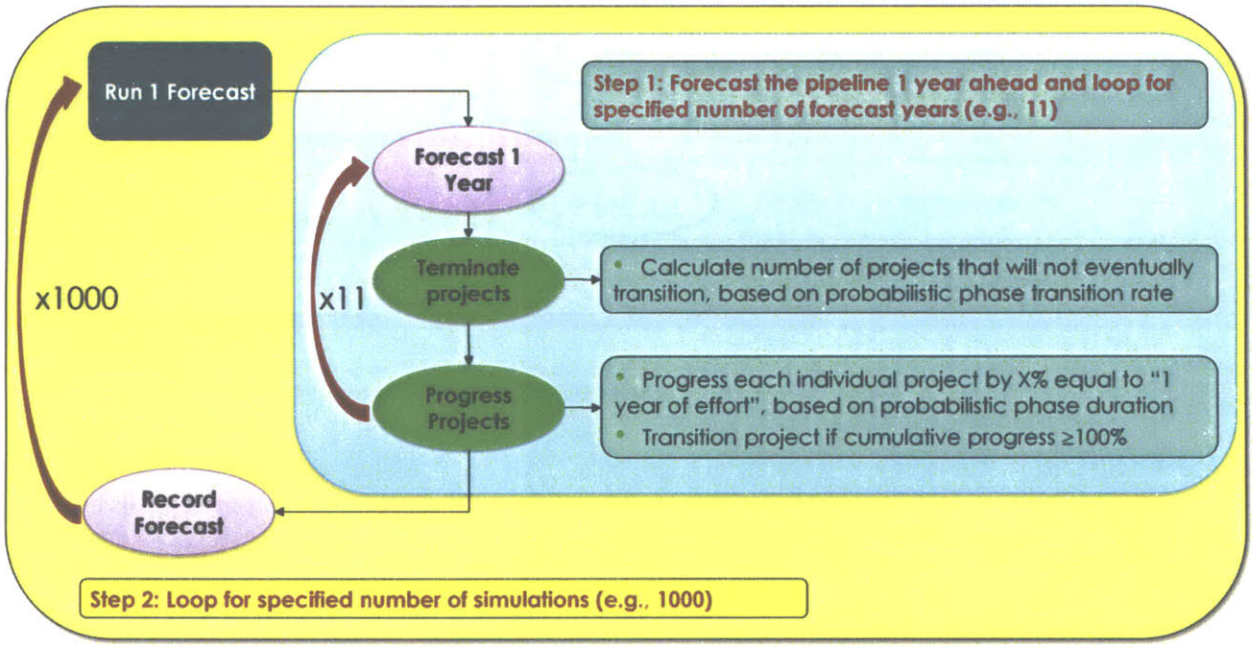


Figure 17: Two Sources of Uncertainty – "Inner" and "Outer" Loops

3.3.3 Model Logic

We describe the model logic chronologically as experienced by the user. We aim to include a sufficient level of detail to understand the general mechanics of the model and how projects progress through it. In addition, we include formulas and calculations that are relevant to understanding specific choices we made on how to best approximate the real-world pipeline. Equations used in this section are stylized to best describe their purpose and application. They do not necessarily include the exact syntax employed in the VBA code; rather, they illustrate the logic behind a given calculation. For additional information on the VBA logic, as well as the complete VBA code, please refer to Appendix A.

As stated, the user navigates the model from the *User Dashboard* worksheet. Excel Form Control buttons and the Crystal Ball user interface trigger the various VBA macros that drive the model. For reference, Figure 18 highlights these buttons. The user sequentially follow the steps under "Instructions for Use" to run the model.

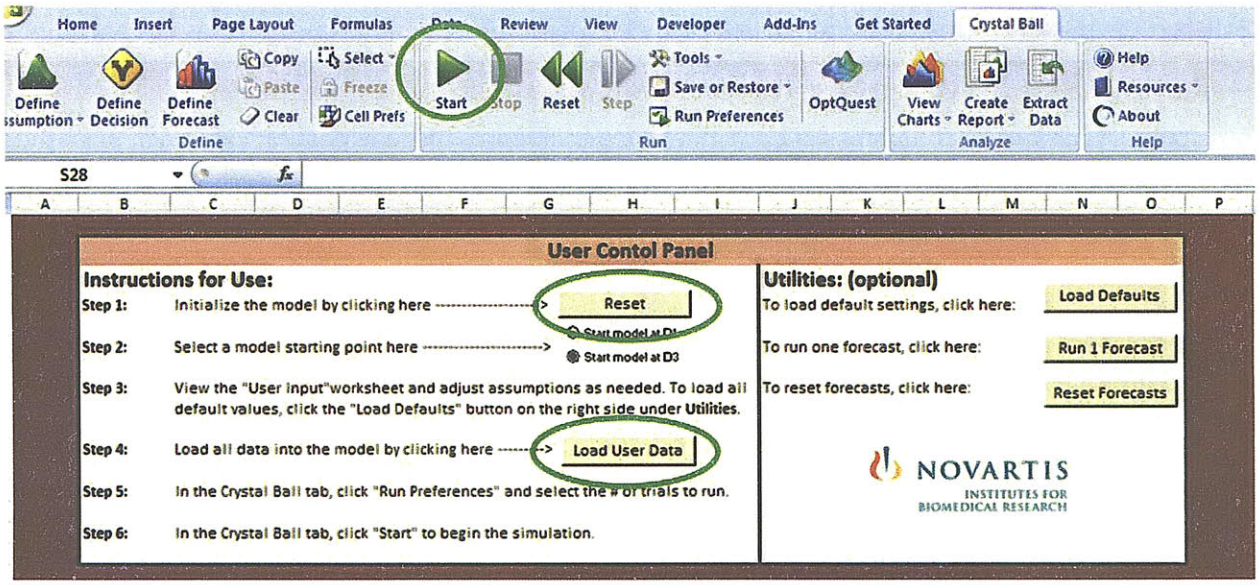


Figure 18: User Dashboard

3.3.3.1 Step 1

In Step 1, the user is instructed to initialize the model by clicking the *Reset* button. This button triggers VBA code that performs various initialization functions to prepare the model for a new simulation. For example, it clears data residing in worksheets and memory corresponding to a previous simulation.

3.3.3.2 Step 2

In Step 2, the user is instructed to select a model starting point by clicking one of two radio buttons that correspond to the desired starting phase. This button triggers VBA code that sets up the model to forecast from the chosen starting phase. This feature adds flexibility to the model by including or omitting TV and HF based on user preference.

3.3.3.3 Step 3

In Step 3, the user is instructed to view the *User Input* worksheet and adjust assumptions as needed. On this worksheet, the user has the option of manually changing values for:

- Current projects (LMW and biologics)
- Incoming projects estimate for each forecast year (average, confidence interval, and percent biologics)
- In-licensed projects estimate for each forecast year(per forecast year, LMW and biologics)
- PIE prevalence and launch delay
- Backup project prevalence and launch delay (LMW and biologics)
- Transition rates (per phase, for each type of project)
- Durations (per phase, for each type of project)

The user may also leave values as they were in the previous simulation, or may load all default values listed in the *Default Values* worksheet by clicking the *Load Defaults* button listed in the "Utilities" section in the *User Dashboard* worksheet. This button triggers VBA code that copies all values from the *Default Values* worksheet into the *User Input* worksheet.

3.3.3.4 Step 4

In Step 4, the user is instructed to load data into the model by clicking the *Load User Data* button. This button triggers VBA code that loads user-entered data into memory and prepares the model for simulation. Below, we highlight the key functions performed.

During this step, the model calculates an "actual" number of incoming LMW and biologics projects per forecast year based on the user-entered mean and standard deviation. The calculation is performed with the following equation:

$$actual\ incoming\ projects = NormInv\ (Rnd, mean\ incoming, standard\ deviation\ incoming)$$

Using Excel's *Rnd* function returns a random value between 0 and 1 for the probability in Excel's *NormInv* function. The result is a random draw from a normal distribution.(17)

The model also initializes the primary construct for managing project progression, which we refer to as the *pipeline_full* array. Conceptually, it is helpful to think of this array as having three-dimensions which together identify a particular project, as represented in Figure 19. Each project resides in a particular data element in the array, with its location identified according to three dimensions:

- R&D phase: TV, HF, LO, CSP, preclinical, or clinical
- Project type[4]: LMW-POC, Biologic-POC, LMW-PIE, Biologic-PIE
- Project number: an index identifying each project among all projects of a particular phase and type

The value of each data element represents the percent progress towards completion of the particular phase, ranging from 1% to 99%. As shown, the project showing "37" currently is in the LO phase, and 37% of work in the phase is complete.

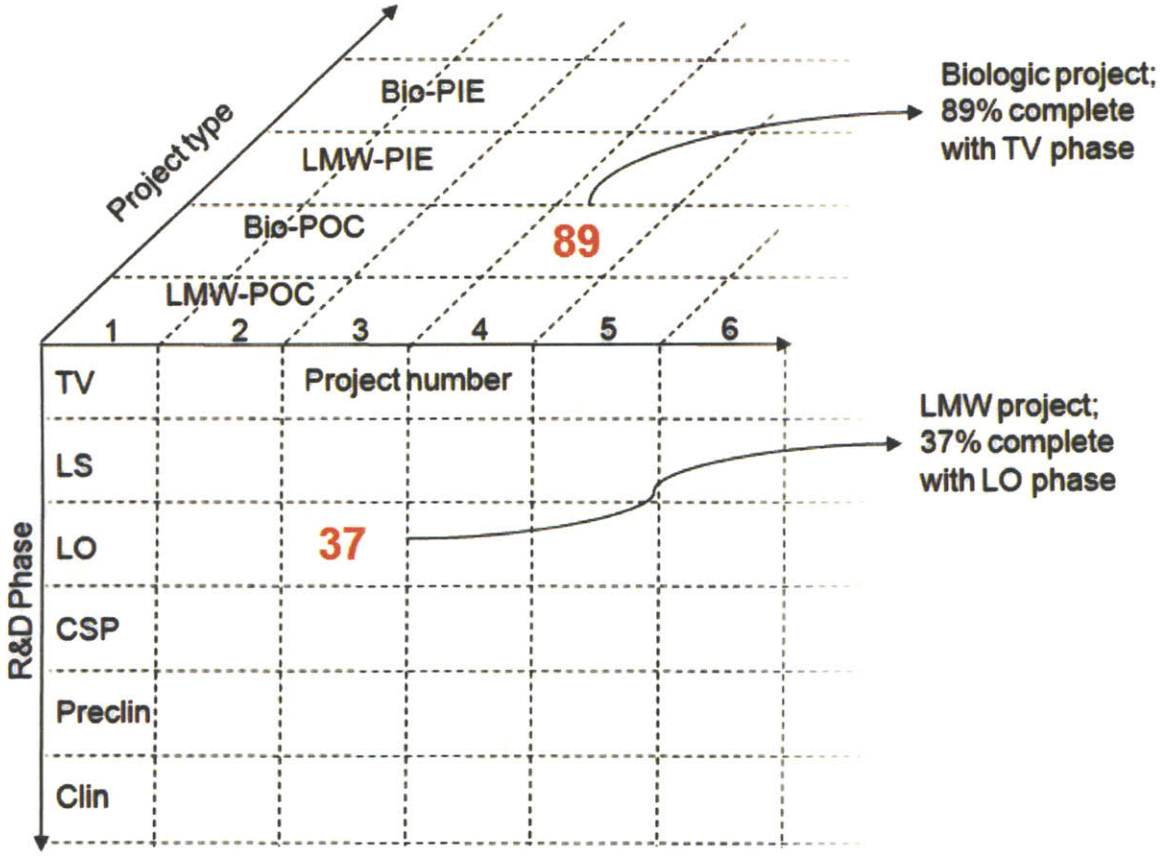


Figure 19: Conceptual Representation of Project Tracking in Model

[4] The distinction between POC and PIE is only applicable to the Clinical phase. Prior to this phase, all projects are designed as either LMW-POC or Bio-POC on the model.

At this step, the *pipeline_full* array is populated with the current project information. It would be cumbersome and likely inaccurate to collect estimates for current project progress for all portfolio projects prior to simulation. Instead, the model assumes random progress for all projects in the portfolio; we populate the *pipeline_full* array with the number of projects indicated by the *current projects* input data, assigning a random progress value between 1 and 99 to each project.

3.3.3.5 Step 5

In Step 5, the user is instructed to click "Run Preferences" on the Crystal Ball tab and select the number of trials to run. The model defaults to 200 trials, which typically completes one simulation in two to three minutes.

3.3.3.6 Step 6

In Step 6, the user is instructed to click "Start" on the Crystal Ball tab to start the simulation. This action launches the looping process shown in Figure 17, above, and progresses the current project portfolio to ultimately create successive 11-year forecasts. The mechanics of this project progression are described below.

As stated, project progression initializes with current projects populating the *pipeline_full* array. The model progresses projects in one-year increments until it has completed an 11-year forecast. The logic for each year is identical:

3.3.3.6.1 Project termination

As described in Section 3.2.2.2, historical transition rate data describes what percentage of projects that reach a particular phase will eventually transition to the subsequent phase. The model approximates this logic by counting how many projects transition into each phase in a given year, terminating a number of projects as dictated by each phase's transition rate, and continuing to progress only those projects that ultimately will transition to the next phase. Capturing the number of projects that enter a phase in a given year is sometime referred to as an entry-class approach. To start the model, we make a simplifying assumption that current projects enter their current R&D phases in Year 0, since this allows common logic to be used for all forecast years.

Terminated projects are not removed from the model immediately, since depending on the phase duration, we cannot assume that all projects that will eventually terminate do so within one year. As such, terminating these projects immediately would yield underestimated project tallies for each year, based only on the number of projects that will eventually transition. Instead, the model tracks

"projects pending termination" separately so that they can be included in interim project tallies prior to the time when they would actually terminate from the portfolio. We assume that projects terminate proportionally up to the average phase duration; for example, if 15 projects from a given entry class will terminate from a phase with a three-year average duration, the model will terminate 5 projects in each of the next three years.

3.3.3.6.2 Project progression

Wherever possible, we aim to capture the independence of each project to accurately reflect the high project-to-project variability in early research. As such, while projects are reported to the user as aggregated "buckets" of projects—e.g., 20 projects in preclinical phase in 2012—projects are progressed and transitioned independently. As described in Step 4 (Section 3.3.3.4), each project is represented separately in *pipeline_full* as an integer from 1 to 99, representing the percentage of progress towards completion of its current phase in a given forecast year. Those preclinical projects consist of 20 independent projects with different levels of completion at any given time. Furthermore, each project progresses without respect to the others; some may take approximately the average phase duration to complete the phase, while others may be particularly fast or slow projects, as is the case in the real-world.

To advance the portfolio by one year, the model begins with the first project—i.e., phase TV, project number one, project type LMW-POC—and progresses the project based on a series of tasks and calculations:

1. The model calculates how much work remains in the particular phase, *percent phase remaining*, by subtracting the current value in *pipeline_full* from 100. For example, if the current value is 37, then 63% remains to be completed.
2. The model calls a function to calculate a unique phase duration based on the BetaPert distribution parameters for the particular phase and project type. The function involves two key calculations.(18) First, the BetaPert parameters *min (a)*, *likeliest(M)*, and *max (b)* are converted into Beta distribution parameters α and β, then the Beta parameters are used to determine a random duration, as follows:

$$\alpha = \left(\frac{2(b + 4M - 5a)}{3(b - a)}\right) * \left(1 + 4\left(\frac{(M - a)(b - M)}{(b - a)^2}\right)\right)$$

$$\beta = \left(\frac{2(5b - 4M - a)}{3(b - a)}\right) * \left(1 + 4\left(\frac{(M - a)(b - M)}{(b - a)^2}\right)\right)$$

$$duration = BetaInv(Rnd, \alpha, \beta, min, max)$$

Similar to the calculation of an actual transition rate based on Excel's *NormInv* function, we calculate an actual duration using Excel's *Rnd* function and the *BetaInv* function. Specifically, α and β define the Beta distribution of durations for the given R&D phase, and the *min* and *max* parameters truncate the distribution to realistic values. The *Rnd* function returns a random value between 0 and 1 to serve as a probability parameter for the *BetaInv* function, effectively providing a random draw from the distribution.

3. Since the duration represents the number of years to accomplish the entire phase, we can use it as a proxy for the pace of work for the project; that is, for one year:

$$percent\ phase\ accomplished = 100 * \left(\frac{1}{duration}\right)$$

4. The model adds *percent phase accomplished* to the existing value for how much work has already been accomplished. If the sum is greater than or equal to 100, the project will transition in the given year; if the sum is less than 100, the project will not complete the phase in the given year. In the latter case, the model replaces the previous *pipeline_full* value (e.g., 37), with the new sum, and no further processing is required. In the former case, additional logic is needed to transition the project in the model.
5. In most cases, when a project transitions, some time remains in the year to accomplish work in the subsequent phase. The model calculates the *percent year used* based on how much work remained in the previous phase and how much work could have been accomplished in an entire year based on the duration. It can then repeat tasks 2 through 4 for the new phase, scaling the *percent phase accomplished* calculation by the portion of the year remaining. The model will repeat tasks 2 through 4 until the *percent phase accomplished* for a given phase is less than 100 when the entire year has been used.
6. For each phase transition, the model adds to various tallies. For forecast metrics, the model records the transition into each phase, as well as transitions out of the clinical phase, indicating a positive POC or PIE study. In addition, the model records each transition as an addition to the entry class for that year, which will be used in the subsequent year to determine the number of

terminated projects. Finally, for each transition into CSP or out of clinical, the model launches subroutines that determine whether a backup project or PIE project, respectively, will be created. These subroutines are discussed separately following this section.

Once a project completes its progression for a given year, the model moves sequentially onto the next project and repeats the same logic. Specifically, the model progresses through all LMW-POC projects in the TV phase, then moves on to the HF phase. Once the Clinical projects are completed, the model moves to the next project type, Bio-POC, and progresses through all phases. This progression continues until all projects in *pipeline_full* have been progressed by one year. The model then repeats the entire progression sequence until 11 years have been forecasted, yielding an 11-year portfolio forecast, equating to one simulation trial. Crystal Ball then records the forecast values, and launches the next simulation trial.

3.3.3.6.3 Expansion point subroutines

As described, expansion points are locations in the pipeline in which projects effectively multiply. Specifically, we focused on creation of backup projects at CSP and PIE projects at the clinical phase. To implement this feature in the model, we rely on what we refer to as "trigger points", or particular project transitions that trigger additional project creation by launching an appropriate subroutine. When a project transitions into the CSP phase, the model launches a *backup_calculator* routine, which determines how many backup projects (if any) to launch and when to launch them based on backup prevalence and timing assumptions. Likewise, when a clinical project of type LMW-POC or Bio-POC creates a positive readout, the model launches a *PIE_calculator* routine, which determines how many PIE projects (if any) to launch and when to launch them.[5]

The two expansion point subroutines employ similar logic; we will use backup projects as an example. Prevalence values such as those shown in Figure 20 define a discrete, custom distribution. For each CSP transition, the model draws from this distribution to determine how many backups to create, employing Excel's Rnd function to return a random integer between 1 and 100. If the number lies between 1 and 50, zero projects are created; if the number lies between 51 and 80, one project is created, and so on.

[5] Although trigger points simplify the timing of expansion point project launches in the model, project transitions are not truly the real-world trigger of new project creation. Rather, research management may launch backup project or PIE studies due to various project-specific reasons. Nonetheless, the trigger-point approach is consistent with our strategy to rely on historical data, since we can observe what percentage of project transitions were followed by new project creation, irrespective of the reason.

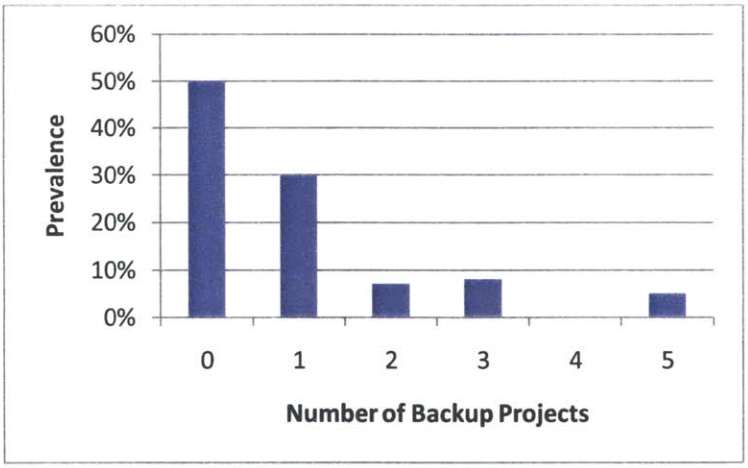


Figure 20: Backup Project Prevalence

Once the model calculated how many projects to launch, it must determine when to launch these projects in the simulation. To do so, the model determines how much of the year remains when the trigger point is reached, and adds the input parameter value that defines the time to launch a backup project. The sum indicates how many years ahead the backup project will launch; for example, a sum of 1.5 indicates that the project will launch halfway through the following forecast year. The partial year creates a challenge to implementation, since the main progression logic will progress all projects by an entire year. To accommodate this, the model handles the first year of each expansion project within the expansion point subroutines. In this case, the model will scale the progress calculation by the portion of the year that will remain in the launch year, or 0.5.

When one CSP project creates multiple backups, we make a simplifying assumption that the time-to-launch input parameter is applied in between the launch of each backup project. In other words, if the first backup will launch 1.5 years after the CSP transition, the second will launch 1.5 years after the first backup launches.

Each expansion project that will launch in the future is stored in an array. For each forecast year, the model loads projects designated to launch that year from the array into *pipeline_full*, after which they are treated like all other projects.

3.3.3.6.4 Leadership report creation

Once a simulation completes, the model automatically updates the custom leadership reports based on the simulation results. We accomplish this by using built-in Crystal Ball worksheet functions that allow extraction of simulation data. We are interested in reporting mean values of various portfolio

performance metrics, as well as the variability around the mean across the many trials. We extract mean, 10th percentile, and 90th percentile forecast values with the following worksheet functions, respectively:

$$= CB.GetForeStatFN(forecast\ cell\ reference, 2)$$
$$= CB.GetForePercentFN(forecast\ cell\ reference, 10)$$
$$= CB.GetForePercentFN(forecast\ cell\ reference, 90)$$

3.4 Model Utilization

This section discusses use of the model within the organization. We first provide detailed examples of how the model can be used to drive strategic portfolio decision-making. We then discuss efforts to aid adoption of the model into NIBR work practices.

3.4.1 Driving strategic decision-making

As stated, our ultimate goal is to deliver a pipeline model that can easily support real-time portfolio decision-making for NIBR's R&D managers. Therefore, to illustrate how the model can be used in this manner, we have created various hypothetical strategic examples. Although it is not feasible to demonstrate use of every model feature and portfolio lever, included examples cover a broad range of model functionality.

Below, we explore how the pipeline model can be used to support decision-making using two types of scenarios analysis. We first perform an input parameter sensitivity analysis, in which we vary the values of selected model inputs (e.g., current projects, incoming projects) to gauge the model's sensitivity to these values. We then walk through two hypothetical strategic scenarios. For each, we first observe the output of a baseline simulation run, and then test different managerial levers in the model to determine possible strategies to achieve a desired change in portfolio output. For simplicity, we will change one lever at a time for comparison to the baseline, but in practice, multiple levers can be changed concurrently to observe the interaction of these changes.

For this analysis, we start all simulations from the LO phase, which NIBR expects to use as its primary model starting point. All scenarios were simulated over 200 trials.

3.4.1.1 Input Parameter Sensitivity Analysis

Prior to attempting to alter the baseline portfolio, we first perform a sensitivity analysis to quantify the sensitivity of two key output metrics—NME portfolio size and positive clinical readouts—to changes to various input parameters. This analysis serves two purposes. First, it provides the user with a

sense of how heavily simulation results depend on the user's assumptions about portfolio status and performance. For example, the user assumes certain values for the future inflow of projects to LO, and model output will vary based on the values used. Second, the sensitivity analysis supports strategic portfolio analysis by elucidating the relationship between changes to the portfolio levers that managers can influence, and the expected impact on output metrics. NME portfolio size and positive clinical readout sensitivities are displayed in Figure 21 and Figure 22, respectively. Sensitivities are provided on a per-year basis, since a given change will impact each forecasted year differently. The sensitivities are color-coded such that larger percentage changes from the baseline are shaded darker than smaller changes. Note, however, that the magnitude of percentage change in output should be compared to the percentage change in input for that parameter to gauge the overall sensitivity.

Input Parameter	%Δ input from baseline	%Δ NME from baseline									
		2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Default Assumptions (Baseline)	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Current Projects - LO (LMW & Bio)	-30%	-19%	-20%	-19%	-16%	-9%	-7%	-7%	-7%	-6%	-6%
Current Projects - LO (LMW & Bio)	10%	6%	6%	6%	4%	2%	1%	1%	2%	1%	1%
Incoming Projects (no change to %bio)	-20%	-1%	-2%	-6%	-11%	-14%	-15%	-16%	-16%	-16%	-16%
Incoming Projects (no change to %bio)	20%	0%	2%	5%	9%	12%	14%	14%	14%	15%	15%
Durations - all phases	-10%	-6%	-9%	-14%	-18%	-16%	-15%	-15%	-15%	-14%	-15%
Durations - all phases	10%	7%	10%	13%	17%	19%	18%	18%	18%	18%	18%
Transition Rates - all phases	-10%	-4%	-10%	-13%	-13%	-11%	-11%	-11%	-11%	-12%	-11%
Transition Rates - all phases	5%	3%	6%	7%	6%	4%	5%	5%	5%	5%	5%
In-licensing - (2 projects/phase/yr)	N/A	4%	7%	11%	14%	19%	20%	20%	20%	19%	20%
PIE Prevelance - (% shift from 1 to 0)	-10%	0%	0%	-1%	-1%	0%	0%	-1%	0%	0%	0%
PIE Prevelance - (% shift from 0 to 1)	10%	-1%	-1%	-1%	-1%	0%	0%	0%	0%	0%	0%
Backup Prevelance - (% shift from 1 to 0)	-10%	-2%	-5%	-6%	-7%	-5%	-5%	-5%	-5%	-5%	-5%
Backup Prevelance - (% shift from 0 to 1)	10%	2%	4%	4%	5%	4%	4%	4%	4%	4%	4%

Figure 21: NME Portfolio Size Sensitivity to Changes in Input Parameter Values

Input Parameter	%Δ input from baseline	%Δ Readouts from baseline									
		2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Default Assumptions (Baseline)	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Current Projects - LO (LMW & Bio)	-30%	7%	0%	0%	-12%	-23%	-18%	-13%	-7%	0%	-7%	-7%
Current Projects - LO (LMW & Bio)	10%	7%	0%	0%	6%	9%	0%	0%	0%	7%	-7%	-7%
Incoming Projects (no change to %bio)	-20%	7%	0%	0%	6%	-5%	-6%	-13%	-13%	-14%	-20%	-27%
Incoming Projects (no change to %bio)	20%	0%	0%	7%	0%	0%	6%	13%	20%	14%	13%	13%
Durations - all phases	-10%	14%	9%	14%	18%	-9%	0%	-6%	0%	0%	-7%	0%
Durations - all phases	10%	0%	-9%	-7%	-12%	0%	-6%	0%	0%	7%	-7%	0%
Transition Rates - all phases	-10%	-7%	-9%	-21%	-29%	-32%	-35%	-38%	-33%	-36%	-40%	-33%
Transition Rates - all phases	5%	7%	9%	14%	24%	23%	18%	19%	20%	21%	20%	13%
In-licensing - (2 projects/phase/yr)	N/A	7%	9%	21%	18%	14%	24%	25%	33%	43%	27%	33%
PIE Prevelance - (% shift from 1 to 0)	-10%	7%	0%	0%	0%	-5%	-6%	-6%	-7%	0%	-20%	-7%
PIE Prevelance - (% shift from 0 to 1)	10%	7%	0%	7%	6%	0%	0%	6%	7%	7%	0%	0%
Backup Prevelance - (% shift from 1 to 0)	-10%	0%	0%	0%	0%	-9%	-6%	-6%	-7%	0%	-13%	-7%
Backup Prevelance - (% shift from 0 to 1)	10%	7%	0%	0%	0%	5%	0%	6%	7%	7%	0%	0%

Figure 22: Positive Readout Sensitivity to Changes in Input Parameter Values

The sensitivity analysis itself yields interesting and somewhat counterintuitive observations related to portfolio performance. Although we initially believed that input assumptions for the current

project count "play out" over roughly five years and no longer influence the portfolio, the results suggest differently. That is, we see that, assuming constant project inflow over the forecast period, the level of projects from 2015 forward is predicted to be approximately 7% lower when we decrease the LO current project count by 30%. After performing additional simulations to investigate this observation, we attribute the cause of this phenomenon to the role of expansion projects in the portfolio. That is, although LO projects themselves will progress through the NME portfolio within approximately 5 years, any backup or PIE projects launched will exist in the portfolio for significantly longer. In essence, early-stage projects create a multiplier effect.

We also observe that phase durations can significantly impact NME portfolio size, a relationship which is not necessarily intuitive. This result can be explained with Little's Law, which states that, in a steady-state system, the average number of items in the system is equal to the average dwell-time in the system times the average number of items arriving per unit time.(19) Therefore, assuming constant project inflow, an increase in phase duration requires an increase in the number of projects in the portfolio at steady-state.

A final observation is that small changes to the assumed phase transition rate dramatically influence both portfolio size and positive readout metrics. We attribute this result to two dynamics. First, transition rates are multiplicative; the chance of any one project reaching a late stage of R&D is the product of all the prior transition rates. To illustrate the effect, consider reducing transition rates for three consecutive phases from 50% to 40%. With a 50% rate, the percentage of projects that complete all phases equals "50% x 50% x 50%", or 12.5%. With a 40% rate, only 6.4% of projects complete all phases, yielding a 51% decrease in project throughput for a 10% drop in transition rate.

Second, transition rate changes would yield a similar multiplier effect as described above; any early-stage project that transitions to CSP or reaches a positive clinical readout has an opportunity to spawn an expansion project; conversely, any project terminated prior to these stages does not have this opportunity, thus, compounding the impact of any given transition rate change.

3.4.1.2 Strategy Scenario 1

For the first scenario, we select a hypothetical set of input parameter values, and achieve the baseline output reports shown in Figure 23. We observe an apparent portfolio gap; following strong growth in NME portfolio size and positive clinical readouts through 2010, these metrics fall significantly in 2011. This suggests a relatively weak current project portfolio. However, large numbers of expected

incoming projects build up the portfolio over the next five years, yielding eventual portfolio results that exceed 2010 levels.

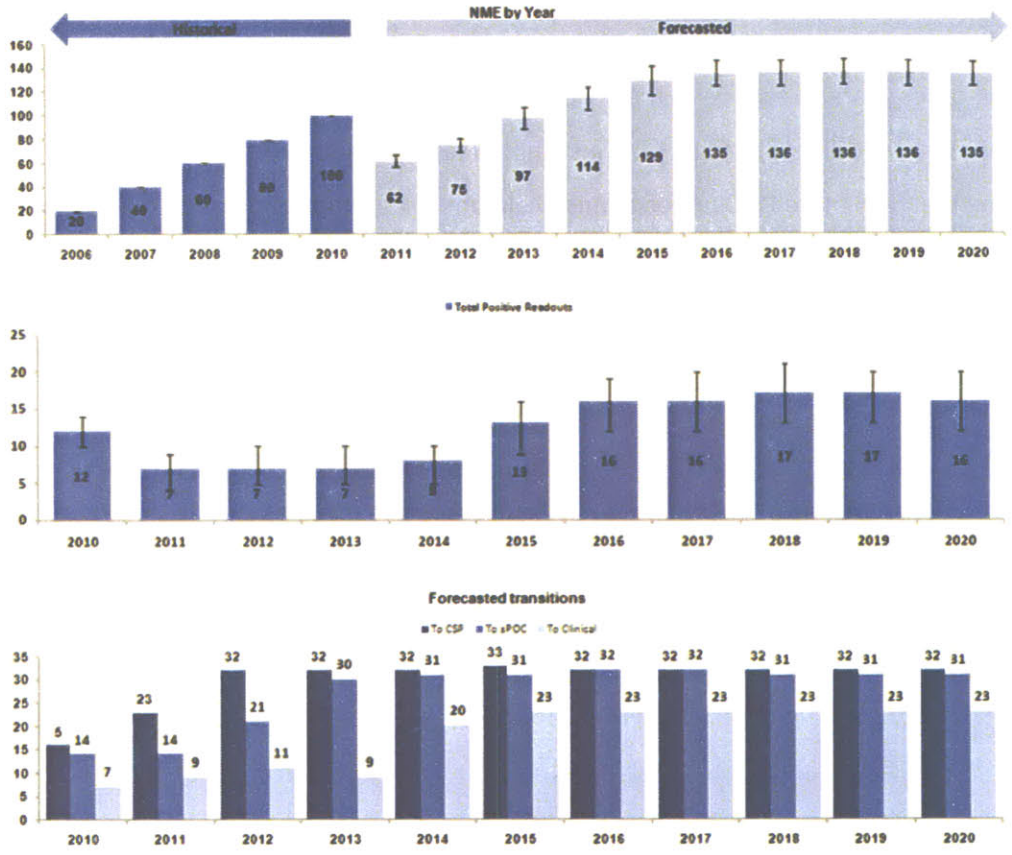


Figure 23: Scenario 1 - Baseline Output

Upon viewing this forecast, we will assume that NIBR aims to "fill in" its portfolio gap by increasing the positive-readout output from approximately seven to ten projects for 2011 to 2014. As such, we will independently utilize various portfolio levers within the model to achieve this goal.

Note that we could perform a similar exercise focusing on affecting the NME portfolio size metric; however, as observed during the sensitivity analysis, NME portfolio size is not directly correlated to positive clinical readouts. As shown, less efficient practices that might lead to long phase durations will increase the NME portfolio size due to slower throughput, but will not increase the numbers of positive readouts.

3.4.1.2.1 Lever #1: Increasing PIE project prevalence

Increasing prevalence of PIE studies will increase the proportion of clinical phase projects that "spin-off" additional PIE studies. From the sensitivity analysis, we observe that a 10% shift from "0 PIEs launched" to "1 PIE launched" yielded a 6-7% increase in positive-readout output for some years. It had no impact on other years, likely due to the time delay in PIE launch. Seeking roughly a 40% increase in positive-readouts, we will assume that we must shift at least 50% from 0 to 1 PIE launched. In practice, we are saying that 50% more POC projects must yield one PIE project. Simulating this scenario, we observe the outputs shown in Figure 24.

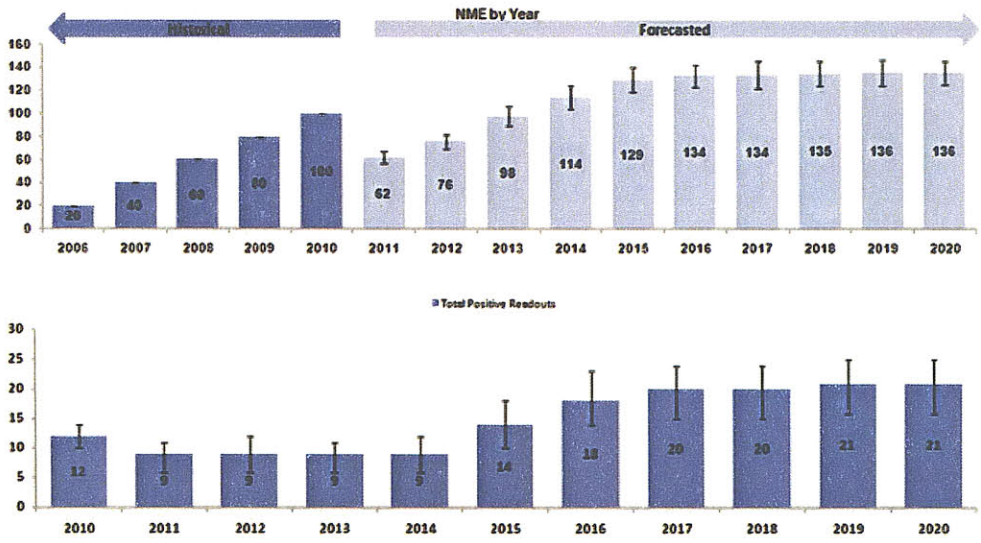


Figure 24: Scenario 1 - PIE projects lever

Even with a fairly significant change in practice—i.e., launching PIE studies for 50% more POC studies—we only observe a two-project increase in positive readouts for the target years. Although we can attempt to increase PIE prevalence further, we question whether such a dramatic change would be feasible in real-world research practices. That is, decisions to launch PIE studies are not made arbitrarily, but are based on hypothesized potential of a therapy for treating a different disease. Therefore, we look to other pipeline levers for achieving the desired portfolio change.

3.4.1.2.2 Lever #2: In-licensing

An additional option available to NIBR is to seek strategic partnerships that fill-in gaps in the portfolio. In-licensing is one such mechanism that can be employed in a variety of forms. Although the sensitivity analysis assumed yearly in-licensing for simplicity, NIBR can apply targeted in-licensing opportunities that address specific needs.

Note that there are practical challenges to employing the in-licensing lever in practice. At the clinical phase, in-licensing decisions require careful evaluation of compound value on a project-specific, scientific basis, and suitable projects may not be available. Nonetheless, this exercise shows the potential impact of in-licensing as a portfolio lever.

For this scenario, we would like to explore how in-licensing can be used to increase the number of positive readouts between 2011 and 2014. Given the complex phase dynamics caused by probabilistic phase durations and PIE project creation, we will forego use of the sensitivity analysis, and rather, experiment with multiple in-licensing values by trial-and-error.

We consider in-licensing three LMW projects per year at the preclinical phase for 2010 to 2014, and three LMW projects at the clinical phase for 2010 to 2015. Simulating this scenario, we observe the outputs shown in Figure 25. We then repeat this scenario assuming in-licensing six LMW project per year, and observe the outputs shown in Figure 26.

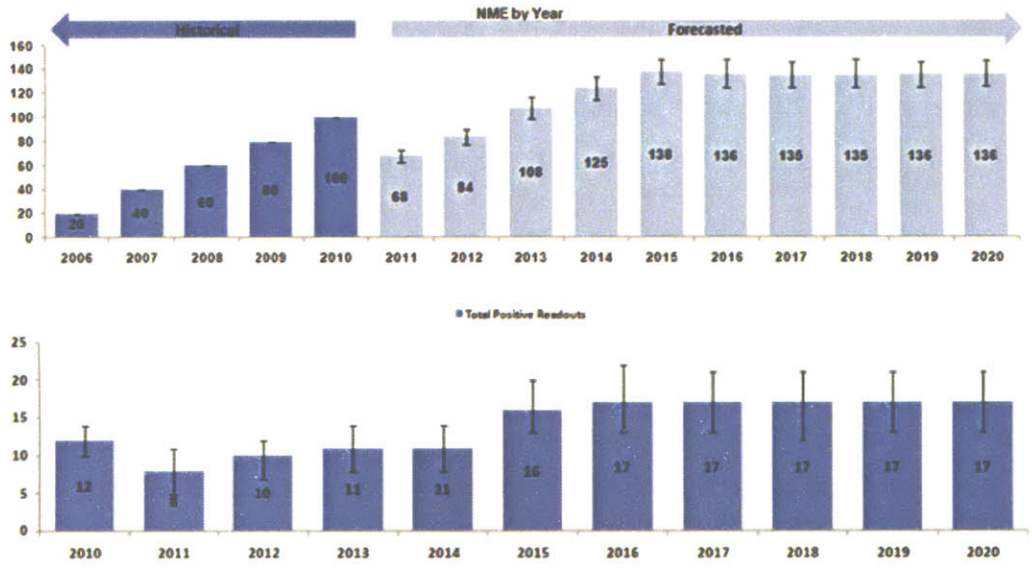


Figure 25: Scenario 1 - In-licensing lever (3 projects)

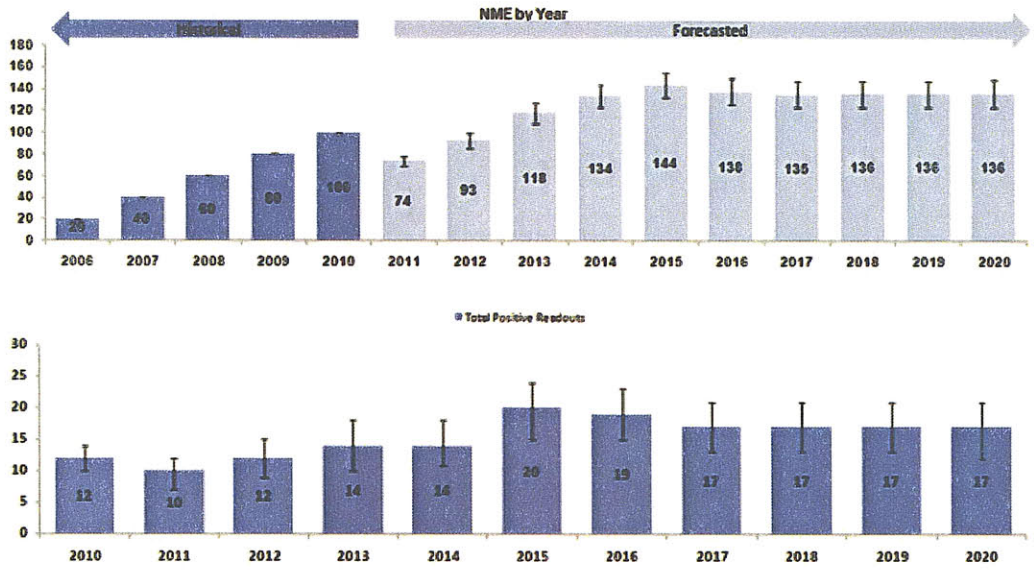


Figure 26: Scenario 1 - In-licensing lever (6 projects)

Observing the respective outputs, we see that the in-licensing lever is quite effective at closing the near-term portfolio gap. In-licensing three projects per year at the preclinical and clinical phases achieved the goal in all but 2011, while having little impact after 2015, as desired. In-licensing six projects yielded readout outputs achieving the 2011 goal, but exceeded the stated goal for all other years.

3.4.1.3 Strategy Scenario #2

For the second scenario, we select a different hypothetical set of input parameter values, and achieve the baseline output reports shown in Figure 27. We observe that the current portfolio appears very strong, yielding a large NME portfolio and high positive-readout output over the next three years. However, this growth is not sustained, and by 2015, both metrics have declined well-below 2010 levels. This decrease suggests that, based on the assumptions for incoming projects, project inflow to LO cannot sustain the near-term NME portfolio growth.

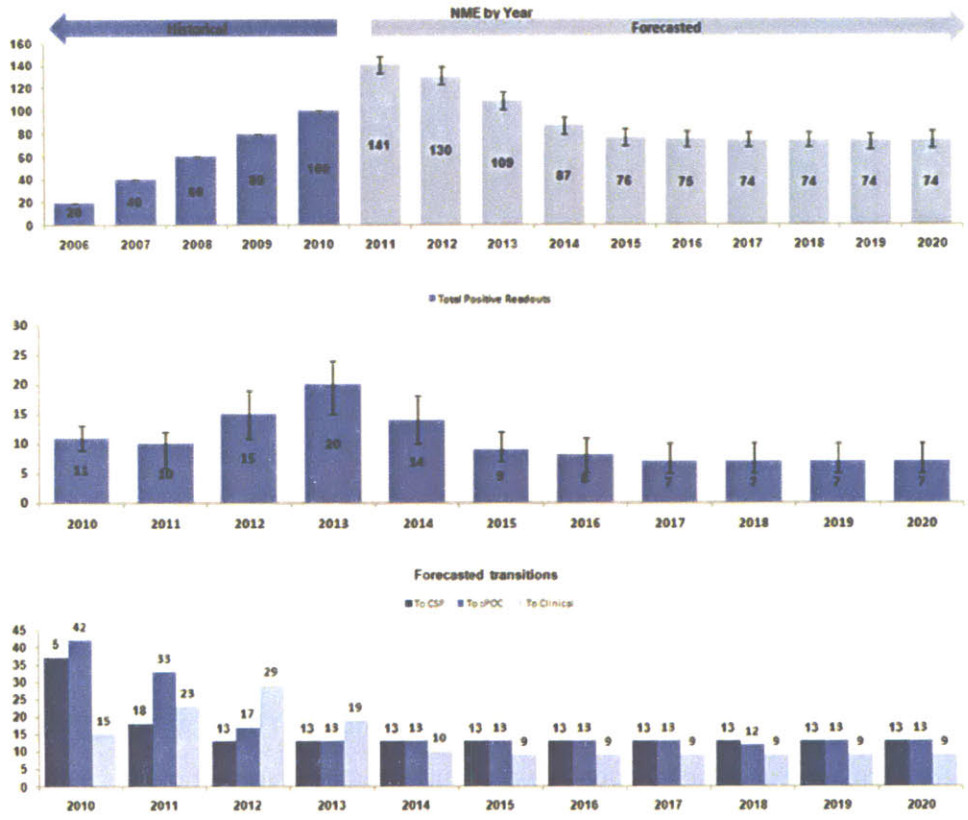


Figure 27: Scenario 2 - Baseline output

Upon viewing this forecast, we will assume that NIBR aims to achieve a steady output of approximately 10 positive readouts per year after 2015. (Readouts between 2010 and 2014 will be largely driven by the strength of the current portfolio.) As such, we will independently utilize various portfolio levers within the model to achieve this goal.

3.4.1.3.1 Lever #1: Increasing resources to increase incoming projects

One strategic lever available to management is to increase staffing at upstream R&D phases to increase the total number of projects that feed the portion of the pipeline we are concerned with. Yearly incoming projects to LO impact both short- and long-term portfolio size and throughput, and therefore, are potential levers for impacting the steady-state project count and positive-readout output. The sensitivity analysis revealed a 20% change in yearly incoming projects to yield a 13-20% change in positive-readout output from 2016 forward. Seeking to increase steady-state positive-readouts from 7 to 10, or 43%, we raise incoming projects by a similar percentage. Simulating this scenario, we observe the outputs shown in Figure 28.

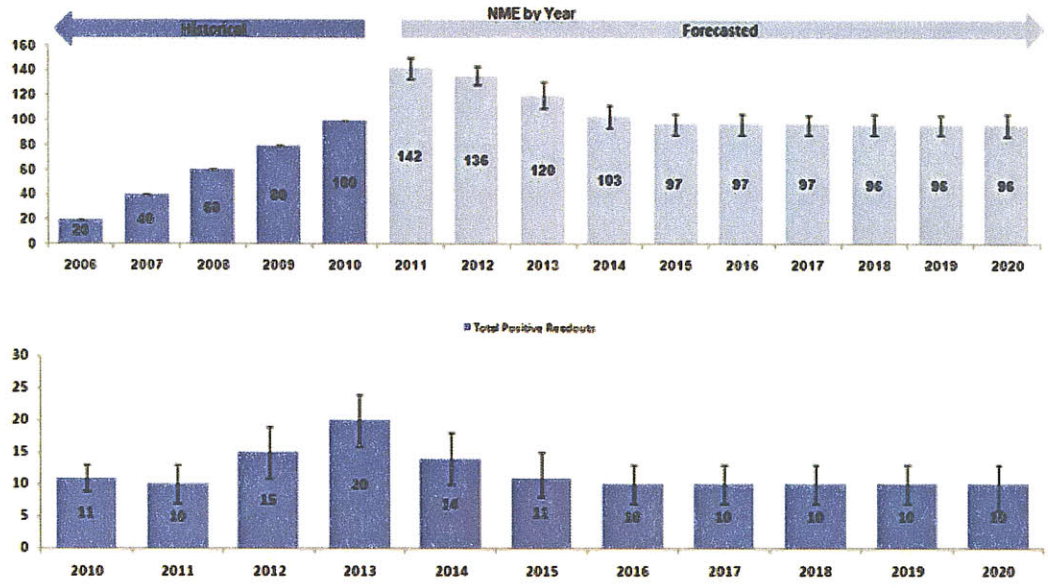


Figure 28: Scenario 2 - Incoming projects lever

Observing this result, we find changes to the project inflow to be an effective lever for influencing long-term portfolio size and positive-readout output. The impact of the change in the near-term is minimal, since incoming projects take multiple years to cascade through the NME portfolio and affect positive-readout output.

3.4.1.3.2 Lever #2: Increasing backup project prevalence

Increasing prevalence of backup projects will increase the proportion of CSP projects that "spin-off" additional projects. From the sensitivity analysis, we observe that a 10% shift from "0 backups launched" to "1 backup launched" yielded a 6-7% increase in positive-readout output for some years. It had no impact on other years, likely due to the time delay in PIE launch. Seeking a 43% increase in positive-readouts, we will assume that we must shift at least 50% from 0 to 1 PIE launched. In practice, we are saying that 50% more CSP projects must yield a backup project. Simulating this scenario, we observe the outputs shown in Figure 29

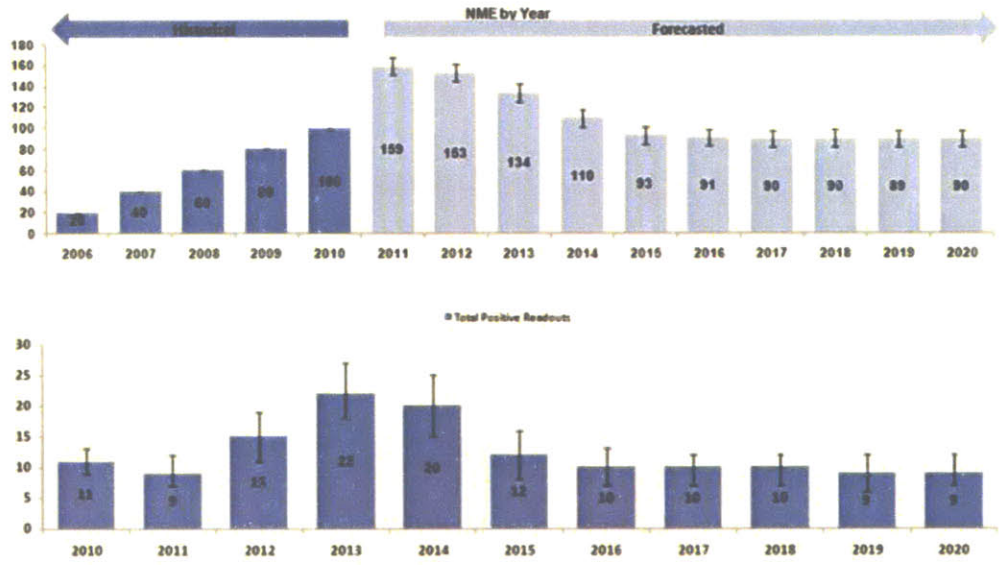


Figure 29: Scenario 2 – Backup projects lever

Observing the result, we find the change in backup project prevalence to be effective in achieving the desired output. However, a significant change to research practice—i.e., launching 50% more backup projects—would be required to achieve this outcome. In practice, managers would need to consider any limits to the feasibility of raising backup project prevalence compared to altering other available portfolio levers.

3.4.1.3.3 Lever #3: In-licensing

As described for Scenario 1, in-licensing of projects is another lever available to NIBR to fill gaps in the portfolio. For this scenario, we would like to explore how in-licensing can be used to increase the number of positive readouts after 2015. Given the complex phase dynamics caused by probabilistic phase durations and PIE project creation, we will forego use of the sensitivity analysis, and rather, experiment with multiple in-licensing values by trial-and-error. We consider in-licensing five LMW projects per year at the preclinical and clinical phases beginning in 2015. Simulating this scenario, we observe the outputs shown in Figure 30.

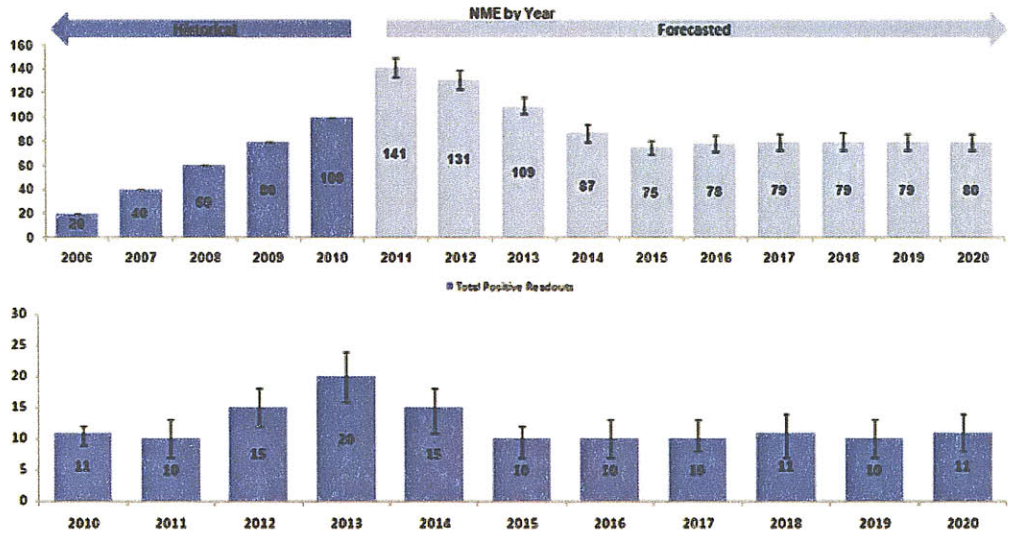


Figure 30: Scenario 2 - In-licensing lever

Observing the respective outputs, we see that the in-licensing lever is quite effective at impacting the future portfolio. In-licensing five projects per year at the preclinical and clinical phases achieved the goal of 10 positive readouts from 2015 forward.

3.4.2 Organizational implementation

Although analysis of NIBR's portfolio management processes was outside the scope of the project, we took various steps to aid seamless integration into NIBR's processes and encourage model usage and further model development over time. The model itself was designed for user-friendliness to provide a shallow learning curve, even with infrequent use. For example, "Instructions for Use" are integrated into the model, and actionable output reports are automatically generated and displayed for each simulation. In addition, as part of project transitioning, we provided model orientation and training to all members of the Portfolio Management Group (PMG), as well as technical training to an intended "model owner" within the PMG. Finally, we provide the organization with discussion of lessons learned through the modeling effort, recommendations for how to most effectively implement and utilize the tool, and recommendations for future model improvements via this thesis document.

4 Model Validation

4.1 Methodology

Given that the model creates a multi-year forecast of the research portfolio, the model output could not be validated by forecasting from today's portfolio. Instead, we validated the model using historical data. That is, we asked the question: Based on the state of the portfolio in the past, how well does the model predict today's portfolio? To answer this question, we created historical portfolio snapshots—i.e., tallies of the number of projects in each phase of research at various moments of time in the past—to serve as "current states". Discussions with the portfolio management team suggest that historical data quality becomes less reliable prior to 2004. Thus, we chose 2004 as the lower date limit for creating portfolio snapshots. (Note that this decision is consistent with the limit used when analyzing historical data to include in the model itself.) Using available historical project data, we constructed three validation scenarios based on three distinct "current states"— the portfolio as of 1/1/2004, 1/1/2006, and 1/1/2008.

To create estimates for yearly incoming projects, we determined the number of new projects that entered phases TV and LO each year over the 2004-2009 validation period, if available. These quantities were gathered through discussions with various research personnel within NIBR. For 2009 forward, we used stated or estimated project quantity goals. Likewise, in-licensing inputs for the validation period were based on research management estimates.

For all validation scenarios, the model was run with 200 simulation trials. Transition rates, durations, PIE prevalence, and backup prevalence were set to the model defaults, determined through historical data analysis as part of the modeling effort. Each validation scenario was first run with a model starting point of TV, and then with a starting point of LO, allowing observation of accuracy with each starting point option.

For each validation scenario we ran the model and compared the output of predicted portfolio metrics with the actual state of the portfolio on 1/1/2010. Results are reported as percent error in Figure 31. Note that the relevant portfolio metrics are the 2010 portfolio sizes—i.e., the state of the portfolio on 1/1/2010, and 2009 readouts and transitions—i.e., the number of readouts or transitions that occurred during 2009. Actual metrics for portfolio size, positive readouts, and transitions were

determined using internal portfolio status reports and through discussion with the portfolio management team.

4.2 Limitations of our approach

Although we believe the validation methodology represents the best feasible option, we recognize a few limitations. First, just as the accuracy of historical data on which the model is built is questionable, the accuracy of data used to determine the portfolio snapshots is also questionable. That is, inaccurate tallies could result from inaccurate or missing project milestone dates, inaccurate project status listings (e.g., active versus terminated)[6], changes in R&D phase definition over time, and ambiguous project data.

In addition, the model and validation approach do not account for some other organizational dynamics over the past six years. For example, NIBR grew rapidly since the start of the data collection/validation period. As such, we expect that resources were gradually added, so presumably, project capacity grew as well. While some of this growth would presumably have been captured in data for incoming TV and LO projects, portfolio growth due to in-licensing, company acquisition, or other partnerships would not. Likewise, interviews revealed a potential "bolus effect" at the end of each year in which projects are pushed towards milestone completion to meet particular goals for the year. Such dynamics are not captured in the model or validation methodology.

4.3 Validation Results and Conclusions

Figure 31 describes the error observed in output metrics for three validation scenarios, representing each of three portfolio snapshot dates as current state inputs.

[6] As identified during interviews, it is not uncommon for early-stage projects (e.g., TV and HF) to linger in the pipeline; that is, projects may remain in the pipeline without much or any ongoing work while resources tend to other projects. As long as the projects have not been officially terminated, they will appear in the project count despite resources no longer being applied. The current model is not designed to handle such dormant projects.

Validation Scenario:	1		2		3	
Snapshot date:	1/1/2004		1/1/2006		1/1/2008	
Starting Point:	TV	LO	TV	LO	TV	LO
			% error			
2010 NME portfolio						
NME	-21.7%	-40.2%	-3.3%	-28.3%	38.0%	7.6%
Clinical	-26.7%	-50.0%	-6.7%	-30.0%	13.3%	-3.3%
Preclinical	12.0%	-20.0%	32.0%	-8.0%	72.0%	44.0%
CSP	-29.7%	-32.4%	-10.8%	-24.3%	37.8%	0.0%
2009 Readouts						
Total Positive	-22.2%	-22.2%	0.0%	0.0%	33.3%	33.3%
POC	25.0%	25.0%	75.0%	75.0%	175.0%	175.0%
PIE	-60.0%	-60.0%	-60.0%	-60.0%	-80.0%	-100.0%
2009 Transitions						
To Clinical	-33.3%	-53.3%	-6.7%	-33.3%	20.0%	20.0%
To Preclinical	25.0%	8.3%	100.0%	16.7%	125.0%	116.7%
To CSP	6.7%	6.7%	26.7%	20.0%	93.3%	46.7%

Figure 31: Percent error for model validation scenarios

4.3.1 Validation Scenario #1

With a snapshot date of 1/1/2004, we generally found the model to underestimate the 2009/2010 portfolio metrics, though the deviation often fell within the simulation's 80% confidence interval. This result was consistent for model starting points of TV and LO.

The model's under-estimation for 2010 NME portfolio size fell below the simulation's 80% confidence interval for both TV and LO model starting points. Possible explanations for this underestimation include:

- Not accounting for NIBR's significant growth in staffing (and therefore, projects) between 2004 and 2009 in the model
- Not effectively accounting for projects from external sources (e.g., in-licensed) from 2004-2009
- Inaccurate early-phase project tallies for 1/1/2004

4.3.2 Validation Scenario #2

With a snapshot date of 1/1/2006, we found mixed validation results. In the case of 2010 NME portfolio size, we found the model with a TV starting point to predict the actual NME portfolio size within 3.3%. However, with a LO starting point, the model underestimated the portfolio size by approximately 28%. From a numerical standpoint, the differences in model output with the two starting points can be traced back to a seemingly large number of HF projects tallied in the historical data query for 1/1/2006. That is, by 2010, this relatively large number of HF projects would flow through the portfolio to create a relatively high tally for NME portfolio size. However, the actual number of LO projects that reportedly began in 2007 and 2008 does not reflect such a large HF project quantity in 2006. Thus, it is difficult to draw definitive conclusions as to the accuracy of the model's NME portfolio size prediction under this scenario. If we assume a smaller 2006 HF portfolio size, we would observe consistent underestimation of this metric for the two starting points.

We found the model to exactly predict the total number of positive clinical readouts in 2009 (0% error). The actual ratio of POC to PIE projects was also within the 80% confidence interval of the simulation.

4.3.3 Validation Scenario #3

With a snapshot date of 1/1/2008, we found mixed validation results. In the case of 2010 NME portfolio size, we found the model with a TV starting point to overestimate the actual NME portfolio size by 38%. However, with a LO starting point, the model predicted the portfolio size within 7.8%. From a numerical standpoint, the differences in model output with the two starting points can be traced back to a seemingly large number of HF projects tallied in the historical data query for 1/1/2008. That is, by 2010, this relatively large number of HF projects would flow through the portfolio to create a relatively high tally for NME portfolio size. This effect can be seen in the relatively high predictions for CSP and preclinical portfolio size, in particular. Thus, it is difficult to draw definitive conclusions as to the accuracy of the model's NME portfolio size prediction under this scenario. If we assume a smaller 2008 HF portfolio size, we would observe consistent, accurate predictions of this metric for the two starting points.

We found the model to overestimate the total number of positive clinical readouts in 2009 by 33%; however, the actual number of readouts is within the 80% confidence interval of the simulation.

We also noted that the overestimation is possibly driven by a relatively high number of preclinical projects tallied from the historical data query in 1/1/2008.

4.3.4 Observations across validation scenarios

As identified in the *Limitations* section, above, organizational growth from 2004 to 2009 is not taken into account in the model and could in fact impact the validation results. Given that we observed an underestimation of the 2009 NME portfolio size for Validation Scenarios #1 and #2, one reasonable hypothesis is that additional projects could have been inserted into the portfolio via in-licensing or acquisition/partnership during this period of growth. This hypothesis is bolstered by the fact that prediction of this metric was fairly accurate in Validation Scenario #3, since NIBR's growth rate has reportedly decreased more recently. If organizational growth is in fact a significant source of the observed underestimation, we would expect such error to be greatly reduced when the model is used for forecasting from today's portfolio, since growth is now significantly less than during the validation period. As an option, the user can also add projects to the "In-licensing" section of the model to account for expected growth during the forecast period.

In all scenarios, we found the model's forecasts of 2009 transitions to be somewhat inaccurate. That said, we can make some reasonable hypotheses as to the sources of the error. First, our concerns regarding inaccurate HF project tallies (discussed above) would impact the transition projections as greatly as they impact the NME portfolio size predictions. Second, not surprisingly, the model was generally more accurate at forecasting transitions aggregated across multiple phases than it was at predicting transitions into any one phase. Many organizational dynamics can impact the number of projects transitioning in any one year (e.g., the "bolus effect" discussed in the *Limitations* section). However, summing across multiple phases, we get a more generalized sense of the numbers of transitions taking place that year.

5 Conclusions and Recommendations

5.1 Conclusions on the modeling process

The modeling process has successfully delivered a pipeline model that outputs probabilistic forecasts of key portfolio metrics, including portfolio size, positive clinical readouts, and research phase transitions. The model utilizes probability distributions for phase durations and transition rates, derived from historical data, and Monte Carlo simulation to capture uncertainty in these input parameters. The model also differentiates between project types (e.g., LMW versus biologics) and accounts for expansion projects added to the portfolio (e.g., in-licensed projects, backup projects). Below, we evaluate the project's effectiveness in the context of accuracy, usability, and organizational implementation.

5.1.1 Model accuracy

Validation of the model against historical data shows good predictability for aggregate forecasts, but weaker predictability for finer portfolio metrics. Observed error is likely attributable to questionable historical data, as detailed in Section 4. The model can only be as accurate as the historical data drives it, and any future efforts to improve the accuracy of captured project data will also improve the accuracy of portfolio forecasting.

In addition, although we selected a reasonable approach to model validation, a reality is that it is quite challenging to prospectively evaluate accuracy of such a model. Although the simulation predicts the likelihood of various portfolio outcomes, the outcome that manifests itself over time could in fact be a less likely one. Thus, our confidence in the accuracy of the model largely stems not from the validation results, but from the rationale of the modeling approach.

5.1.2 Model usability

An iterative design approach incorporated frequent user feedback and yielded a user-friendly model structure. Research management has shown a positive initial response to the model, and additional efforts are currently underway to gain further leadership buy-in. Leadership noted, in particular, the intuitiveness of the built-in Instructions For Use, which walk the user through the modeling steps. Managers also commented on the fast simulation time, which allows real-time scenario analysis to support strategic decision-making.

5.1.3 Organizational issues

Due to project timing, we were unable to observe and evaluate implementation of the model, as the finalized model was delivered shortly before on-site work was completed. Nonetheless, discussions with NIBR leadership suggest that the model will be utilized for upcoming planning processes. Given that processes for model use, such as updating input parameter data, have not yet been fully established, we recommend that NIBR consider formal efforts to institutionalize the use of modeling within the Portfolio Management Group.

5.2 Recommendations

5.2.1 General recommendations

Broadly, the research proved successful at creating a prototype model for portfolio forecasting in early drug discovery. Research organizations seeking to better understand the development of the research portfolio can follow a similar modeling approach as presented in this thesis, considering applicability of the approach to many common problems, such as handling of stochastic input parameters through simulation, and analysis of historical project data.

For NIBR, we recommend additional internal efforts to integrate the model into the portfolio management process. This may include defining appropriate circumstances to incorporate the model into planning efforts (e.g., during periodic portfolio reviews), as well as defining processes to easily update input parameter assumptions prior to model use. The Portfolio Management Group should continue to build and maintain its modeling competencies to get the most out of the existing model and leverage opportunities for its future expansion.

5.2.2 Opportunities for model expansion

5.2.2.1 Feature addition

A valuable addition to the portfolio model could be resource-dependency. Although difficult to quantify given the staffing approach in early research, there undoubtedly are human and technology resource constraints that impact project capacity at each R&D phase. Once constraints are effectively incorporated into the model, it may be possible to not only simulate, but optimize the portfolio based on resource availability in different pipeline stages. Such an approach could be a powerful extension to the model, aiding resource allocation and project scheduling.

Another area in which the model could be expanded is in how projects are differentiated. For this study, we chose two dimensions for differentiation—modality and clinical study type. However, other types of differentiation could be added if deemed significant for decision-making. For example, we assumed that backup projects progress with the same duration and transition rate assumptions as non-backup projects. Anecdotal evidence within NIBR suggests that this may not be the case, but historical data supporting this distinction could not be garnered from the data set. Likewise, project progression assumptions could likely differ between Diseases Areas—e.g., a disease that reacts to a topical therapy over the course of weeks would be expected to have significantly shorter clinical trials than a disease for which long-term therapy over many years is required to gauge efficacy.

5.2.2.2 Data improvement

Even if real-world processes are perfectly modeled, the portfolio model will only be as accurate as the historical data that drives it. In this thesis, we have discussed some of the challenges related to capture and analysis of historical data. Given recent efforts to improve collection of project data at NIBR, we expect that data availability will improve over time. As such, these improvements can be incorporated into the model assumptions.

In addition, NIBR can consider the value in automatically populating the model with up-to-date historical data. That is, our research involved manual collection and analysis of data to reach summary statistics suitable for populating the model. However, as additional projects progress through the pipeline, they create additional data over time that should be captured in the historical data set. Rather than frequently re-analyzing the data set for modeling purposes, NIBR can consider developing a direct interface to internal information systems to automatically pull current data into the model.

5.2.2.3 Organizational implementation

We expect that as the model is further rolled out into the organization, users will identify additional use cases that can be accommodated in the model. One already identified is the ability to capture output data for successive simulations to allow comparison across multiple sets of input assumptions. Such a capability could be easily added to the model through additional VBA code modules. Should the eventual demand for new model capabilities create complexity exceeding Excel's capabilities (e.g., creating excessively slow simulations), the organization can explore conversion to a more powerful simulation platform.

Finally, our research was somewhat decoupled from other modeling efforts throughout Novartis, allowing a novel approach to the problem. That said, additional modeling competencies exist

throughout NIBR and Novartis as a whole. Furthermore, portfolio forecasting in early research can potentially feed similar efforts in downstream development. As buy-in grows, Novartis may wish to evaluate where NIBR's portfolio modeling efforts and competencies best fit within the context of the broader organization, and how the organization can best leverage the knowledge created through these efforts.

6 References

1. **Congress of the United States - Congressional Budget Office.** *Research and Development in the Pharmaceutical Industry.* s.l. : Congressional Budget Office, October 2006.

2. **Pharmaceutical Research and Manufacturers of America.** *Pharmaceutical Industry Profile 2010.* Washington, DC : PhRMA, March 2010.

3. *The Impact of Incremental Innovation in Biopharmaceuticals: Drug Utilisation in Original and Supplemental Indications.* **Berndt, Ernst R, Cockburn, Iain M and Grepin, Karen A.** Suppl. 2, 2006, Pharmacoeconomics, Vol. 24, pp. 69-86.

4. **Novartis AG.** *Novartis Company Website.* [Online] [Cited: March 30, 2010.]

5. **Novartis Institutes for Biomedical Research.** *NIBR intranet.* [Online] [Cited: March 31, 2010.]

6. **Novartis AG.** Novartis drug discovery and development process. [Online] [Cited: March 29, 2010.] http://www.nibr.novartis.com/research/drug-discovery.shtml.

7. **Capell, Kerry.** Novartis: Radically Remaking Its Drug Business. *BusinessWeek.* June 22, 2009.

8. **Cooper, Robert G, Edgett, Scott J and Kleinschmidt, Elko J.** *Portdolio Management for New Products.* 2nd Edition. Cambridge : Perseus Publishing, 2001.

9. **Novartis.** *Proof-of-Concept: Innovation in Novartis R&D.* March 2009. promotional material. brochure provided in NIBR lobby.

10. *Realize hidden value through timely portfolio decisions.* **Mohr, Roland, Pacl, Harald and Hartmann, Marcus.** November-December 2008, Research Technology Management, pp. 44-50.

11. **Sterman, J.** *Business Dynamics.* New York : McGraw-Hill, 2000.

12. *Risk management in the development of new products in highly regulated industries.* **Blau, Gary, et al.** 2-7, July 15, 2000, Computers & Chemical Engineering, Vol. 24, pp. 659-664.

13. *Value-driven project and portfolio management in the pharmaceutical industry: Drug discovery versus drug development - Commonalities and differences in portfolio management practice.* **Bode-Greuel, Kerstin M and Nickisch, Klaus J.** 4, s.l. : Palgrave Macmillan, October 2008, Journal of Commercial Biotechnology, Vol. 14, pp. 307-325.

14. *Quantitative risk modelling for new pharmaceutical compounds.* **Tang, Zhengru, et al.** 22, November 2005, Drug Discovery Today, Vol. 10, pp. 1520-1526.

15. *A Simulation-Optimization Framework for Research and Development Pipeline Management.* **Subramanian, Dharmashankar, Pekny, Joseph F. and Reklaitis, Gintaras V.** 10, October 2001, AIChE Journal, Vol. 47, pp. 2226-2242.

16. KMR Group . [Online] [Cited: March 7, 2010.] www.kmrgroup.com.

17. **Excel User.** An Introduction to Excel's Norma Distribution Functions. [Online] [Cited: August 21, 2009.] http://www.exceluser.com/explore/statsnormal.htm.

18. *Teaching Project Simulation in Excel Using PERT-Beta Distributions.* **Davis, Ron.** 3, May 2008, INFORMS Transactions on Education, Vol. 8, pp. 139-148.

19. *Little's Law.* [Online] [Cited: March 30, 2010.] http://web.mit.edu/sgraves/www/papers/Little%27s%20Law-Published.pdf.

20. *The price of innovation: new estimates of drug development costs.* **DiMasi, J.A., Hansen, R.W. and Grabowski, H.G.** 2, 2003, Journal of health economics, Vol. 22, pp. 151-185.

21. **Honkomp, Steven Joseph.** *Solving mathematical programming planning models subject to stochastic task success.* Purdue University. Indiana : Purdue University, 1998. p. 276.

22. *An Analytical Study of the Pert Assumptions.* **MacCrimmon, Kenneth R. and Ryavec, Charles A.** 1, January 1964, Operations research, Vol. 12, pp. 16-37.

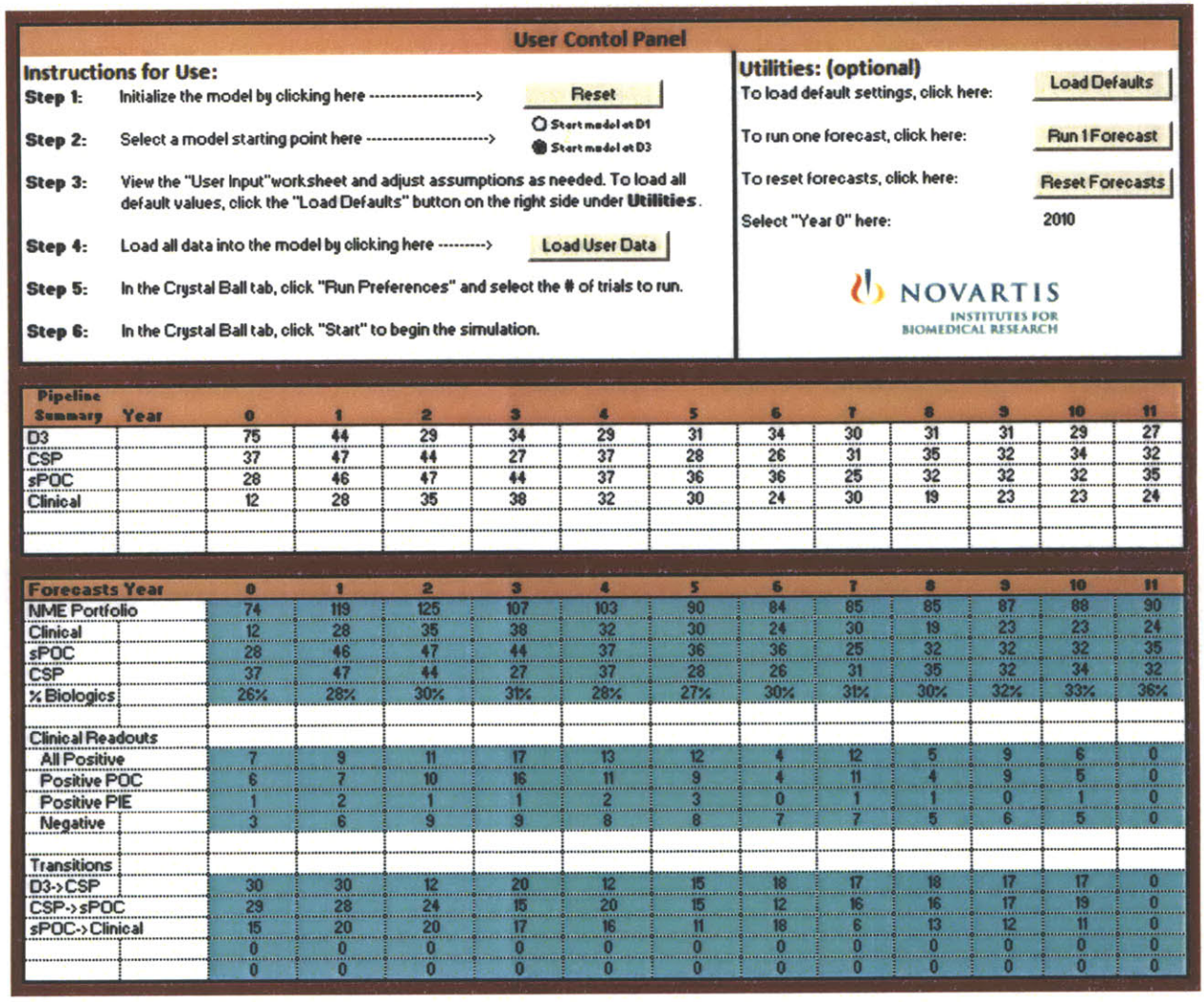
23. *Pharmaceutical Forecasting Model Simulation Guidelines.* **Triantis, John E. and Song, Hu.** 2, s.l. : Graceway Publishing Company, 2007, Journal of Business Forecasting, Vol. 26, pp. 31-37.

7 Appendix A: The NIBR Portfolio Forecasting Model

7.1 User Interface

The following screenshots comprise the Excel workbook that houses the forecasting model.

7.1.1 Worksheet 1: User Dashboard



User Contol Panel

Instructions for Use:

Step 1: Initialize the model by clicking here ----------> Reset

Step 2: Select a model starting point here ----------> ○ Start model at D1 ● Start model at D3

Step 3: View the "User Input" worksheet and adjust assumptions as needed. To load all default values, click the "Load Defaults" button on the right side under **Utilities**.

Step 4: Load all data into the model by clicking here ---------> Load User Data

Step 5: In the Crystal Ball tab, click "Run Preferences" and select the # of trials to run.

Step 6: In the Crystal Ball tab, click "Start" to begin the simulation.

Utilities: (optional)

To load default settings, click here: Load Defaults

To run one forecast, click here: Run 1 Forecast

To reset forecasts, click here: Reset Forecasts

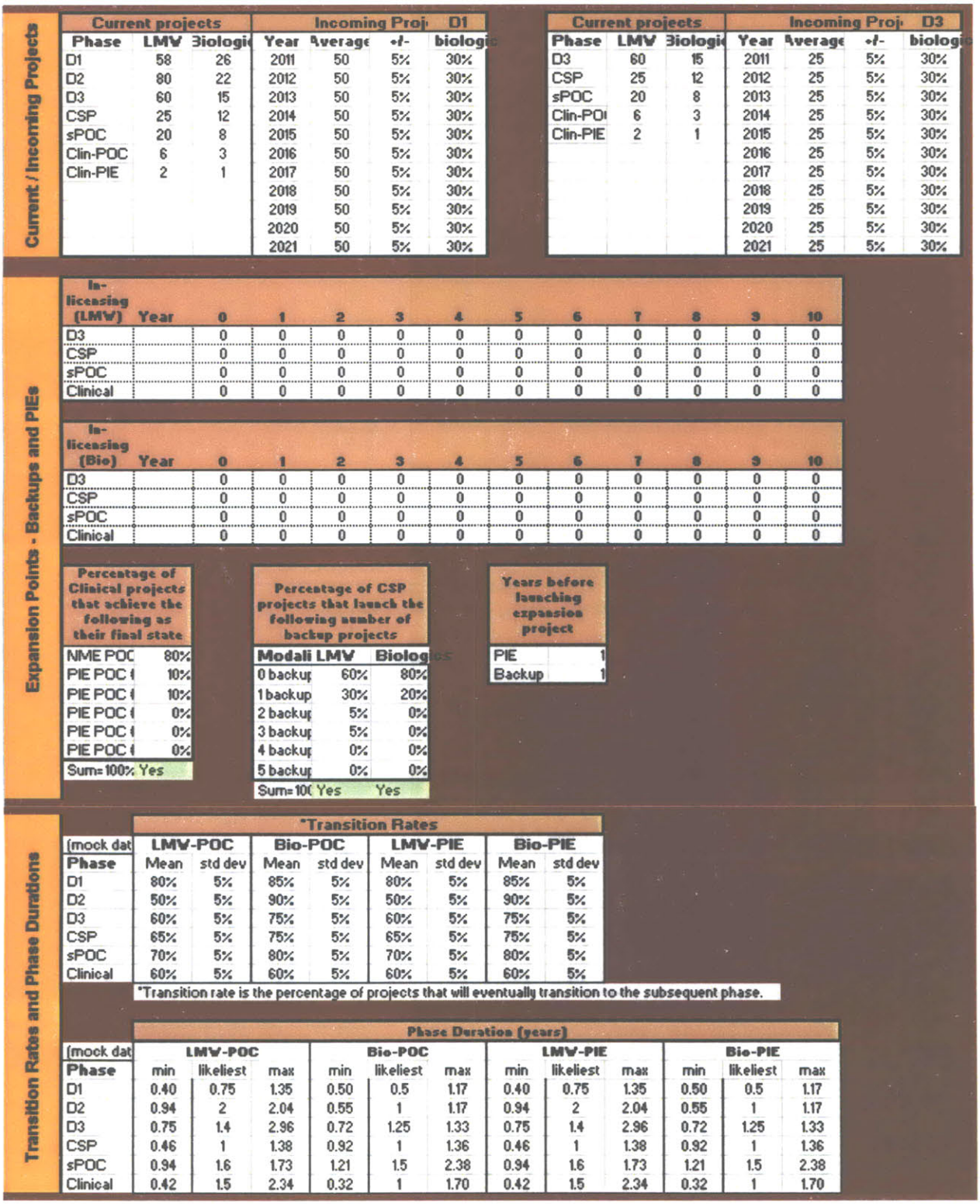
Select "Year 0" here: 2010

NOVARTIS INSTITUTES FOR BIOMEDICAL RESEARCH

Pipeline Summary	Year	0	1	2	3	4	5	6	7	8	9	10	11
D3		75	44	29	34	29	31	34	30	31	31	29	27
CSP		37	47	44	27	37	28	26	31	35	32	34	32
sPOC		28	46	47	44	37	36	36	25	32	32	32	35
Clinical		12	28	35	38	32	30	24	30	19	23	23	24

Forecasts	Year	0	1	2	3	4	5	6	7	8	9	10	11
NME Portfolio		74	119	125	107	103	90	84	85	85	87	88	90
Clinical		12	28	35	38	32	30	24	30	19	23	23	24
sPOC		28	46	47	44	37	36	36	25	32	32	32	35
CSP		37	47	44	27	37	28	26	31	35	32	34	32
% Biologics		26%	28%	30%	31%	28%	27%	30%	31%	30%	32%	33%	35%
Clinical Readouts													
All Positive		7	9	11	17	13	12	4	12	5	9	6	0
Positive POC		6	7	10	16	11	9	4	11	4	9	5	0
Positive PIE		1	2	1	1	2	3	0	1	1	0	1	0
Negative		3	6	9	9	8	8	7	7	5	6	5	0
Transitions													
D3->CSP		30	30	12	20	12	15	18	17	18	17	17	0
CSP->sPOC		29	28	24	15	20	15	12	16	16	17	19	0
sPOC->Clinical		15	20	20	17	16	11	18	6	13	12	11	0
		0	0	0	0	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0	0	0	0	0

7.1.2 Worksheet 2: User Input



Current / Incoming Projects

Current projects			Incoming Proj			D1
Phase	LMV	Biologic	Year	Average	+/-	biologic
D1	58	26	2011	50	5%	30%
D2	80	22	2012	50	5%	30%
D3	60	15	2013	50	5%	30%
CSP	25	12	2014	50	5%	30%
sPOC	20	8	2015	50	5%	30%
Clin-POC	6	3	2016	50	5%	30%
Clin-PIE	2	1	2017	50	5%	30%
			2018	50	5%	30%
			2019	50	5%	30%
			2020	50	5%	30%
			2021	50	5%	30%

Current projects			Incoming Proj			D3
Phase	LMV	Biologic	Year	Average	+/-	biologic
D3	60	15	2011	25	5%	30%
CSP	25	12	2012	25	5%	30%
sPOC	20	8	2013	25	5%	30%
Clin-POC	6	3	2014	25	5%	30%
Clin-PIE	2	1	2015	25	5%	30%
			2016	25	5%	30%
			2017	25	5%	30%
			2018	25	5%	30%
			2019	25	5%	30%
			2020	25	5%	30%
			2021	25	5%	30%

Expansion Points - Backups and PIEs

In-licensing (LMV)	Year	0	1	2	3	4	5	6	7	8	9	10
D3		0	0	0	0	0	0	0	0	0	0	0
CSP		0	0	0	0	0	0	0	0	0	0	0
sPOC		0	0	0	0	0	0	0	0	0	0	0
Clinical		0	0	0	0	0	0	0	0	0	0	0

In-licensing (Bio)	Year	0	1	2	3	4	5	6	7	8	9	10
D3		0	0	0	0	0	0	0	0	0	0	0
CSP		0	0	0	0	0	0	0	0	0	0	0
sPOC		0	0	0	0	0	0	0	0	0	0	0
Clinical		0	0	0	0	0	0	0	0	0	0	0

Percentage of Clinical projects that achieve the following as their final state	
NME POC	80%
PIE POC t	10%
PIE POC t	10%
PIE POC t	0%
PIE POC t	0%
PIE POC t	0%
Sum=100%	Yes

Percentage of CSP projects that launch the following number of backup projects		
Modali	LMV	Biologics
0 backup	60%	80%
1 backup	30%	20%
2 backup	5%	0%
3 backup	5%	0%
4 backup	0%	0%
5 backup	0%	0%
Sum=100	Yes	Yes

Years before launching expansion project	
PIE	1
Backup	1

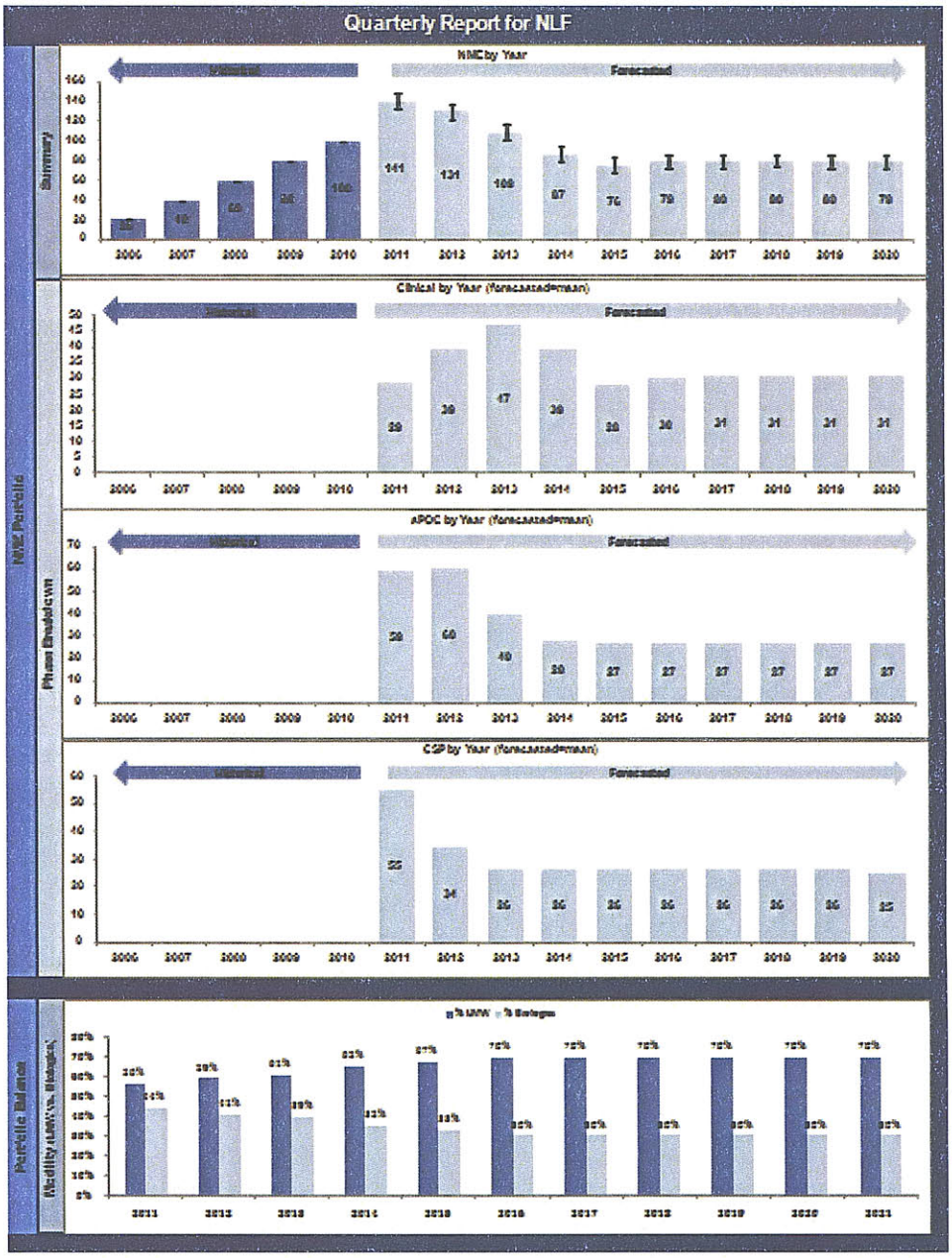
Transition Rates and Phase Durations

	*Transition Rates							
(mock dat	LMV-POC		Bio-POC		LMV-PIE		Bio-PIE	
Phase	Mean	std dev	Mean	std dev	Mean	std dev	Mean	std dev
D1	80%	5%	85%	5%	80%	5%	85%	5%
D2	50%	5%	90%	5%	50%	5%	90%	5%
D3	60%	5%	75%	5%	60%	5%	75%	5%
CSP	65%	5%	75%	5%	65%	5%	75%	5%
sPOC	70%	5%	80%	5%	70%	5%	80%	5%
Clinical	60%	5%	60%	5%	60%	5%	60%	5%

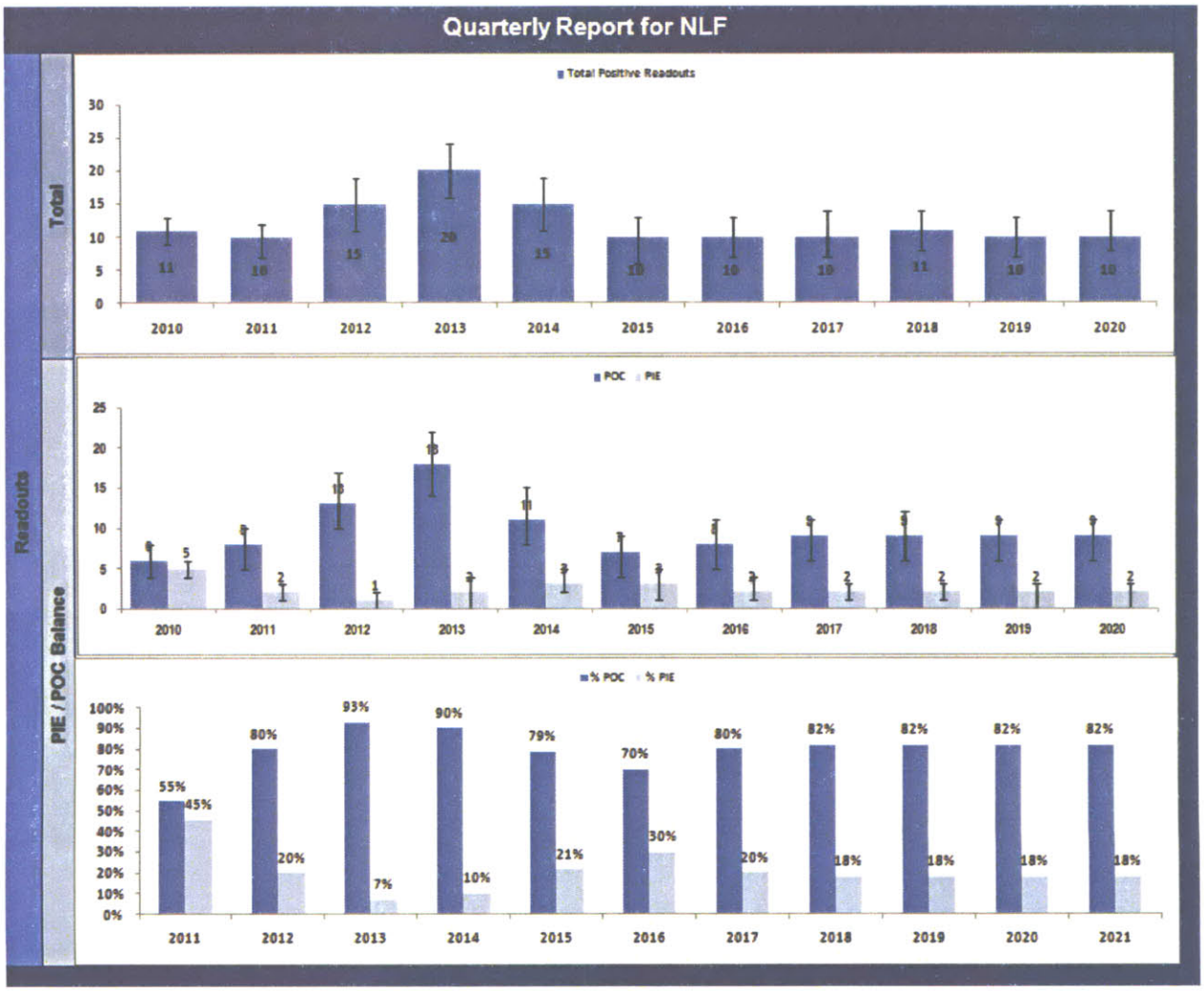
*Transition rate is the percentage of projects that will eventually transition to the subsequent phase.

	Phase Duration (years)											
(mock dat	LMV-POC			Bio-POC			LMV-PIE			Bio-PIE		
Phase	min	likeliest	max	min	likeliest	max	min	likeliest	max	min	likeliest	max
D1	0.40	0.75	1.35	0.50	0.5	1.17	0.40	0.75	1.35	0.50	0.5	1.17
D2	0.94	2	2.04	0.55	1	1.17	0.94	2	2.04	0.55	1	1.17
D3	0.75	1.4	2.96	0.72	1.25	1.33	0.75	1.4	2.96	0.72	1.25	1.33
CSP	0.46	1	1.38	0.92	1	1.36	0.46	1	1.38	0.92	1	1.36
sPOC	0.94	1.6	1.73	1.21	1.5	2.38	0.94	1.6	1.73	1.21	1.5	2.38
Clinical	0.42	1.5	2.34	0.32	1	1.70	0.42	1.5	2.34	0.32	1	1.70

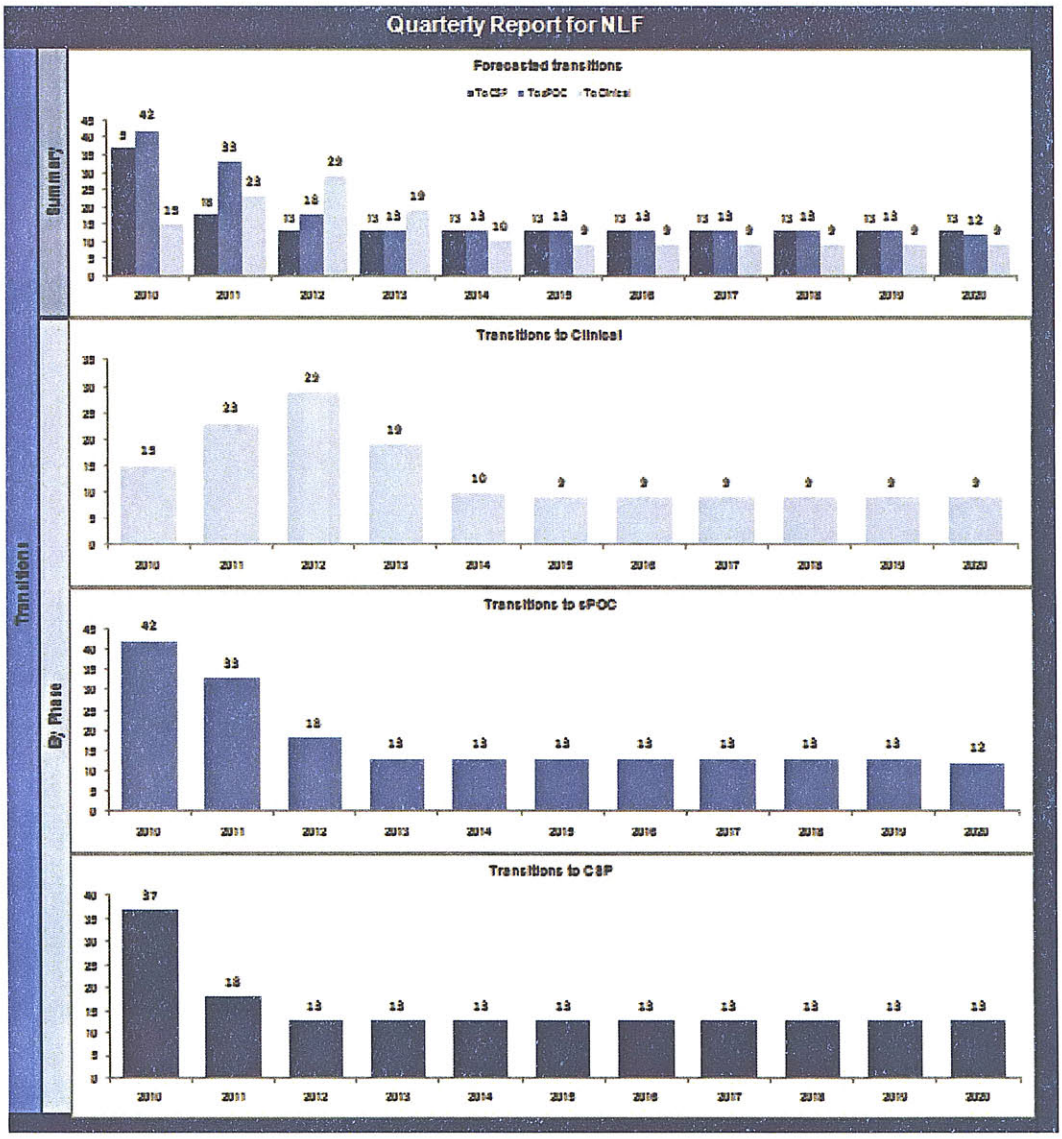
7.1.3 Worksheet 3: Leadership Report – NMEs



7.1.4 Worksheet 4: Leadership Report – Readouts



7.1.5 Worksheet 4: Leadership Report – Transitions



7.1.6 Worksheet 4: Leadership Report – Data

NME by Year

	Mean	10%	90%	Mean-10%	90%-Mean
2006	20	0	0	0	0
2007	40	0	0	0	0
2008	60	0	0	0	0
2009	80	0	0	0	0
2010	100	0	0	0	0
2011	141	134	149	7	8
2012	131	123	138	8	7
2013	109	103	117	6	8
2014	87	80	95	7	8
2015	76	68	83	8	7
2016	79	72	86	7	7
2017	80	73	86	7	6
2018	80	74	87	6	7
2019	80	73	86	7	6
2020	79	72	87	7	8

NME by Year (mean)

	CSP	sPOC	Clinical
2006			
2007			
2008			
2009			
2010			
2011	55	59	29
2012	34	60	39
2013	26	40	47
2014	26	28	39
2015	26	27	28
2016	26	27	30
2017	26	27	31
2018	26	27	31
2019	26	27	31
2020	25	27	31

Paritive Readouts

	Total Paritiv	POC	PIE	Negative	Total 10%	Total 90%	Total LB	Total UB	POC 10%	POC 90%	POC LB	POC UB	PIE 10%	PIE 90%	PIE LB	PIE UB
2010	11	6	5	6	9	13	2	2	4	8	2	2	4	6	1	1
2011	10	8	2	9	7	12	3	2	5	10	3	2	1	3	1	1
2012	15	13	1	11	11	19	4	4	10	17	3	4	0	2	1	1
2013	20	18	2	14	16	24	4	4	14	22	4	4	0	4	2	2
2014	15	11	3	11	11	19	4	4	8	15	3	4	2	5	1	2
2015	10	7	3	8	6	13	4	3	4	9	3	2	1	5	2	2
2016	10	8	2	8	7	13	3	3	5	11	3	3	1	4	1	2
2017	10	9	2	9	7	14	3	4	6	11	3	2	1	3	1	1
2018	11	9	2	9	8	14	3	3	6	12	3	3	1	3	1	1
2019	10	9	2	9	7	13	3	3	6	11	3	2	0	3	2	1
2020	10	9	2	9	8	14	2	4	6	11	3	2	0	3	2	1

Transitions by Year (mean

	To CSP	To sPOC	To Clinical
2010	37	42	15
2011	18	33	23
2012	13	18	29
2013	13	13	19
2014	13	13	10
2015	13	13	9
2016	13	13	9
2017	13	13	9
2018	13	13	9
2019	13	13	9
2020	13	12	9

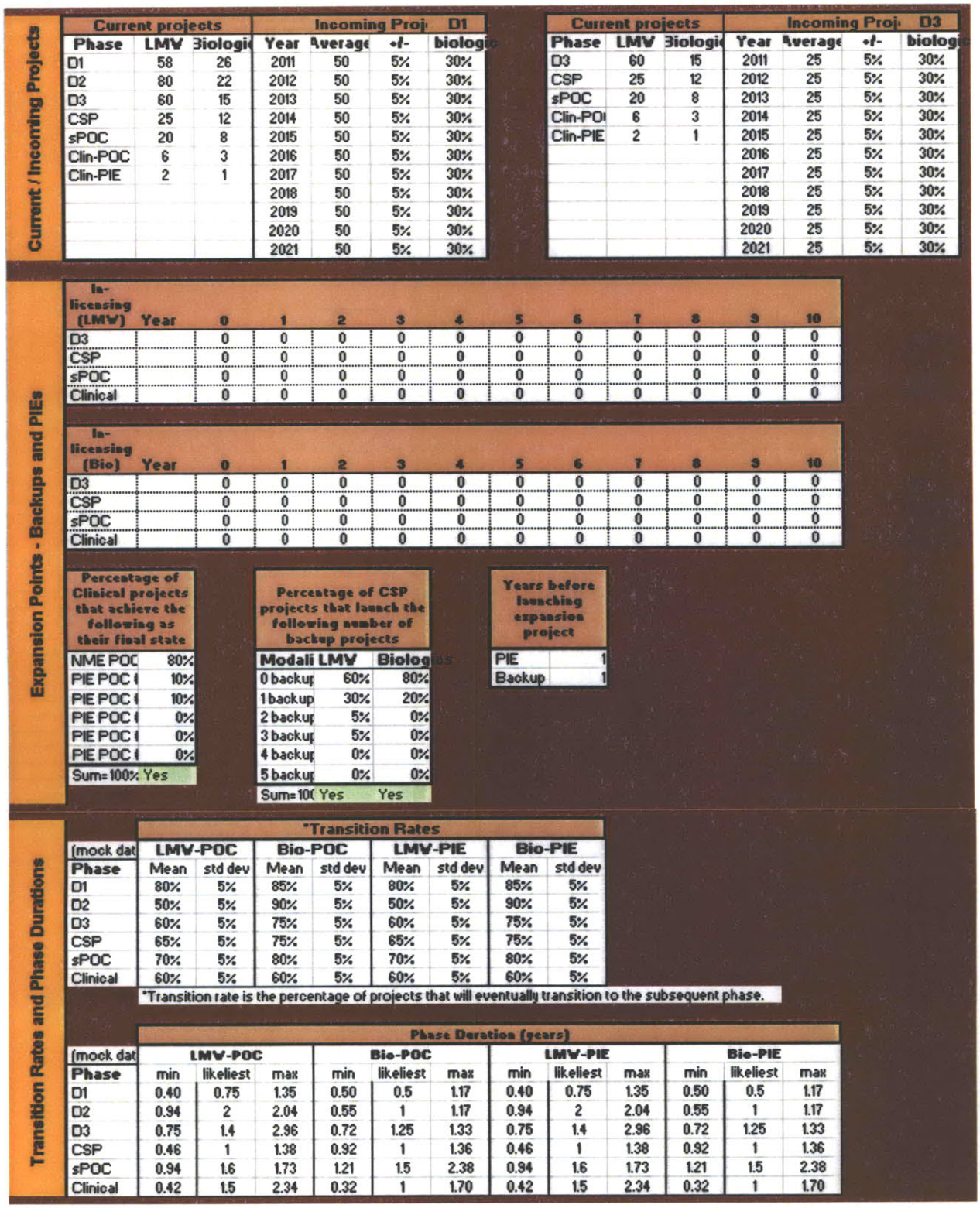
Modality

	% Biologi	% LMW
2011	44%	56%
2012	41%	59%
2013	39%	61%
2014	35%	65%
2015	33%	67%
2016	30%	70%
2017	30%	70%
2018	30%	70%
2019	30%	70%
2020	30%	70%
2021	30%	70%

PIE/POC ratio

	% PIE	% POC
2011	45%	55%
2012	20%	80%
2013	7%	93%
2014	10%	90%
2015	21%	79%
2016	30%	70%
2017	20%	80%
2018	18%	82%
2019	18%	82%
2020	18%	82%
2021	18%	82%

7.1.7 Worksheet 7: Default Values



Current / Incoming Projects

Current projects			Incoming Proj			D1
Phase	LMV	Biologi	Year	Average	+/-	biologi
D1	58	26	2011	50	5%	30%
D2	80	22	2012	50	5%	30%
D3	60	15	2013	50	5%	30%
CSP	25	12	2014	50	5%	30%
sPOC	20	8	2015	50	5%	30%
Clin-POC	6	3	2016	50	5%	30%
Clin-PIE	2	1	2017	50	5%	30%
			2018	50	5%	30%
			2019	50	5%	30%
			2020	50	5%	30%
			2021	50	5%	30%

Current projects			Incoming Proj			D3
Phase	LMV	Biologi	Year	Average	+/-	biologi
D3	60	15	2011	25	5%	30%
CSP	25	12	2012	25	5%	30%
sPOC	20	8	2013	25	5%	30%
Clin-PO	6	3	2014	25	5%	30%
Clin-PIE	2	1	2015	25	5%	30%
			2016	25	5%	30%
			2017	25	5%	30%
			2018	25	5%	30%
			2019	25	5%	30%
			2020	25	5%	30%
			2021	25	5%	30%

Expansion Points - Backups and PIEs

In-licensing (LMV)	Year	0	1	2	3	4	5	6	7	8	9	10
D3		0	0	0	0	0	0	0	0	0	0	0
CSP		0	0	0	0	0	0	0	0	0	0	0
sPOC		0	0	0	0	0	0	0	0	0	0	0
Clinical		0	0	0	0	0	0	0	0	0	0	0

In-licensing (Bio)	Year	0	1	2	3	4	5	6	7	8	9	10
D3		0	0	0	0	0	0	0	0	0	0	0
CSP		0	0	0	0	0	0	0	0	0	0	0
sPOC		0	0	0	0	0	0	0	0	0	0	0
Clinical		0	0	0	0	0	0	0	0	0	0	0

Percentage of Clinical projects that achieve the following as their final state	
NME POC	80%
PIE POC	10%
PIE POC	10%
PIE POC	0%
PIE POC	0%
PIE POC	0%
Sum=100%	Yes

Percentage of CSP projects that launch the following number of backup projects		
Modali	LMV	Biolog
0 backup	60%	80%
1 backup	30%	20%
2 backup	5%	0%
3 backup	5%	0%
4 backup	0%	0%
5 backup	0%	0%
Sum=10	Yes	Yes

Years before launching expansion project	
PIE	1
Backup	1

Transition Rates and Phase Durations

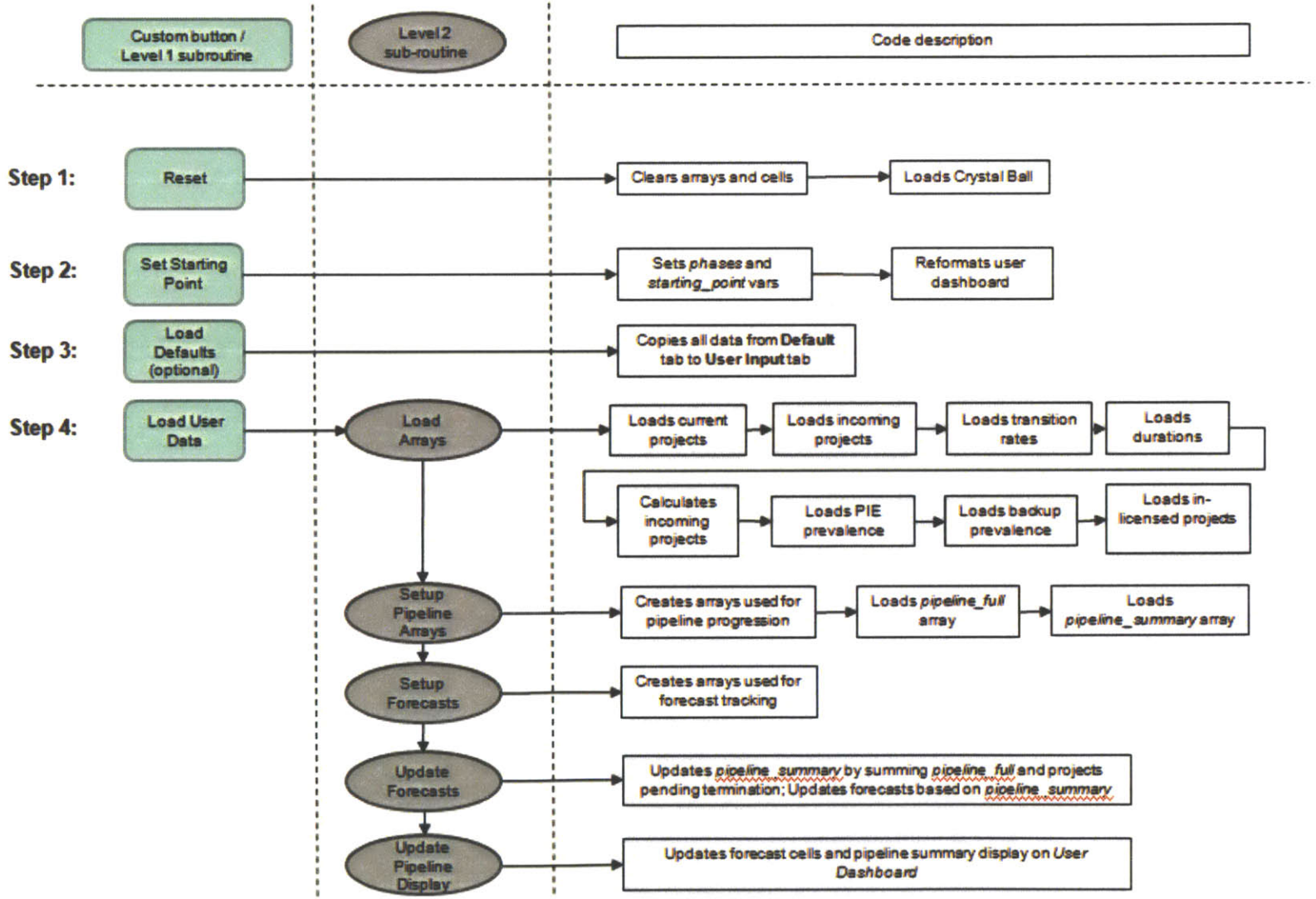
	*Transition Rates							
(mock dat	LMV-POC		Bio-POC		LMV-PIE		Bio-PIE	
Phase	Mean	std dev	Mean	std dev	Mean	std dev	Mean	std dev
D1	80%	5%	85%	5%	80%	5%	85%	5%
D2	50%	5%	90%	5%	50%	5%	90%	5%
D3	60%	5%	75%	5%	60%	5%	75%	5%
CSP	65%	5%	75%	5%	65%	5%	75%	5%
sPOC	70%	5%	80%	5%	70%	5%	80%	5%
Clinical	60%	5%	60%	5%	60%	5%	60%	5%

*Transition rate is the percentage of projects that will eventually transition to the subsequent phase.

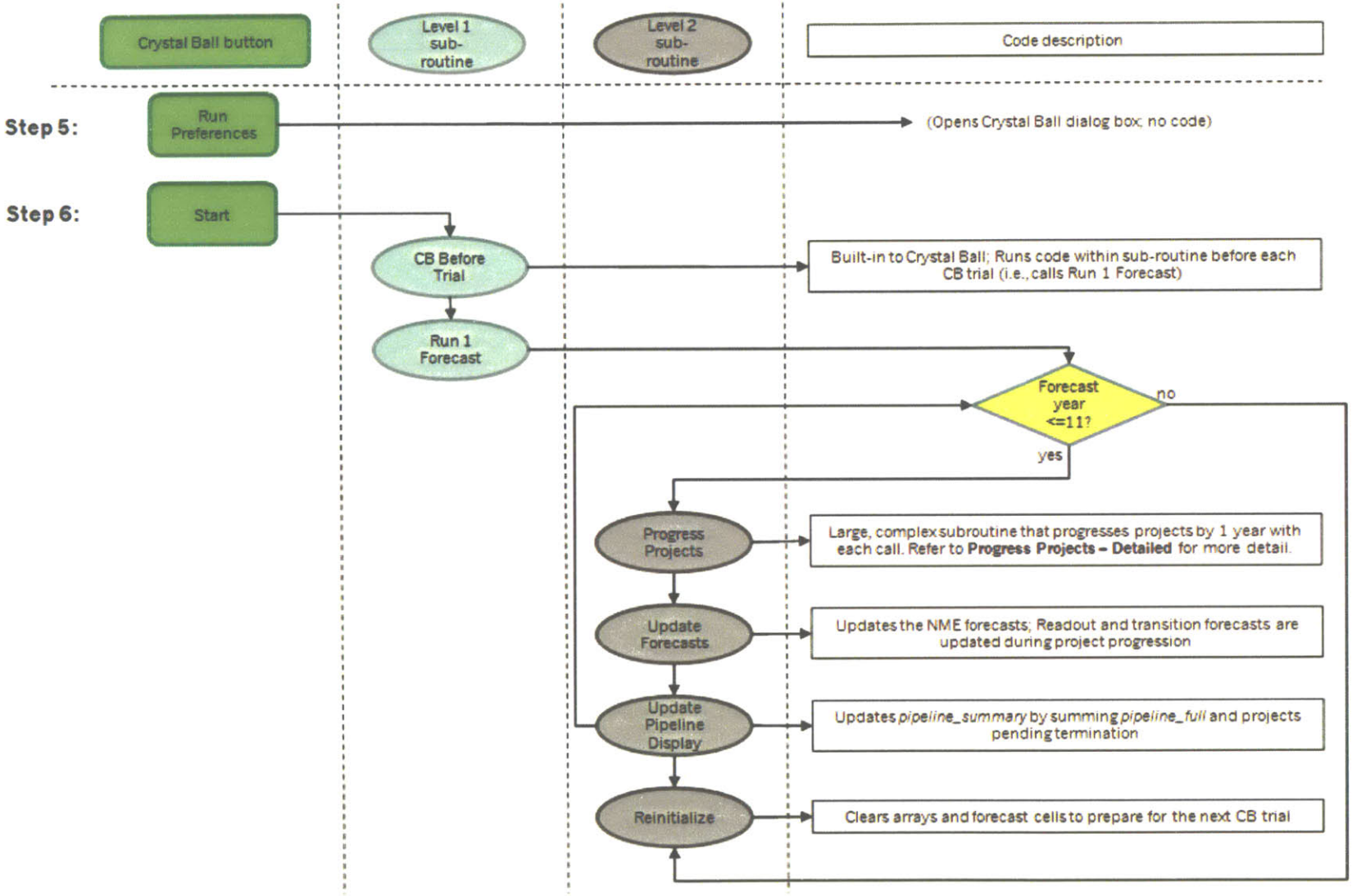
	Phase Duration (years)											
(mock dat	LMV-POC			Bio-POC			LMV-PIE			Bio-PIE		
Phase	min	likeliest	max	min	likeliest	max	min	likeliest	max	min	likeliest	max
D1	0.40	0.75	1.35	0.50	0.5	1.17	0.40	0.75	1.35	0.50	0.5	1.17
D2	0.94	2	2.04	0.55	1	1.17	0.94	2	2.04	0.55	1	1.17
D3	0.75	1.4	2.96	0.72	1.25	1.33	0.75	1.4	2.96	0.72	1.25	1.33
CSP	0.46	1	1.38	0.92	1	1.36	0.46	1	1.38	0.92	1	1.36
sPOC	0.94	1.6	1.73	1.21	1.5	2.38	0.94	1.6	1.73	1.21	1.5	2.38
Clinical	0.42	1.5	2.34	0.32	1	1.70	0.42	1.5	2.34	0.32	1	1.70

7.2 Model Logic Flowcharts

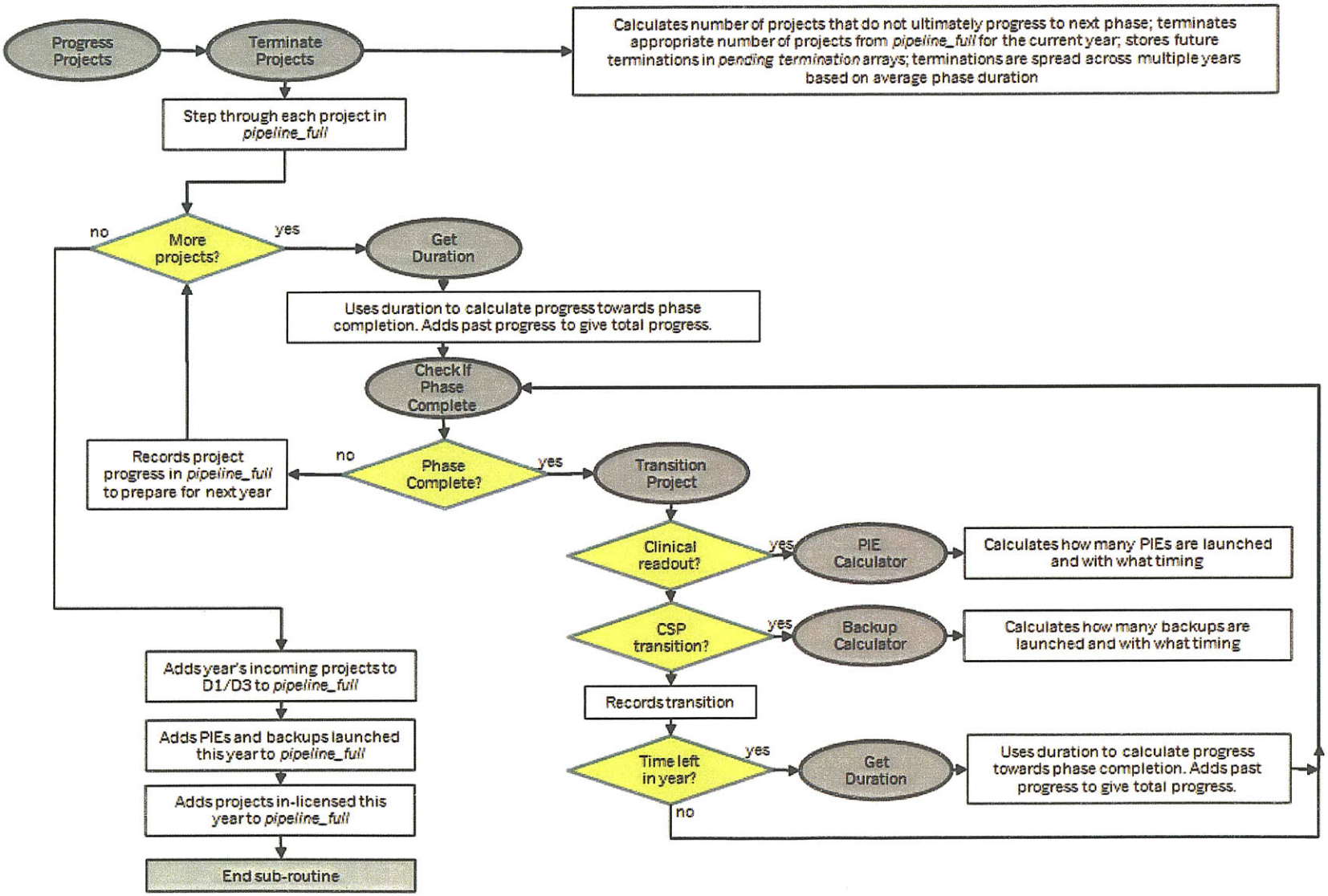
Pipeline Model VBA Flowchart (1 of 2)



Pipeline Model VBA Flowchart (2 of 2)



Progress Projects – Detailed



7.3 VBA Code

7.3.1 Project object: ThisWorkbook

```
Public Function CBBeforeTrial(atrial As Long) As Integer

    Run_1Forecast

End Function
```

7.3.2 Module 1: Main code body

```
Option Base 1   'sets array default array subscripts to 1
Const forecast_years As Integer = 11           'number of forecast years
Const assumption_sets As Integer = 4           'number of assumption sets (sets of durations and transition_rates)
'Below declares various arrays and variables
Dim current_projects() As Variant, pipeline_full() As Variant, temp_pipeline_full() As Variant, pipeline_summary() As Variant,
percent_biologics() As Single, PIE_summary() As Integer
Dim forecasts_NME() As Single, forecasts_readouts() As Integer, forecasts_transitions() As Integer
Dim projects_to_terminate() As Variant, temp_projects_to_terminate() As Variant, new_projects_pending_termination() As Variant,
old_projects_pending_termination() As Variant, entry_class() As Variant
Dim PIE_tracker(forecast_years, 100, assumption_sets - 2) As Integer, backup_tracker(forecast_years, 100, assumption_sets - 2) As
Integer
Dim transition_rates() As Variant, transition_rates_assumptions As Variant, termination_rates() As Variant, iteration_rates() As
Variant
Dim durations() As Variant, durations_assumptions As Variant
Dim incoming_projects_entered() As Variant, incoming_projects_calculated() As Variant, inlicensed_projects() As Variant
Dim phases As Integer                  'number of phases in model
Dim starting_point As Integer          'number 1 or 2 indicating start at D1 or D3
Dim step_counter As Integer            'tracks how many years forward the model is stepped
Dim col_counter As Integer
Dim percent_phase_remaining As Integer
Dim percent_phase_accomplished As Integer
Dim percent_year_used As Single
Dim POC_only As Single, PIE1 As Single, PIE2 As Single, PIE3 As Single, PIE4 As Single, PIE5 As Single
Dim PIE_boundary1 As Single, PIE_boundary2 As Single, PIE_boundary3 As Single, PIE_boundary4 As Single, PIE_boundary5 As Single,
PIE_boundary6 As Single
Dim PIE_time_delay As Single, backup_time_delay As Single
Dim backup_boundary1_LMW As Single, backup_boundary2_LMW As Single, backup_boundary3_LMW As Single, backup_boundary4_LMW As Single,
backup_boundary5_LMW As Single, backup_boundary6_LMW As Single
Dim backup_boundary1_Bio As Single, backup_boundary2_Bio As Single, backup_boundary3_Bio As Single, backup_boundary4_Bio As Single,
backup_boundary5_Bio As Single, backup_boundary6_Bio As Single
Dim backups(6, 2) As Single
Dim i As Integer, j As Integer, k As Integer, row_counter As Integer
```

```
Sub Reset()
    Worksheets("User Dashboard").Activate
    step_counter = 0
    Erase pipeline_summary              'clears pipeline array values
    Erase PIE_summary
    Erase pipeline_full
    Erase temp_pipeline_full
    Erase new_projects_pending_termination
    Erase old_projects_pending_termination
    Erase PIE_tracker
    Erase backup_tracker
    Worksheets("User Dashboard").OptionButtons("Option Button 413").Value = xlOff
    Worksheets("User Dashboard").OptionButtons("Option Button 415").Value = xlOff

    Worksheets("User Dashboard").Range("D18:O23,D26:O43").ClearContents 'clears worksheets showing pipeline array values and forecasts
    Worksheets("User Dashboard").Range("D26:O30,D33:O36,D39:O43").Value = 0         'Needed for cystal ball forecasts, since blank
forecast cells will cause an error
    Worksheets("Pipeline Summary").UsedRange.Clear
    Worksheets("Pipeline Full LMW-POC").UsedRange.Clear
    Worksheets("Pipeline Full Bio-POC").UsedRange.Clear
    Worksheets("Pipeline Full LMW-PIE").UsedRange.Clear
    Worksheets("Pipeline Full Bio-PIE").UsedRange.Clear
    ActiveSheet.Cells.Select
    ActiveSheet.Cells(1, 1).Select

    If cb.CBLoaded() = False Then cb.Startup           'checks to see if Crystal Ball is open; if not, opens it
    cb.ResetND
    cb.ClearDataND
End Sub

Sub SetStartingPoint()
    starting_point = Worksheets("User Dashboard").Range("H6").Value
    If starting_point = 1 Then
        phases = 6
        Worksheets("User Dashboard").Range("B18:B22").Value = Worksheets("Default Values").Range("B4:B8").Value
        Worksheets("User Dashboard").Range("B23").Value = "Clinical"
        Worksheets("User Dashboard").Range("B39").Value = "D1->D2"
        Worksheets("User Dashboard").Range("B40").Value = "D2->D3"
        Worksheets("User Dashboard").Range("B41").Value = "D3->CSP"
        Worksheets("User Dashboard").Range("B42").Value = "CSP->sPOC"
        Worksheets("User Dashboard").Range("B43").Value = "sPOC->Clinical"
    ElseIf starting_point = 2 Then
        phases = 4
        Worksheets("User Dashboard").Range("B18:B20") = Worksheets("Default Values").Range("J4:J6").Value
        Worksheets("User Dashboard").Range("B21").Value = "Clinical"
        Worksheets("User Dashboard").Range("B22:B23") = ""
        Worksheets("User Dashboard").Range("B39").Value = "D3->CSP"
        Worksheets("User Dashboard").Range("B40").Value = "CSP->sPOC"
        Worksheets("User Dashboard").Range("B41").Value = "sPOC->Clinical"
        Worksheets("User Dashboard").Range("B42:B43") = ""
    Else
```

```
        MsgBox ("There is a problem with this control.")
    End If
End Sub

Sub LoadUserData()
    Application.ScreenUpdating = False
    LoadArrays
    SetupPipelineArrays
    SetupForecasts
    UpdateForecasts
    UpdatePipelineDisplay
    Worksheets("User Dashboard").Activate
    Application.ScreenUpdating = True
End Sub

Sub Run_1Forecast()
    If Worksheets("User Dashboard").Range("H6").Value = 0 Then
        MsgBox ("Please select a starting point in Step 2.")
        Exit Sub
    End If
    Application.ScreenUpdating = False
    cb.ClearDataND
    'LoadUserData                                    'Note: This can be toggled on to update input data from the spreadsheet with each
simulation trial. This would be neede to use variable CB assumptions or to use CB features such as Decision table
    For years = 1 To forecast_years
        Worksheets("User Dashboard").Activate
        step_counter = step_counter + 1
        ProgressProjects
        UpdateForecasts
        UpdatePipelineDisplay
    Next
    ReInitialize                'resets variables and arrays after forecast period before CB runs next simulation trial
    Application.ScreenUpdating = True
End Sub

Sub ReInitialize()
    'The following sets forecast arrays to 0 (can't be empty for CB)
    For i = 1 To UBound(forecasts_NME)
        For j = 1 To UBound(forecasts_NME, 2)
            forecasts_NME(i, j) = 0
        Next
    Next
    For i = 1 To UBound(forecasts_readouts)
        For j = 1 To UBound(forecasts_readouts, 2)
            forecasts_readouts(i, j) = 0
        Next
    Next
    For i = 1 To UBound(forecasts_transitions)
        For j = 1 To UBound(forecasts_transitions, 2)
            forecasts_transitions(i, j) = 0
        Next
```

```
    Next
    step_counter = 0
    Erase pipeline_summary
    Erase PIE_summary
    Erase pipeline_full
    Erase temp_pipeline_full
    Erase new_projects_pending_termination
    Erase old_projects_pending_termination
    Erase PIE_tracker
    Erase backup_tracker
    LoadArrays
    SetupPipelineArrays
    UpdateForecasts
    Worksheets("User Dashboard").Activate
End Sub

Sub LoadArrays()
    'Note that some parts of this subroutine differ depending on the model starting point (D1 or D3), so much of the code is repeated
with different cell ranges as part of a select case statement.
    ReDim current_projects(phases, assumption_sets)
    ReDim incoming_projects_entered(forecast_years, 3)                  'dimension 1=phases, 2=mean, stdev as a percentage, %bio
    ReDim transition_rates_assumptions(phases, 2, assumption_sets)      'dimension 1=phases, 2=mean&stdev, 3=assumption set
    ReDim durations_assumptions(phases, 3, assumption_sets)             'dimension 1=phases, 2=min,likeliest,max 3=assumption set

    Select Case starting_point
    Case 1                      'Starting point = D1
        'current projects
        With Worksheets("User Input").Range("C4:D10")
            For i = 1 To phases
                For j = 1 To 2
                    current_projects(i, j) = .Cells(i, j)
                    If i = phases Then current_projects(i, j + 2) = .Cells(i + 1, j)
                Next
            Next
        End With
        'incoming projects
        With Worksheets("User Input").Range("F4:H14")
            For i = 1 To UBound(incoming_projects_entered, 1)
                 For j = 1 To UBound(incoming_projects_entered, 2)
                     incoming_projects_entered(i, j) = .Cells(i, j)
                 Next
            Next
        End With
        'transition rates - each column represents different assumption set
        With Worksheets("User Input").Range("C41:D46")            'LMW-POC
            For i = 1 To UBound(transition_rates_assumptions, 1)
                For j = 1 To UBound(transition_rates_assumptions, 2)
                    transition_rates_assumptions(i, j, 1) = .Cells(i, j)
                Next
            Next
        End With
```

```
With Worksheets("User Input").Range("E41:F46")            'Bio-POC
    For i = 1 To UBound(transition_rates_assumptions, 1)
        For j = 1 To UBound(transition_rates_assumptions, 2)
            transition_rates_assumptions(i, j, 2) = .Cells(i, j)
        Next
    Next
End With
With Worksheets("User Input").Range("G41:H46")            'LMW-PIE
    For i = 1 To UBound(transition_rates_assumptions, 1)
        For j = 1 To UBound(transition_rates_assumptions, 2)
            transition_rates_assumptions(i, j, 3) = .Cells(i, j)
        Next
    Next
End With
With Worksheets("User Input").Range("I41:J46")            'Bio-PIE
    For i = 1 To UBound(transition_rates_assumptions, 1)
        For j = 1 To UBound(transition_rates_assumptions, 2)
            transition_rates_assumptions(i, j, 4) = .Cells(i, j)
        Next
    Next
End With

'durations - each column represents different assumption set
With Worksheets("User Input").Range("C52:E57")            'LMW-POC
    For i = 1 To UBound(durations_assumptions, 1)
        For j = 1 To UBound(durations_assumptions, 2)
            durations_assumptions(i, j, 1) = .Cells(i, j)
        Next
    Next
End With
With Worksheets("User Input").Range("F52:H57")            'Bio-POC
    For i = 1 To UBound(durations_assumptions, 1)
        For j = 1 To UBound(durations_assumptions, 2)
            durations_assumptions(i, j, 2) = .Cells(i, j)
        Next
    Next
End With
With Worksheets("User Input").Range("I52:K57")            'LMW-PIE
    For i = 1 To UBound(durations_assumptions, 1)
        For j = 1 To UBound(durations_assumptions, 2)
            durations_assumptions(i, j, 3) = .Cells(i, j)
        Next
    Next
End With
With Worksheets("User Input").Range("L52:N57")            'Bio-PIE
    For i = 1 To UBound(durations_assumptions, 1)
        For j = 1 To UBound(durations_assumptions, 2)
            durations_assumptions(i, j, 4) = .Cells(i, j)
        Next
    Next
End With
```

```
Case 2          'Starting point = D3
    'current projects
    With Worksheets("User Input").Range("K4:L8")
        For i = 1 To phases
            For j = 1 To 2
                current_projects(i, j) = .Cells(i, j)
                If i = phases Then current_projects(i, j + 2) = .Cells(i + 1, j)
            Next
        Next
    End With
    'incoming projects
    With Worksheets("User Input").Range("N4:P14")
        For i = 1 To UBound(incoming_projects_entered, 1)
             For j = 1 To UBound(incoming_projects_entered, 2)
                 incoming_projects_entered(i, j) = .Cells(i, j)
             Next
        Next
    End With
    'transition rates - each column represents different assumption set
    With Worksheets("User Input").Range("C43:D46")            'LMW-POC
        For i = 1 To UBound(transition_rates_assumptions, 1)
            For j = 1 To UBound(transition_rates_assumptions, 2)
                transition_rates_assumptions(i, j, 1) = .Cells(i, j)
            Next
        Next
    End With
    With Worksheets("User Input").Range("E43:F46")            'Bio-POC
        For i = 1 To UBound(transition_rates_assumptions, 1)
            For j = 1 To UBound(transition_rates_assumptions, 2)
                transition_rates_assumptions(i, j, 2) = .Cells(i, j)
            Next
        Next
    End With
    With Worksheets("User Input").Range("G43:H46")            'LMW-PIE
        For i = 1 To UBound(transition_rates_assumptions, 1)
            For j = 1 To UBound(transition_rates_assumptions, 2)
                transition_rates_assumptions(i, j, 3) = .Cells(i, j)
            Next
        Next
    End With
    With Worksheets("User Input").Range("I43:J46")            'Bio-PIE
        For i = 1 To UBound(transition_rates_assumptions, 1)
            For j = 1 To UBound(transition_rates_assumptions, 2)
                transition_rates_assumptions(i, j, 4) = .Cells(i, j)
            Next
        Next
    End With

    'durations - each column represents different assumption set
    With Worksheets("User Input").Range("C54:E57")            'LMW-POC
```

```
            For i = 1 To UBound(durations_assumptions, 1)
                For j = 1 To UBound(durations_assumptions, 2)
                    durations_assumptions(i, j, 1) = .Cells(i, j)
                Next
            Next
        End With
        With Worksheets("User Input").Range("F54:H57")            'Bio-POC
            For i = 1 To UBound(durations_assumptions, 1)
                For j = 1 To UBound(durations_assumptions, 2)
                    durations_assumptions(i, j, 2) = .Cells(i, j)
                Next
            Next
        End With
        With Worksheets("User Input").Range("I54:K57")            'LMW-PIE
            For i = 1 To UBound(durations_assumptions, 1)
                For j = 1 To UBound(durations_assumptions, 2)
                    durations_assumptions(i, j, 3) = .Cells(i, j)
                Next
            Next
        End With
        With Worksheets("User Input").Range("L54:N57")            'Bio-PIE
            For i = 1 To UBound(durations_assumptions, 1)
                For j = 1 To UBound(durations_assumptions, 2)
                    durations_assumptions(i, j, 4) = .Cells(i, j)
                Next
            Next
        End With
    End Select

    'The following calculates actual incoming project count based on user-entered data
    ReDim incoming_projects_calculated(forecast_years, assumption_sets)
    For i = 1 To UBound(incoming_projects_calculated, 1)
        project_total = Round(Application.WorksheetFunction.NormInv(Rnd, incoming_projects_entered(i, 1),
(incoming_projects_entered(i, 2) * incoming_projects_entered(i, 1))))
        incoming_projects_calculated(i, 1) = Round(project_total * (1 - incoming_projects_entered(i, 3)))       'incoming LMW
        incoming_projects_calculated(i, 2) = project_total - incoming_projects_calculated(i, 1)         'incoming Bio
        incoming_projects_calculated(i, 3) = 0
        incoming_projects_calculated(i, 4) = 0
    Next

    'PIE prevalence
    Worksheets("User Input").Activate
    With Range("C29:C34")
        POC_only = 100 * .Cells(1, 1)
        PIE1 = 100 * .Cells(2, 1)
        PIE2 = 100 * .Cells(3, 1)
        PIE3 = 100 * .Cells(4, 1)
        PIE4 = 100 * .Cells(5, 1)
        PIE5 = 100 * .Cells(6, 1)
    End With
    PIE_time_delay = Range("J29")
```

```
    'The following are used later to calculate the number of PIEs
    PIE_boundary1 = POC_only
    PIE_boundary2 = POC_only + PIE1
    PIE_boundary3 = POC_only + PIE1 + PIE2
    PIE_boundary4 = POC_only + PIE1 + PIE2 + PIE3
    PIE_boundary5 = POC_only + PIE1 + PIE2 + PIE3 + PIE4
    PIE_boundary6 = POC_only + PIE1 + PIE2 + PIE3 + PIE4 + PIE5

    'Backup prevalence
    With Range("F30:G35")
        For i = 1 To UBound(backups, 1)
            For j = 1 To UBound(backups, 2)
                backups(i, j) = 100 * .Cells(i, j)
            Next
        Next
    End With
    backup_time_delay = Range("J30")
    'The following are used later to calculate the number of backups
    backup_boundary1_LMW = backups(1, 1)
    backup_boundary2_LMW = backups(1, 1) + backups(2, 1)
    backup_boundary3_LMW = backups(1, 1) + backups(2, 1) + backups(3, 1)
    backup_boundary4_LMW = backups(1, 1) + backups(2, 1) + backups(3, 1) + backups(4, 1)
    backup_boundary5_LMW = backups(1, 1) + backups(2, 1) + backups(3, 1) + backups(4, 1) + backups(5, 1)
    backup_boundary6_LMW = backups(1, 1) + backups(2, 1) + backups(3, 1) + backups(4, 1) + backups(5, 1) + backups(6, 1)
    backup_boundary1_Bio = backups(1, 2)
    backup_boundary2_Bio = backups(1, 2) + backups(2, 2)
    backup_boundary3_Bio = backups(1, 2) + backups(2, 2) + backups(3, 2)
    backup_boundary4_Bio = backups(1, 2) + backups(2, 2) + backups(3, 2) + backups(4, 2)
    backup_boundary5_Bio = backups(1, 2) + backups(2, 2) + backups(3, 2) + backups(4, 2) + backups(5, 2)
    backup_boundary6_Bio = backups(1, 2) + backups(2, 2) + backups(3, 2) + backups(4, 2) + backups(5, 2) + backups(6, 2)

    'in-licensed projects
    ReDim inlicensed_projects(phases, forecast_years, assumption_sets - 2)  'dimension 1=phases, 2=year, 3=assumption set (LMW-POC and
Bio-POC only)
    With Worksheets("User Input").Range("D17:N20")          'LMW-POC
        For i = 1 To 4
            For j = 1 To UBound(inlicensed_projects, 2)
                Select Case phases
                Case 4
                    inlicensed_projects(i, j, 1) = .Cells(i, j)
                Case 6
                    inlicensed_projects(i + 2, j, 1) = .Cells(i, j)
                End Select
            Next
        Next
    End With
    With Worksheets("User Input").Range("D23:N26")        'Bio-POC
        For i = 1 To 4
            For j = 1 To UBound(inlicensed_projects, 2)
                Select Case phases
                Case 4
```

```
                inlicensed_projects(i, j, 2) = .Cells(i, j)
            Case 6
                inlicensed_projects(i + 2, j, 2) = .Cells(i, j)
            End Select
        Next
    Next
End With

ReDim transition_rates(phases, assumption_sets) As Variant          'dimension 1=phases, 2=assumption set
ReDim durations(phases, assumption_sets) As Variant                 'dimension 1=phases, 2=assumption set
End Sub

Sub SetupPipelineArrays()
ReDim pipeline_summary(phases, forecast_years + 1)
ReDim PIE_summary(forecast_years + 1)
ReDim percent_biologics(forecast_years + 1)
ReDim pipeline_full(phases, 200, assumption_sets)               'Note: 200 chosen as arbitrary max project number. Avoids issues
with array resizing compared to earlier model iterations
ReDim entry_class(phases, assumption_sets)
entry_class = current_projects
ReDim new_projects_pending_termination(phases, 10, assumption_sets)
ReDim old_projects_pending_termination(phases, 10, assumption_sets)

'The following randomly allocates current projects in each phase to different percentages of completion
Dim percentage_done As Integer
For i = 1 To phases
    For k = 1 To assumption_sets
        For j = 1 To current_projects(i, k)
            percentage_done = Int((100 * Rnd) + 1)  'generates a random number between 1 and 100
            pipeline_full(i, j, k) = percentage_done
        Next
    Next
Next

'The following populates pipeline_summary by summing the projects in pipeline_full
For i = 1 To phases
    temp_project_count = 0
    For k = 1 To assumption_sets
        j = 1
        Do While (Not IsEmpty(pipeline_full(i, j, k)))
            j = j + 1
        Loop
        temp_project_count = temp_project_count + j - 1
    Next
    pipeline_summary(i, step_counter + 1) = temp_project_count
Next
End Sub

Sub SetupForecasts()
   'This subroutine creates arrays to hold the forecasts. In addition, it initializes forecast cells to "0", since Crystal Ball does
not recognize blank forecast cells.
```

```
    ReDim forecasts_NME(5, UBound(pipeline_summary, 2))
    ReDim forecasts_readouts(4, UBound(pipeline_summary, 2))
    ReDim forecasts_transitions(phases - 1, UBound(pipeline_summary, 2))
End Sub

Sub ProgressProjects()
    TerminateProjects
    ReDim temp_pipeline_full(phases + 1, UBound(pipeline_full, 2), assumption_sets)         'erases temp_pipeline_full to prepare for
project progression
    ReDim entry_class(phases, assumption_sets) As Variant                                   'clears entry class to prepare for
tally (for next year's entry classes)
    'The following steps through all elements in pipeline_full, pulls a phase duration to determine how much work can be accomplished
in a year,
        'compares this against the amount of work left to complete phase, and transitions the project if appropriate. The code re-
checks after each
        'transition in case multiple transitions are needed for a given year.
    For i = 1 To phases
        For k = 1 To assumption_sets
            j = 1
            Do While (Not IsEmpty(pipeline_full(i, j, k)))
                'load new duration assumption for each project
                durations(i, k) = GetDuration(durations_assumptions(i, 1, k), durations_assumptions(i, 2, k), durations_assumptions(i,
3, k))
                row_counter = 0
                percent_year_used = 0
                percent_phase_remaining = 100 - pipeline_full(i, j, k)
                'scale duration to a percentage of phase that can be completed in 1 year
                percent_phase_accomplished = Round(100 * (1 / durations(i, k)))
                'compare percentage accomplished against percentage remaining to finish phase
                CheckIfPhaseComplete
                j = j + 1
            Loop
        Next
    Next

    'This section adds incoming projects started this year.
    For k = 1 To assumption_sets - 2
        transition_rates(1, k) = GetTransitionRate(transition_rates_assumptions(1, 1, k), transition_rates_assumptions(1, 2, k))
        temp_incoming_after_termination = Round(incoming_projects_calculated(step_counter, k) * transition_rates(1, k))
'terminates percentage of incoming projects; for simplicity, chose to not use more complicated tracking of "pending termination"
projects for future years
        temp_incoming_projects = temp_incoming_after_termination
        j = 1
        Do Until temp_incoming_projects = 0
            quarter_incoming = Application.WorksheetFunction.min(temp_incoming_projects, Application.WorksheetFunction.max(1,
Round(temp_incoming_after_termination / 4)))
            temp_incoming_projects = temp_incoming_projects - quarter_incoming
            If j = 4 Then
                quarter_incoming = quarter_incoming + temp_incoming_projects 'if rounding leaves any projects left to add during the
4th quarter, then add
                temp_incoming_projects = 0
```

```
            End If
            incoming_year_remaining = 1 - (j / 4)
            incoming_year_used = 1 - incoming_year_remaining
            For i = 1 To quarter_incoming
                incoming_row_counter = 0
                durations(1, k) = GetDuration(durations_assumptions(1, 1, k), durations_assumptions(1, 2, k), durations_assumptions(1,
3, k))
                incoming_progress = Application.WorksheetFunction.max(1, Round(incoming_year_remaining * (100 * (1 / durations(1,
k)))))        'Note: pipeline_full can't handle 0's
                temp_incoming_year_used = incoming_year_used
                temp_incoming_year_remaining = incoming_year_remaining
Line1:          If incoming_progress >= 100 Then
                    temp_incoming_year_used = WorksheetFunction.min(1, temp_incoming_year_used + ((1 - temp_incoming_year_used) * (100
/ incoming_progress)))
                    temp_incoming_year_remaining = 1 - temp_incoming_year_used
                    incoming_row_counter = incoming_row_counter + 1
                    incoming_progress = 1
                    entry_class(1 + incoming_row_counter, k) = entry_class(1 + incoming_row_counter, k) + 1
                    forecasts_transitions((1 + incoming_row_counter - 1), step_counter) = forecasts_transitions((1 +
incoming_row_counter - 1), step_counter) + 1
                    durations(1 + incoming_row_counter, k) = GetDuration(durations_assumptions(1 + incoming_row_counter, 1, k),
durations_assumptions(1 + incoming_row_counter, 2, k), durations_assumptions(1 + incoming_row_counter, 3, k))
                    incoming_progress = Application.WorksheetFunction.max(1, Round(temp_incoming_year_remaining * (100 * (1 /
durations(1 + incoming_row_counter, k)))))
                    GoTo Line1
                End If
                temp_pipeline_full(1 + incoming_row_counter, NextOpenElement(1 + incoming_row_counter, k, temp_pipeline_full), k) =
incoming_progress
            Next
        j = j + 1
        Loop
    Next

    'This section adds PIEs that will launched next year
    For k = 3 To 4
        PIE_starting_col = NextOpenElement(phases, k, temp_pipeline_full)
        For j = 1 To PIE_tracker(step_counter, 1, k - 2)
            temp_pipeline_full(phases, PIE_starting_col + j - 1, k) = PIE_tracker(step_counter, j + 1, k - 2)
        Next
        entry_class(phases, k) = entry_class(phases, k) + PIE_tracker(step_counter, 1, k - 2)
    Next

    'This section adds backups that will launch next year
    For k = 1 To 2
        backup_starting_col = NextOpenElement(phases - 2, k, temp_pipeline_full)
        For j = 1 To backup_tracker(step_counter, 1, k)
            temp_pipeline_full(phases - 2, backup_starting_col + j - 1, k) = backup_tracker(step_counter, j + 1, k)
        Next
        entry_class(phases - 2, k) = entry_class(phases - 2, k) + backup_tracker(step_counter, 1, k)
    Next
```

```
    'This section adds in-licensed projects
    For i = 1 To phases
        For k = 1 To assumption_sets - 2
            If inlicensed_projects(i, step_counter, k) = 0 Then GoTo Line2       'Note: this line avoids unnecessary calling of
NextOpenElement function if there are no in-licensed projects; cuts to next value in For loop
            inlicense_starting_col = NextOpenElement(i, k, temp_pipeline_full)
            For j = 1 To inlicensed_projects(i, step_counter, k)
                temp_pipeline_full(i, inlicense_starting_col + j - 1, k) = Int((100 * Rnd + 1))
            Next
            entry_class(i, k) = entry_class(i, k) + inlicensed_projects(i, step_counter, k)
Line2:  Next
    Next

    'copy temp_pipeline_full back into pipeline_full
    pipeline_full = temp_pipeline_full
    Erase temp_pipeline_full

End Sub

Sub TerminateProjects()
    'The following calculates number of projects that do not make it to the next phase
    ReDim projects_to_terminate(phases, assumption_sets) As Variant
    ReDim temp_projects_to_terminate(phases, assumption_sets) As Variant

    For i = 1 To UBound(projects_to_terminate)
        For j = 1 To assumption_sets
            transition_rates(i, j) = GetTransitionRate(transition_rates_assumptions(i, 1, j), transition_rates_assumptions(i, 2, j))
            projects_to_terminate(i, j) = Round(entry_class(i, j) * (1 - transition_rates(i, j)))
            temp_projects_to_terminate(i, j) = projects_to_terminate(i, j)      'used later in subroutine
        Next
    Next

    'The following maintains an array of projects that will ultimately by terminated. This provides for an accurate project count in
        'pipeline summary even though the model technically deletes projects from an entry class (in their first year) if they
ultimately will not transition.
    For i = 1 To UBound(new_projects_pending_termination)
        For k = 1 To assumption_sets
            j = 1
            Do Until temp_projects_to_terminate(i, k) = 0
                new_projects_pending_termination(i, j, k) = old_projects_pending_termination(i, j, k) +
Application.WorksheetFunction.min(temp_projects_to_terminate(i, k), Round(projects_to_terminate(i, k) / durations_assumptions(i, 2,
k)))
                temp_projects_to_terminate(i, k) = temp_projects_to_terminate(i, k) - (new_projects_pending_termination(i, j, k) -
old_projects_pending_termination(i, j, k))
                j = j + 1
            Loop
        Next
    Next

    'The line below records negative Clinical readouts for the current forecast year
```

```
    forecasts_readouts(4, step_counter) = new_projects_pending_termination(phases, 1, 1) + new_projects_pending_termination(phases, 1,
2) + new_projects_pending_termination(phases, 1, 3) + new_projects_pending_termination(phases, 1, 4)

    'The following deletes from pipeline full all projects that will not eventually make it to the next stage
    ReDim temp_pipeline_full(phases + 1, UBound(pipeline_full, 2), assumption_sets)
    For i = 1 To phases
        For k = 1 To assumption_sets
            j = 1
            Do While (Not IsEmpty(pipeline_full(i, j + projects_to_terminate(i, k), k)))
                temp_pipeline_full(i, j, k) = pipeline_full(i, j + projects_to_terminate(i, k), k)
                j = j + 1
            Loop
        Next
    Next
    pipeline_full = temp_pipeline_full

    'The following prepares the projects_pending_termination arrays for the next year
    For i = 1 To UBound(new_projects_pending_termination)
        For j = 1 To UBound(new_projects_pending_termination, 2) - 1
            For k = 1 To assumption_sets
                old_projects_pending_termination(i, j, k) = new_projects_pending_termination(i, j + 1, k)
            Next
        Next
    Next
End Sub

Sub CheckIfPhaseComplete()
    If percent_phase_accomplished >= percent_phase_remaining Then          'calc %yr used to complete phase and transition
        percent_year_used = WorksheetFunction.min(1, percent_year_used + ((1 - percent_year_used) * (percent_phase_remaining /
percent_phase_accomplished)))
        TransitionProject
    Else                                                                   'add year's accomplishment to project
        If row_counter = 0 Then
            temp_pipeline_full(i, NextOpenElement(i, k, temp_pipeline_full), k) = pipeline_full(i, j, k) + percent_phase_accomplished
        Else
            temp_pipeline_full(i + row_counter, NextOpenElement(i + row_counter, k, temp_pipeline_full), k) =
percent_phase_accomplished
        End If
    End If
End Sub

Sub TransitionProject()
    percent_phase_remaining = 100       'after transition, next phase is just being started
    percent_phase_accomplished = 1
    row_counter = row_counter + 1       'prepares for transition in array
    'The following checks for a positive readout; if yes, launch PIEs (if necessary), record for forecasts, and exit project
    If i + row_counter = phases + 1 Then
        If k = 1 Or k = 2 Then
            PIE_calculator          'For any POC readout, calls a subroutine that determines whether a PIE is launched
            forecasts_readouts(2, step_counter) = forecasts_readouts(2, step_counter) + 1                          'counts POCs
        End If
```

```
        If k = 3 Or k = 4 Then forecasts_readouts(3, step_counter) = forecasts_readouts(3, step_counter) + 1        'counts PIEs
        Exit Sub    'exits since this project has completed its readout and needs no more transitioning
    End If
    If i + row_counter = phases - 2 Then Backup_calculator 'For any transition into CSP, calls a subroutine that determines whether a
backup is launched
    forecasts_transitions((i + row_counter - 1), step_counter) = forecasts_transitions((i + row_counter - 1), step_counter) + 1
'counts transitions each year
    entry_class(i + row_counter, k) = entry_class(i + row_counter, k) + 1
    If percent_year_used < 1 Then      'if part of year remains, progress next phase of project
        'pulls a duration for the following phase to progress the project for the time left in the year
        'percent_phase_accomplished scaled by how much of year if left
        durations(i + row_counter, k) = GetDuration(durations_assumptions(i + row_counter, 1, k), durations_assumptions(i +
row_counter, 2, k), durations_assumptions(i + row_counter, 3, k))
        percent_phase_accomplished = Round((1 - percent_year_used) * (100 * (1 / durations(i + row_counter, k))))
    End If
    CheckIfPhaseComplete
End Sub

Sub PIE_calculator()
    'This subroutine determines whether a PIE is launched and in what proportion; note that because only the first "true" case is
executed, okay to use successive upper-bounds
    PIE_year_index = 0
    last_PIE_year = 0
    PIE_year_used = percent_year_used
    Randomize
    random_number = Int((100 * Rnd) + 1)
    Select Case random_number
        Case Is < PIE_boundary1                'no PIEs
            PIE_number = 0
        Case Is < PIE_boundary2                '1 PIE
            PIE_number = 1
        Case Is < PIE_boundary3                '2 PIEs
            PIE_number = 2
        Case Is < PIE_boundary4                '3 PIEs
            PIE_number = 3
        Case Is < PIE_boundary5                '4 PIEs
            PIE_number = 4
        Case Is <= PIE_boundary6               '5 PIEs
            PIE_number = 5
        Case Else
            Debug.Print "not between 0 and PIE_boundary 6"
    End Select

    PIE_counter = 1
    Do While (PIE_counter < PIE_number + 1)
        PIE_total_delay = PIE_year_used + PIE_time_delay
        PIE_year_index = WorksheetFunction.RoundDown(PIE_total_delay, 0)          'determines how many years ahead the next PIE will
launch; 0 = the current progression year
        PIE_year_remaining = 1 - (PIE_total_delay - PIE_year_index)               'determines what percentage of a year is left
during whatever year the PIE launches
```

```
        PIE_year_used = 1 - PIE_year_remaining                                  'sets up PIE_year_used for next PIE by knowing how
much of year has passed when PIE is launched
        durations(phases, k) = GetDuration(durations_assumptions(phases, 1, k), durations_assumptions(phases, 2, k),
durations_assumptions(phases, 3, k))
        PIE_progress = Round(PIE_year_remaining * (100 * (1 / durations(phases, k))))
        If last_PIE_year + PIE_year_index < 1 Then                              'if the PIE launches during the current progression year,
then check to see if it terminates. If not enter straight into pipeline_full
            transition_rates(phases, k) = GetTransitionRate(transition_rates_assumptions(phases, 1, k),
transition_rates_assumptions(phases, 2, k))
            Randomize
            If Rnd > transition_rates(phases, k) Then GoTo Line1            'single-project check to see if the PIE will be a negative
readout
            If PIE_progress >= 100 Then
                forecasts_readouts(3, step_counter) = forecasts_readouts(3, step_counter) + 1
            Else
                temp_pipeline_full(phases, NextOpenElement(phases, k + 2, temp_pipeline_full), k + 2) = PIE_progress
            End If
        ElseIf step_counter + last_PIE_year + PIE_year_index - 1 <= forecast_years Then                 'if the PIE launches during
subsequent years within the forecast range, must track until model reaches this point
            If PIE_progress >= 100 Then PIE_progress = 100              'Progress code can't handle a value in pipeline full>100, but
this makes it transition immediately
            PIE_tracker(step_counter + last_PIE_year + PIE_year_index - 1, 1, k) = PIE_tracker(step_counter + last_PIE_year +
PIE_year_index - 1, 1, k) + 1 'adds one to the first element, which tracks the total
            PIE_tracker(step_counter + last_PIE_year + PIE_year_index - 1, 1 + PIE_tracker(step_counter + last_PIE_year +
PIE_year_index - 1, 1, k), k) = PIE_progress 'uses the total to know which element to add the actual progress value to
        End If
Line1:  last_PIE_year = last_PIE_year + PIE_year_index              'Note: when PIE_year_index=0, last_PIE_year stays the same; when >
0, it increases to mark the year of the last PIE
        PIE_counter = PIE_counter + 1
        Loop
End Sub

Sub Backup_calculator()
    'This subroutine determines whether a backup is launched and in what proportion; note that because only the first "true" case is
executed, okay to use successive upper-bounds
    backup_year_index = 0
    last_backup_year = 0
    backup_year_used = percent_year_used
    Randomize
    random_number = Int((100 * Rnd) + 1)
    If k = 1 Then
        Select Case random_number
            Case Is < backup_boundary1_LMW                  'no backups
                backup_number = 0
            Case Is < backup_boundary2_LMW                  '1 backup
                backup_number = 1
            Case Is < backup_boundary3_LMW                  '2 backups
                backup_number = 2
            Case Is < backup_boundary4_LMW                  '3 backups
                backup_number = 3
            Case Is < backup_boundary5_LMW                  '4 backups
```

```
            backup_number = 4
        Case Is <= backup_boundary6_LMW                '5 backups
            backup_number = 5
        Case Else
            Debug.Print "not between 0 and backup_boundary 6"
    End Select
ElseIf k = 2 Then
    Select Case random_number
        Case Is < backup_boundary1_Bio                 'no backups
            backup_number = 0
        Case Is < backup_boundary2_Bio                 '1 backup
            backup_number = 1
        Case Is < backup_boundary3_Bio                 '2 backups
            backup_number = 2
        Case Is < backup_boundary4_Bio                 '3 backups
            backup_number = 3
        Case Is < backup_boundary5_Bio                 '4 backups
            backup_number = 4
        Case Is <= backup_boundary6_Bio                '5 backups
            backup_number = 5
        Case Else
            Debug.Print "not between 0 and backup_boundary 6"
    End Select
Else
    MsgBox ("There is a problem with the backup code")
End If

backup_counter = 1
Do While (backup_counter < backup_number + 1) ' Determine if rest is needed - And (step_counter + backup_year_index - 1 <=
forecast_years)
    backup_row_counter = 0
    backup_total_delay = backup_year_used + backup_time_delay
    backup_year_index = WorksheetFunction.RoundDown(backup_total_delay, 0)           'determines how many years ahead the next
backup will launch; 0 = the current progression year
    backup_year_remaining = 1 - (backup_total_delay - backup_year_index)                'determines what percentage of a year is
left during whatever year the PIE launches
    backup_year_used = 1 - backup_year_remaining                                      'sets up backup_year_used for next backup by
knowing how much of year has passed when PIE is launched
    durations(phases - 2, k) = GetDuration(durations_assumptions(phases - 2, 1, k), durations_assumptions(phases - 2, 2, k),
durations_assumptions(phases - 2, 3, k))
    backup_progress = Round(backup_year_remaining * (100 * (1 / durations(phases - 2, k))))
    If last_backup_year + backup_year_index < 1 Then                        'if the backup launches during the current progression
year, enter straight into pipeline_full
        transition_rates(phases - 2, k) = GetTransitionRate(transition_rates_assumptions(phases - 2, 1, k),
transition_rates_assumptions(phases - 2, 2, k))
        Randomize
        If Rnd > transition_rates(phases - 2, k) Then GoTo Line2         'single-project check to see if the backup will terminate
        temp_backup_year_used = backup_year_used            'Needed since backup_year_used can't be changed; need it to know when
to launch next backup
        temp_backup_year_remaining = backup_year_remaining
Line1:  If backup_progress >= 100 Then
```

```
                temp_backup_year_used = WorksheetFunction.min(1, temp_backup_year_used + ((1 - temp_backup_year_used) * (100 /
backup_progress)))
                temp_backup_year_remaining = 1 - temp_backup_year_used
                backup_row_counter = backup_row_counter + 1
                backup_progress = 1
                entry_class(1 + backup_row_counter, k) = entry_class(1 + backup_row_counter, k) + 1
                forecasts_transitions((i + row_counter + backup_row_counter - 1), step_counter) = forecasts_transitions((i +
row_counter + backup_row_counter - 1), step_counter) + 1
                durations(phases - 2 + backup_row_counter, k) = GetDuration(durations_assumptions(phases - 2 + backup_row_counter, 1,
k), durations_assumptions(phases - 2 + backup_row_counter, 2, k), durations_assumptions(phases - 2 + backup_row_counter, 3, k))
                backup_progress = Round(temp_backup_year_remaining * (100 * (1 / durations(phases - 2 + backup_row_counter, k))))
                GoTo Line1
            End If
                temp_pipeline_full(phases - 2 + backup_row_counter, NextOpenElement(phases - 2 + backup_row_counter, k,
temp_pipeline_full), k) = backup_progress

        ElseIf step_counter + last_backup_year + backup_year_index - 1 <= forecast_years Then               'if the backup launches
during subsequent years within the forecast range, must track until model reaches this point
            If backup_progress >= 100 Then backup_progress = 100        'Progress code can't handle a value in pipeline full>100, but
this makes it transition immediately
            backup_tracker(step_counter + last_backup_year + backup_year_index - 1, 1, k) = backup_tracker(step_counter +
last_backup_year + backup_year_index - 1, 1, k) + 1 'adds one to the first element, which tracks the total
            backup_tracker(step_counter + last_backup_year + backup_year_index - 1, 1 + backup_tracker(step_counter + last_backup_year
+ backup_year_index - 1, 1, k), k) = backup_progress 'uses the total to know which element to add the actual progress value to
        End If
Line2:  last_backup_year = last_backup_year + backup_year_index             'Note: when backup_year_index=0, last_backup_year stays
the same; when > 0, it increases to mark the year of the last PIE
        backup_counter = backup_counter + 1
    Loop
End Sub

Sub UpdatePipelineSummary()
   'The following updates the pipeline_summary array based on the status of pipeline_full and project pending termination but still in
pipeline for current year
    temp_project_count_LMW = 0
    temp_project_count_Bio = 0
    For i = 1 To phases
        temp_project_count = 0
        For k = 1 To assumption_sets
            j = 1
            Do While (Not IsEmpty(pipeline_full(i, j, k)))
                j = j + 1
            Loop
            temp_project_count = temp_project_count + j - 1                                    'counts total projects for
pipeline_summary
            If k = 1 Or k = 3 Then temp_project_count_LMW = temp_project_count_LMW + j - 1          'counts LMWs for LMW/Bio ratio
            If k = 2 Or k = 4 Then temp_project_count_Bio = temp_project_count_Bio + j - 1          'counts Bios for LMW/Bio ratio
            If i = phases And (k = 3 Or k = 4) Then PIE_summary(step_counter + 1) = PIE_summary(step_counter + 1) + j - 1       'Note:
It is vital to ensure that this subroutine is only called once, for each forecast year. Otherwise, this value will be multiplied.
        Next
        For k = 1 To UBound(old_projects_pending_termination, 2)
```

```
            total_pending_termination = total_pending_termination + (old_projects_pending_termination(i, k, 1) +
old_projects_pending_termination(i, k, 2) + old_projects_pending_termination(i, k, 3) + old_projects_pending_termination(i, k, 4))
        Next
        pipeline_summary(i, step_counter + 1) = temp_project_count + total_pending_termination
    Next
    If (temp_project_count_LMW + temp_project_count_Bio <> 0) Then percent_biologics(step_counter + 1) = temp_project_count_Bio /
(temp_project_count_LMW + temp_project_count_Bio)
End Sub

Sub UpdateForecasts()
    UpdatePipelineSummary
    'NME Portfolio
    For i = 1 To UBound(pipeline_summary, 2)
        forecasts_NME(1, i) = pipeline_summary(phases - 2, i) + pipeline_summary(phases - 1, i) + (pipeline_summary(phases, i) -
PIE_summary(i))
    Next
    'Clinical Portfolio
    For i = 1 To UBound(pipeline_summary, 2)
        forecasts_NME(2, i) = pipeline_summary(phases, i)
    Next
    'sPOC Portfolio
    For i = 1 To UBound(pipeline_summary, 2)
        forecasts_NME(3, i) = pipeline_summary(phases - 1, i)
    Next
    'CSP Portfolio
    For i = 1 To UBound(pipeline_summary, 2)
        forecasts_NME(4, i) = pipeline_summary(phases - 2, i)
    Next
    'Percent biologics in NME Portfolio
    For i = 1 To UBound(pipeline_summary, 2)
        forecasts_NME(5, i) = percent_biologics(i)
    Next

    'Note: Readouts and Transitions updated during project progression
    'Totals positive POC and PIE readouts
    For i = 1 To UBound(pipeline_summary, 2)
        forecasts_readouts(1, i) = forecasts_readouts(2, i) + forecasts_readouts(3, i)
    Next

End Sub

Sub UpdatePipelineDisplay()
    'UpdatePipelineSummary                      'Note: This is currently called as part fo UpdateForecasts, which always precedes this
subroutine.
    'UpdatePipelineFullDisplay                  'toggle on for debugging pipeline_full code in worksheets
    Worksheets("Pipeline Summary").UsedRange.Clear  'clears worksheets showing pipeline array values
    With Worksheets("Pipeline Summary")
        .Range(.Cells(1, 1), .Cells(UBound(pipeline_summary, 1), UBound(pipeline_summary, 2))).Value = pipeline_summary
    End With
    Worksheets("User Dashboard").Range("D18:O23") = Worksheets("Pipeline Summary").Range("A1:L6").Value
```

```
    'These With statements update forecasts
    Worksheets("User Dashboard").Activate
    'NME Portfolio
    With Range("D26:O30")
        For i = 1 To 5
            For j = 1 To forecast_years + 1
                .Cells(i, j) = forecasts_NME(i, j)
            Next
        Next
    End With
    'Readouts
    With Range("D33:O36")
        For i = 1 To 4
            For j = 1 To forecast_years + 1
                .Cells(i, j) = forecasts_readouts(i, j)
            Next
        Next
    End With
    'Transitions
    With Range("D39:O43")
        For i = 1 To phases - 1
            For j = 1 To forecast_years + 1
                .Cells(i, j) = forecasts_transitions(i, j)
            Next
        Next
    End With
End Sub

Sub UpdatePipelineFullDisplay()
    Worksheets("Pipeline Full LMW-POC").UsedRange.Clear
    Worksheets("Pipeline Full Bio-POC").UsedRange.Clear
    Worksheets("Pipeline Full LMW-PIE").UsedRange.Clear
    Worksheets("Pipeline Full Bio-PIE").UsedRange.Clear

    'These With statements load the pipeline arrays into two worksheets for visual reference

    With Worksheets("Pipeline Full LMW-POC")
        For i = 1 To UBound(pipeline_full, 1)
            j = 1
            Do While (Not IsEmpty(pipeline_full(i, j, 1)))
                .Cells(i, j) = pipeline_full(i, j, 1)
                j = j + 1
            Loop
        Next
    End With

    With Worksheets("Pipeline Full Bio-POC")
        For i = 1 To UBound(pipeline_full, 1)
            j = 1
            Do While (Not IsEmpty(pipeline_full(i, j, 2)))
                .Cells(i, j) = pipeline_full(i, j, 2)
```

```
            j = j + 1
        Loop
    Next
End With

With Worksheets("Pipeline Full LMW-PIE")
    For i = 1 To UBound(pipeline_full, 1)
        j = 1
        Do While (Not IsEmpty(pipeline_full(i, j, 3)))
            .Cells(i, j) = pipeline_full(i, j, 3)
            j = j + 1
        Loop
    Next
End With

With Worksheets("Pipeline Full Bio-PIE")
    For i = 1 To UBound(pipeline_full, 1)
        j = 1
        Do While (Not IsEmpty(pipeline_full(i, j, 4)))
            .Cells(i, j) = pipeline_full(i, j, 4)
            j = j + 1
        Loop
    Next
End With
End Sub

Sub LoadDefaults()
    Worksheets("User Input").Range("C4:D10") = Worksheets("Default Values").Range("C4:D10").Value     'current projects
    Worksheets("User Input").Range("K4:L8") = Worksheets("Default Values").Range("K4:L8").Value       'current projects
    Worksheets("User Input").Range("F4:H14") = Worksheets("Default Values").Range("F4:H14").Value   'incoming projects
    Worksheets("User Input").Range("N4:P14") = Worksheets("Default Values").Range("N4:P14").Value     'incoming projects
    Worksheets("User Input").Range("D17:N20") = Worksheets("Default Values").Range("D17:N20").Value   'inlicensed LMW
    Worksheets("User Input").Range("D23:N26") = Worksheets("Default Values").Range("D23:N26").Value   'inlicensed Bio
    Worksheets("User Input").Range("C29:C34") = Worksheets("Default Values").Range("C29:C34").Value   'PIE ratios
    Worksheets("User Input").Range("F30:G35") = Worksheets("Default Values").Range("F30:G35").Value   'backup ratios
    Worksheets("User Input").Range("J29:J30") = Worksheets("Default Values").Range("J29:J30").Value   'PIE and backup time delays
    Worksheets("User Input").Range("C41:J46") = Worksheets("Default Values").Range("C41:J46").Value   'transition rates
    Worksheets("User Input").Range("C52:N57") = Worksheets("Default Values").Range("C52:N57").Value   'durations

End Sub

Sub ResetForecasts()
'This subroutine sets up forecast cells in Crystal Ball, in case they accidentally get corrupted.
    If cb.CBLoaded() = False Then cb.Startup           'checks to see if Crystal Ball is open; if not, opens it
    'NME Portfolio
    With Worksheets("User Dashboard").Range("D26:O26")
        For i = 1 To 12
            .Cells(1, i).Select
            .Cells(1, i).Value = 0                  'Note: "0" used to initialize cell so that CB can assign a forecast
            cb.DefineForeND "Year " & ActiveCell.Offset(-1, 0).Text & " NME Portfolio", "Projects", False, False
        Next
```

```
    End With
    'Clinical Portfolio
    With Worksheets("User Dashboard").Range("D27:O27")
        For i = 1 To 12
            .Cells(1, i).Select
            .Cells(1, i).Value = 0
            cb.DefineForeND "Year " & ActiveCell.Offset(-2, 0).Text & " Clinical Portfolio", "Projects", False, False
        Next
    End With
    'sPOC Portfolio
    With Worksheets("User Dashboard").Range("D28:O28")
        For i = 1 To 12
            .Cells(1, i).Select
            .Cells(1, i).Value = 0
            cb.DefineForeND "Year " & ActiveCell.Offset(-3, 0).Text & " sPOC Portfolio", "Projects", False, False
        Next
    End With
     'CSP Portfolio
    With Worksheets("User Dashboard").Range("D29:O29")
        For i = 1 To 12
            .Cells(1, i).Select
            .Cells(1, i).Value = 0
            cb.DefineForeND "Year " & ActiveCell.Offset(-4, 0).Text & " CSP Portfolio", "Projects", False, False
        Next
    End With
     '% Biologics
    With Worksheets("User Dashboard").Range("D30:O30")
        For i = 1 To 12
            .Cells(1, i).Select
            .Cells(1, i).Value = 0
            cb.DefineForeND "Year " & ActiveCell.Offset(-5, 0).Text & " % Biologics", "Projects", False, False
        Next
    End With
    'Readouts
    With Worksheets("User Dashboard").Range("D33:O36")
        For i = 1 To 4
            For j = 1 To 12
                .Cells(i, j).Select
                .Cells(i, j).Value = 0
                cb.DefineForeND "Year " & ActiveCell.Offset(-(i + 7), 0) & ActiveCell.Offset(0, -(j + 1)).Text & " Clinical Readouts",
"Projects", False, False
            Next
        Next
    End With
    'Transitions Portfolio
    With Worksheets("User Dashboard").Range("D39:O43")
        For i = 1 To 5
            For j = 1 To 12
                .Cells(i, j).Select
                .Cells(i, j).Value = 0
```

```
                cb.DefineForeND "Year " & ActiveCell.Offset(-(i + 13), 0) & ActiveCell.Offset(0, -(j + 1)).Text & " Transitions",
"Projects", False, False
            Next
        Next
    End With
End Sub
```

7.3.3 Module 2: Public functions

```
'This function pulls a new transition rate based on the phase and assumption set being assessed.
Public Function GetTransitionRate(mean_tr, stdev_tr) As Double
    Randomize   'randomize seed for Rnd function below based on system clock
5   GetTransitionRate = Application.WorksheetFunction.NormInv(Rnd, mean_tr, stdev_tr)
    If GetTransitionRate <= 0.2 Or GetTransitionRate > 1 Then GoTo 5       'truncates the normal distribution at .2 and 1
End Function

'This function pulls a new duration based on the phase and assumption set being assessed.
Public Function GetDuration(min, likeliest, max) As Double
    Randomize   'randomize seed for Rnd function below based on system clock
    a = min
    M = likeliest
    b = max
    'Note: below formulas taken from Ron Davis article in "Informs: Transactions on Education"
    alpha = ((2 * (b + 4 * M - 5 * a)) / (3 * (b - a))) * (1 + (4 * (((M - a) * (b - M)) / ((b - a) ^ 2))))
    beta = ((2 * (5 * b - 4 * M - a)) / (3 * (b - a))) * (1 + (4 * (((M - a) * (b - M)) / ((b - a) ^ 2))))
    GetDuration = Application.WorksheetFunction.BetaInv(Rnd, alpha, beta, a, b)
End Function

'This function is used to find the next unused element in the pipeline_full arrays
Public Function NextOpenElement(i_phase, k_assumption_set, temp_pipeline_full)
    NextOpenElement = 1
    Do Until temp_pipeline_full(i_phase, NextOpenElement, k_assumption_set) = 0 Or NextOpenElement = UBound(temp_pipeline_full, 2)
'finds first 0 or upper bound of array
        NextOpenElement = NextOpenElement + 1
        If NextOpenElement = UBound(temp_pipeline_full, 2) Then MsgBox ("Problem: you've reached the end of pipeline_full-fix the
code")    'if above found UB, adds a column to array
    Loop
End Function
```