



**OPERATIONS RESEARCH CENTER**  
United States Military Academy  
West Point, New York 10996

June 2020

# Leveraging Multi-Tier Workforce and Complexity Modeling Technical Report

## Prepared By

MAJ JD Caddell

Department of Systems Engineering  
United States Military Academy

## Prepared For

Directorate for Analysis (DI)  
Defense Intelligence Agency  
200 MacDill Blvd  
Joint Base Anacostia-Bolling (JBAB)  
Washington, DC 20301



**DISTRIBUTION A.** Approved for public release; distribution is unlimited

**Report 2020-TBD**  
**DTIC: AXXXXXXX**

The views and opinions expressed or implied in this publication are solely those of the author(s) and should not be construed as policy or carrying the official sanction of the United States Military Academy, the Department of Army, the Department of Defense, or other agencies or departments of the US government.



## Abstract

Organizational change remains an active vein of research area for operations research. Many models attempt to capture how changes in a workforce structure, work processes, and skill composition interact to generate output. While many models attempt to conduct discrete optimizations, these efforts fail to explain how an organization, and its performance, might evolve or transition between the current and desired state. Furthermore, many models attempt to aggregate workers and work, oversimplifying the operational environment in which the organization operates. By leveraging system dynamics, this work reveals the benefits of disaggregating workers and work complexities to predict emergent behavior experienced by changes in organizational structure and composition. It also demonstrates a method to transform a system dynamics model into a web based Shiny application to enable client engagement.

**Key Words:** workforce modeling, system dynamics, shiny, r



This Page Intentionally Left Blank



---

## About Us

Established in 1988 inside the Department of Systems Engineering (DSE), the Operations Research Center (ORCEN) provides USMA with a dedicated analytical capability that routinely tackles problems of national significance for the Army and the Department of Defense (DoD). Leveraging the systems engineering and mathematical modeling skills of a small cadre of full time analysts, the ORCEN executes an annual research portfolio of six to eight reimbursable projects across varied domains, including (but not limited to) operations assessment, human resource management, geospatial modeling, maintenance process improvement, and force structure decisions.

Immaterial of the domain, when the ORCEN analyst's abilities and the project sponsor's needs properly align, history has shown that the results can be quite good. Specifically, since 2005 the ORCEN has won the Dr. Wilbur B. Payne Memorial Award for Excellence in Analysis four times – an annual award, given under the signature of the Secretary of the Army, which recognizes the best operations research work that makes a significant contribution to the force. In addition to providing invaluable insights to their sponsors, ORCEN projects enrich cadet education, professionally develop Operations Research / Systems Analysis (ORSA) faculty, integrate emerging technologies into the Academic Program, and sustain ties between the Academy and the Army.

Beyond executing year-long, faculty-led research projects, the ORCEN also manages DSE's Academic Individual Advanced Development (AIAD or summer internship) program, where cadets are placed inside Army, DoD, industry, and academic organizations around the world for three to four weeks between May and August. This program broadens the cadets' perspectives, and it provides an opportunity for them to apply what they have learned in the classroom to the real-world. Cadet interns are typically rising juniors or seniors majoring in one of DSE's four majors: Systems Engineering, Engineering Management, Systems Decision Sciences, or Operations Research, and they work on an appropriately scoped project under the mentorship of the sponsoring organization. Without exception, the benefits to both the cadets and the sponsors are substantial. As one organization recently put it, "Not only did the internship program allow cadets to learn about acquisition and the processes we manage, but it also provided new ideas, thoughts, and creative concepts for us."

For more information regarding the ORCEN's reimbursable research or AIAD programs please contact:



Operations Research Center  
Attn: LTC James H. Schreiner, PhD, PMP, CPEM, F.ASEM, Director  
Department of Systems Engineering  
Mahan Hall, Bldg. 752  
West Point, NY 10996  
[james.schreiner@westpoint.edu](mailto:james.schreiner@westpoint.edu)  
845-938-5672



This Page Intentionally Left Blank



---

## Table of Contents

Chapter 1 Introduction and Organization .....	1
1.1. Introduction .....	1
1.2. Project Overview .....	1
1.3. Organization .....	1
1.3.1. Chapter 2 Summary: Multi-Tier Workforce and Work Complexity Modeling .....	1
1.3.2. Chapter 3 Summary: Building Interactive System Dynamics Simulations in R .....	1
1.3.3. Chapter 4 Summary: How to Leverage the Workforce Application .....	2
Chapter 2 Multi-Tier Workforce and Work Complexity Modeling .....	3
2.1. Introduction .....	3
2.2. Literature Review .....	3
2.2.1. System Dynamics .....	3
2.2.2. System Dynamics and Workforce Modeling .....	4
2.3. Initial System Structure .....	5
2.3.1. Work/Rework Cycle .....	5
2.3.2. Workforce Creation .....	6
2.4. Expanding the Structure .....	7
2.4.1. Multi-Tiered Work Complexity .....	7
2.4.2. Multi-Tiered Workforce .....	12
2.4.3. Dynamic Work Assignment .....	15
2.4.4. Model Adjustment to Fit Differing Environments .....	16
2.5. Model Analysis .....	17
2.6. Limitations of Model .....	19
2.7. Conclusion and Future Work for Model .....	20
Chapter 3 Building Interactive System Dynamics Simulations in R .....	21
3.1. Introduction .....	21
3.2. Literature Review .....	21
3.2.1. System Dynamics .....	21
3.2.2. R Background .....	22
3.2.3. System Dynamics in R .....	23
3.3. Methodology .....	23
3.3.1. Modeling in Vensim .....	23



---

3.3.2. Conversion from Vensim to R.....	24
3.3.3. Visualizing Model in R .....	26
3.3.4. Conversion from R to Shiny Application .....	27
3.4. Results.....	28
3.4.1. Model Manipulation and Data Extraction.....	28
3.4.2. Current Limitations with Method .....	29
3.5. Conclusion and Future Work for Method .....	30
Chapter 4 How to Leverage the Workforce Application.....	31
4.1. Introduction.....	31
4.2. Application Overview .....	31
4.2.1. Application Navigation .....	31
4.2.2. Model Investigation.....	31
4.2.2.1 Parameters.....	31
4.2.2.2 Visualizations .....	33
4.2.3. Model Assumptions .....	35
4.2.3.1 Changing Model Settings.....	35
4.2.3.2 Picking and Setting a Baseline .....	35
4.2.3.3 Saving Model Settings .....	35
4.2.3.4 Uploading Custom Model Settings/Resetting to Default.....	35
4.2.4. Data .....	36
4.3. Benchmarking.....	36
4.3.1. Workers.....	36
4.3.2. Work.....	36
4.4. Model Execution and Interpretation.....	37
4.4.1. Scenario Development .....	37
4.4.2. Setting Parameters .....	38
4.4.3. Over Capacity / Under Capacity .....	38
4.4.4. Work Prioritization.....	39
4.4.5. The Effect of Growing the Workforce .....	40
Chapter 5 Limitations and Future Work .....	42
5.1. Limitations .....	42
5.2. Future Work.....	42
Chapter 6 Conclusion.....	43



---

Appendix A References .....	44
Appendix B Workforce Model Equations .....	47



---

## List of Figures

Figure 2.1: Macro Level Causal Loop Diagram .....	5
Figure 2.2: Re-Work Generation Causal Loop Diagram .....	6
Figure 2.3: Personnel Causal Loop Diagram.....	7
Figure 2.4: Rework Cycle.....	8
Figure 2.5: Morale .....	9
Figure 2.6: Work Pressure .....	10
Figure 2.7: Team Instability.....	11
Figure 2.8: Single Tier Workforce Creation.....	12
Figure 2.9: Three-Tiered Workforce.....	14
Figure 2.10: Dynamic Work Assignment.....	15
Figure 2.11: Team Experience Level .....	16
Figure 2.12: All Work Clears and Reach Steady State in Scenario 1.....	17
Figure 2.13: Team 1 operates under increased pressure when compared to Team 2 in Scenario 1 .....	18
Figure 2.14: Workforce Break down for Scenario 2 .....	19
Figure 3.15: CRAN package releases (aggregated by year).....	22
Figure 3.16: Proposed Interactive System Dynamics Modeling Process in R.....	23
Figure 3.17: Project dynamics model adapted from previous work [5] [32].....	24
Figure 3.18: Vensim Table Lookup Function in R.....	25
Figure 3.19: R System Dynamics Model Benchmarking .....	26
Figure 3.20: Example Project Dynamics Output in R .....	27
Figure 3.21: Shiny Application – Model Investigation Panel.....	28
Figure 3.22: Shiny Application - Data Panel .....	29
Figure 4.23: Application Navigation Bar.....	31
Figure 4.24: Application Basic Parameters.....	32
Figure 4.25: Application Baseline Comparison .....	33
Figure 4.26: Application Advanced Parameters .....	33
Figure 4.27: Work and Team Function Visualization .....	34
Figure 4.28: Workforce Visualization .....	34
Figure 4.29: Application Model Assumptions.....	35
Figure 4.30: Application Data Page .....	36
Figure 4.31: Benchmarking Work .....	37



---

Figure 4.32: Simple Scenario 1.....	37
Figure 4.33: Scenario 1 Output – Starting Staff at 100 Projects/Year .....	38
Figure 4.34: Scenario 1 Output - Starting Staff at 180 Projects/Year .....	39
Figure 4.35: Scenario 1 Output - Starting Staff at 140 Projects/Year .....	40
Figure 4.36: Scenario 1 Output - Desired Staff at 140 Projects/Year - Work .....	41
Figure 4.37: Scenario 1 Output - Desired Staff at 140 Projects/Year – People.....	41



---

## Acronyms

DIA	Defense Intelligence Agency
DI	Directorate for Analysis
ADO	Analytic Development Office
SD	System Dynamics
DIAAE	Defense Intelligence All-source Analytic Enterprise
CRAN	Comprehensive R Archive Network
GUI	Graphical User Interface



---

# Chapter 1

## Introduction and Organization

### 1.1. Introduction

This report summarizes the work conducted by MAJ JD Caddell during the period 01 June 2019 through 30 June 2020 in support of the Directorate of Analysis within the Defense Intelligence Agency.

### 1.2. Project Overview

The Defense Intelligence Agency's (DIA) Directorate for Analysis (DI) supports warfighters, defense planners, and senior policymakers with integrated, all-source analysis. The DI Director also serves as the functional manager for analysis for the Defense Intelligence Analysis Program, providing broad oversight to a wide range of requirements and activities across the Defense Intelligence All-source Analytic Enterprise (DIAAE). Within DI, the Analytic Development Office (ADO) is responsible for analytic workforce development and shaping DIA's modernization efforts related to analytic capabilities. As collection systems and data continue to proliferate, the DI seeks to capitalize on this change in environment. As such, this project sought to develop a mechanism to examine varying organizational constructs and their ability to support anticipated analytic requirements.

### 1.3. Organization

The work done, although mutually supporting, naturally separated itself into three different categories. As such, this paper is organized similarly. The following paragraphs briefly summarize each chapter.

#### 1.3.1. Chapter 2 Summary: Multi-Tier Workforce and Work Complexity Modeling

Organizations often struggle to design and maintain adequate workforce structures to fill capacity requirements. System dynamics provides a mechanism to investigate this situation by modeling the accumulations, flows, and causal relationships in the system. This vein of research yielded several insights and structures to better understand concepts such as the effects of schedule pressure, the capability trap, and rework. However, organizations frequently request additional complexities that complicate the previous methods. For example, organizations contain several different types of workers. These workers may have different capacities and most likely completely different types of work. Some work, through either training requirements or complexity cannot be completed by every worker. This chapter addresses such a situation, building upon previous workforce models, by developing a three-tiered workforce structure that completes two separate types of work by leveraging a dynamic assignment mechanism for workers. This chapter also demonstrates how developing flexible parameters can help organizations leverage these simple models to explore both existing and envisioned work environments and their relative effectiveness with different workforce structures. By creating these additional flexible structures organizations address more nuanced workforce questions and anticipate the future effects of workforce policy on both people and work.

#### 1.3.2. Chapter 3 Summary: Building Interactive System Dynamics Simulations in R

System dynamics provides insights into the dynamic behavior of a system through modeling the accumulations, flows, and causal relationships in a system. This allows modelers to develop simulations



and models to understand the structure of a system, identify leverage points, and evaluate policy options for modifying the behavior of the system. An essential tool for system dynamics is the management flight simulator that enables business leaders and decision makers to interact with the model to evaluate policy options based on the simulated behavior. However, building and developing these flight simulators is time and resource intensive and can limit the ability of modelers to share their models with clients in an interactive manner. This chapter uses a classic system dynamics model to explore the possibility of building and simulating a system dynamics model in the R coding language. The methodology transforms a traditional system dynamics model from Vensim into R code and simulates the model in this language. Additionally, the chapter examines methods to display the model using a Shiny application with the R language. Ideally, this work will lower the threshold required to generate management flight simulators to provide insights for decision makers in a more timely, efficient manner.

### **1.3.3. Chapter 4 Summary: How to Leverage the Workforce Application**

This chapter provides an overview of the application developed for the effort, recommendations for work and worker benchmarking, and common model interpretations from sample simulation runs. The goal of this chapter is to act as a simple Ho-To guide for organizations attempting to leverage the current web-based system dynamics model.



---

## Chapter 2

# Multi-Tier Workforce and Work Complexity Modeling<sup>1</sup>

### 2.1. Introduction

To help a client organization explore and determine appropriate workforce structures for a pending reorganization and development of a new capacity, our team explored many modeling paradigms. Ultimately, System Dynamics (SD) proved the most suitable because of the client's concern about dynamics, how the system would behave across time, and the extensive previous work in project dynamics and workforce modeling. Previous SD work in this field focused primarily on aggregated workers and single work/task types. However, unique internal hiring processes, differences in work complexity, and concerns about retention warranted an adjusted approach. This chapter outlines the extensions taken to represent a multi-tiered work force and varying complexities of work. By including these representations, the client can better design, anticipate performance, and implement new workforce structures. The chapter starts out with a description of SD, overview of previous SD workforce models, outlines the general structures in a series of causal loop diagrams, explores the expanded structures and modeling choices, details the limitations of the approach, and then concludes with brief discussion of the work's relevance and future.

### 2.2. Literature Review

The literature review addresses SD as a general modeling technique and then focus workforce modeling more specifically.

#### 2.2.1. System Dynamics

System dynamics is a graphical and mathematical methodology to understand the dynamic behavior of systems using accumulations, flows, and causal relationships within a system. Practitioners believe that a system's structure generates its behavior and attempt to model the underlying structure of a system and models the behavior over time. Modelers seek to adjust individuals' mental models of the system and test potential policy alternatives. Forrester's [1] work describes a system as "a grouping of parts that operate together for a common purpose". He further classifies two types of systems: open systems, in which exogenous (external) variables affect the system, or closed systems where all variables are endogenous (internal) to the system [2]. Forrester famously presented an example system model of the world that described the interrelations between population, capital investment, geographical space, natural resources, pollution, and food production [3]. A system's underlying structure creates the dynamics of a system, or the behavior of a system over time, which are often complex and non-linear [2]. This complexity stems from feedback within the system, time delays between decisions and effects, and the learning process of the system [4]. These dynamic features of systems make them difficult to understand and identify the cause and effect relationships without an effective model of the system.

Forrester described the potential for system dynamics to help decision makers understand the implications of their policies and potentially identify and mitigate unintended consequences of their decisions [2]. Applications of System Dynamics have provided insights across several domains; including

---

<sup>1</sup> Parts of this chapter were submitted and accepted into the 38<sup>th</sup> International Conference of the System Dynamics Society (Caddell, J., Driscoll, P., Connell, T., & Enos, J. (2020). Multi-Tiered Workforce and Work Complexity Modeling. The 38th International Conference of the System Dynamics Society.).



government policy, corporate policy, healthcare, and commodity markets [3, 4]. Often Companies and consultants utilize System Dynamics in the development of corporate strategy, examining possible business decisions after a crisis or complex problem triggers shifts in business strategy. System Dynamics assists in determining the impetus of the crisis and recognizing potential consequences of possible decisions [4]. SD makes explicit causal relationships between variables to understand how the underlying structure and accumulations affect the behavior of the system over time.

### 2.2.2. System Dynamics and Workforce Modeling

At its core, this research aims to capture the basic interaction between organizational work that needs completed and the workforce available to complete the work. Lyneis and Ford [5] review and analyze existing research and structures surrounding the work / rework cycle initially introduced by Cooper [6]. The work / rework cycle introduced in these papers remains foundational in system dynamics project models and appears in a majority of project management focused SD models. Additionally, captured within these works are adverse “knock-on” effects generating added feedback. These knock-on effects affect work quality and in-turn increase error rates and decrease productivity [5]. These works remain fundamental in the application of SD to project and task management.

A large body of SD research has focused on the impacts of service quality, work backlog, and workforce management in the presence of work pressure and work intensity stress, changing morale, and changes in productivity [7, 8]. Further, focused research has investigated why so many projects focused on improving processes or systems fail. The “Capability Trap” shows that pressure to yield short term improvements often causes firms to focus on the short-term fixes neglecting longer term investment or system improvement [9, 10]. Though its origins are in an industrial or manufacturing sense, additional work has shown the capability trap can be applied to a broader context across social systems. The capability trap structure has been expanded from its original use to cover topics in humanitarian operation [11], Infrastructure maintenance [12], and foster care [13].

Lastly, the organizational context surrounding our client led the team to investigate the existing body of research addressing the use of SD in modeling workforce management in military or governmental organizations. Much of this research established multi-stage aging chains to evaluate retention policy and desired workforce levels. The preponderance of these models address policies surrounding retention, training bubbles, and the impacts of force-shaping policies (promotion rates, desired workforce size, recruitment policies) but do little to show the impacts of work capacity or productivity as a result of the policy decisions [14, 15, 16]. Further, little is mentioned in literature depicting the dynamics of a multitiered workforce interacting with a multitiered body of work.

The existing body of SD literature served as the foundation for much of the team’s work, however a few gaps were discovered. First, the nature of the client’s work requires that the body of work be separated into (at least) two different levels of complex. Additionally, when modeling the pool of workers available to complete a building backlog of work, there are three distinct skill levels associated with available workers. Additionally, the differing from Olivia and Sterman [8], the workers associated with this research gain experience and thus increase their productivity by successfully completing work, not simply after some assimilation time. The combination of these two produced an additional structure capturing the progression of workers across distinct skill levels. Lastly, as workers exit their roles either through attrition or progression, the organization must decide whether to promote from within or search for potential replacements external to the organization. As a consequence, our initial modeling effort contained basic decision rules governing the likelihood managers would promote or seek new hires. In exchanges with the client, it became clear that there exists a skill level at which the organization could not directly hire and fulfill

vacancies with an internal promotion. This places an added emphasis on the importance of internal promotion, workforce development, and the importance of fulfilling vacancies with highly skilled, low-density workers who are internal to the workforce.

### 2.3. Initial System Structure

In this section we outline the overall system structure at a high level of aggregation. This helps identify how the model fits together, the conceptual development of the model, and provides a framework with which model expansions can reference.

This section presents a series of causal loop diagrams that describe the structure of the system that generates the observed behavior of the system. This model depicts how employees, at various skill levels, gain experience through performing work as opposed to a set amount of time to gain experience as a worker. The first diagram presents a macro view of the system that links work and experience through productivity. The second diagram describes the basic re-work loop that provides a basis for the model to describe how errors introduce additional work in the system. The final diagram depicts the structure associated with the personnel system who come from a limited pool of talent, are onboarded to the organization, and eventually become experienced employees.

Figure 2.1 presents the macro level causal loop diagram for the system that consist of a major balancing loop and a major reinforcing loop. As *Work* increases, *Work Pressure* within the organization also increases as this creates a need to do the work. The increased in *Work Pressure* increases the *Productivity* of the organization which eventually reduces the amount of *Work* to close this balancing loop in the model. The reinforcing loop accounts for the introduction of errors and the *Experience* of the work force. As the *Productivity* increases, the amount of *Experience* gained also increases. A more experienced work force will reduce the *Error Rate* as the employees have a better understanding of how to accomplish the tasks. An increase in *Error Rate* also decreases the *Productivity* of the organization which then increases the amount of *Work*.

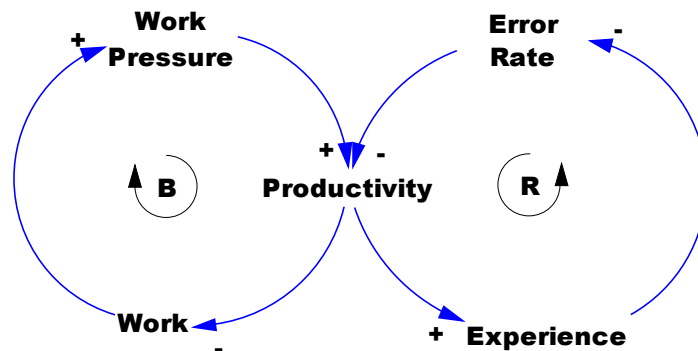
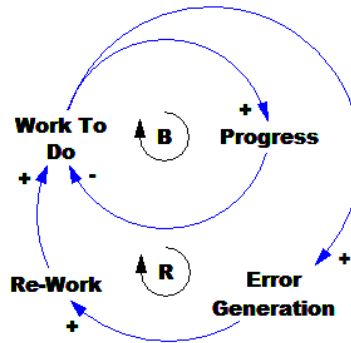


Figure 2.1: Macro Level Causal Loop Diagram

#### 2.3.1. Work/Rework Cycle

The Figure 2.2 presents the basic re-work loop that is common to the system dynamics literature which consists of both a reinforcing and balancing loop. This diagram presents a portion of the model that is repeated for both simple and complex work in the overall model. The *Work To Do* is completed through

*Progress*, which increases with more work, in turn decreasing the amount of *Work To Do*. However, a portion of the *Work To Do* results in *Error Generation*, represented in the model as a function of the experience of the team completing the work. This *Error Generation* increases the amount of *Re-Work* and the total amount of *Work To Do*.

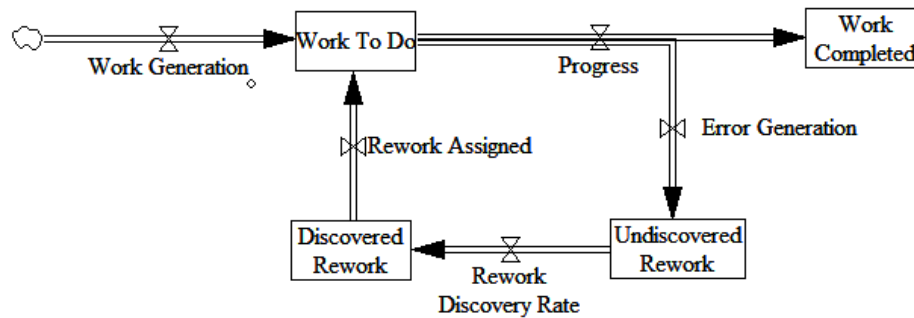


**Figure 2.2:** Re-Work Generation Causal Loop Diagram

### 2.3.2. Workforce Creation

The final causal loop diagram, *Figure 2.3*, presents a view of the system through the personnel lens that is repeated for several skill levels and combined into teams of workers. An important aspect of this model, it considers the limited pool of *Potential Skill* as the model accounts for the fact that only a small group satisfies the requirements for these types of jobs. The *Hiring Rate* decreases the pool of *Potential Skill*, outside the organization, while increasing the *Skill Onboarding* within the organization. As more people are onboarded, the *Onboarding Rate* increases which increases the *Skill New* variable which represents the number of new employees at a particular skill level. As the *Progress at Level* increases, employees move from *Skill New* to *Skill Experienced* through *Gaining Experience*. *Total Skills* includes all employees in the *Skill Onboarding*, *Skill New*, and *Skill Experienced* in this aging chain for each level of skill required by the organization. An increase in *Total Skills* decreases the *Skill Gap*, the difference between the *Total Skills* and the *Desired Skill Level*. As the *Skill Gap* increases it triggers the *Hiring Rate* to ensure the organization has sufficient people to perform the work required.





**Figure 2.4:** Rework Cycle

Team performance literature highlights many aspects that drive performance. With the guidance from the literature and the client, this model limited team performance to three factors: *Morale*, *Work Pressure*, and *Team Instability*. These stocks help modulate how work is accomplished through *Progress* and *Error Generation*. The first factor, *Morale*, shown in Figure 2.5, provides an excellent example as an increase in *Progress* reduces the *Rework Proportion*, which in turn increases *Morale*, raising *Productivity*, and creating a reinforcing loop (R - Happy People Work Hard). A coupled balancing loop exists as errors rise to the surface and the team realizes their actual pace (B - Finding Errors Slows Progress). The links between job performance and satisfaction have been attributed in some work as an expression of Maslow's hierarchy of needs, where workers that perform well exhibit elements of self-actualization and thus express high morale [17]. While literature remains divided on the directionality of the relationship between job performance and satisfaction [18], the feedback loop structure supports the bidirectional view. Similar research on person-role fit suggests that performance significantly impacts satisfaction [19].

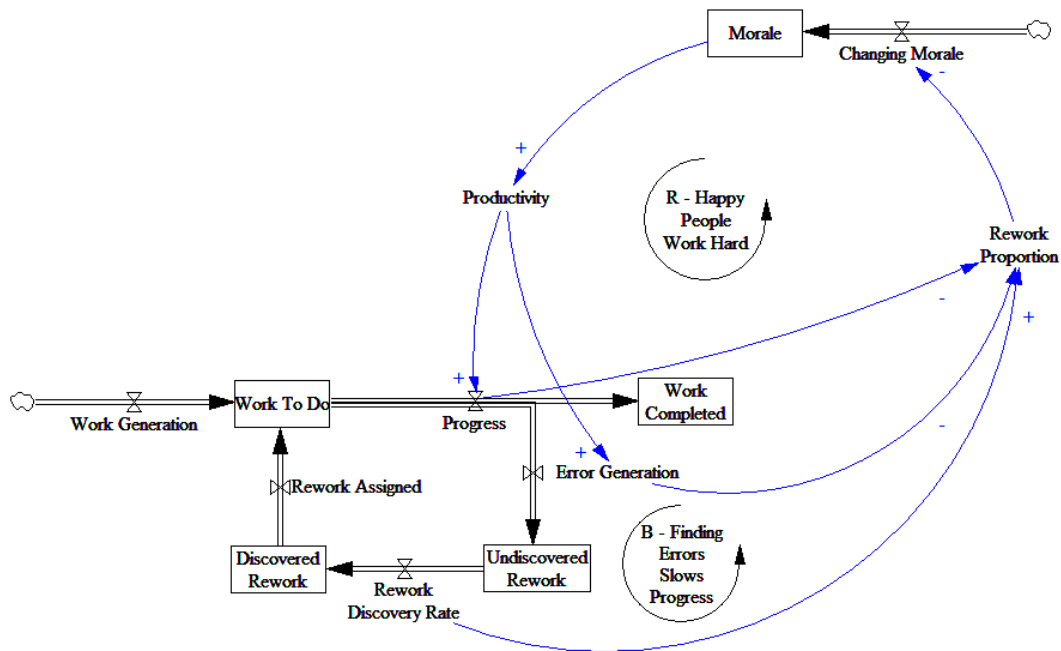


Figure 2.5: Morale

The second factor, *Work Pressure*, shown in Figure 2.6, models how a team responds when work starts to accumulate. *Work Pressure* builds as *Incoming Work* outpaces *Outgoing Work*, where the team's current pace will fail to meet continued demands. This growing pressure increases *Productivity* (an effect of increasing work spent working), raising the rate of *Progress* and *Error Generation*, which in turn increases the *Outgoing Work*, and reduces *Work Pressure* (B – Work Pressure Increases Work Rate). However, increasing *Work Pressure* also happens to reduce *Morale*, which leads to lower *Productivity*, reduced *Progress*, less *Outgoing Work*, and ultimately more *Work Pressure* (R – Work Pressure Lowers Morale). This setup holds similarities to Oliva and Sterman's workforce modeling [7, 8], but allows for pressure to accumulate and decrease at differing rates. One significant difference from their efforts is that workers do not cut corners to address their backlog. This represents an organizational artifact and may warrant adjustment depending on the application.

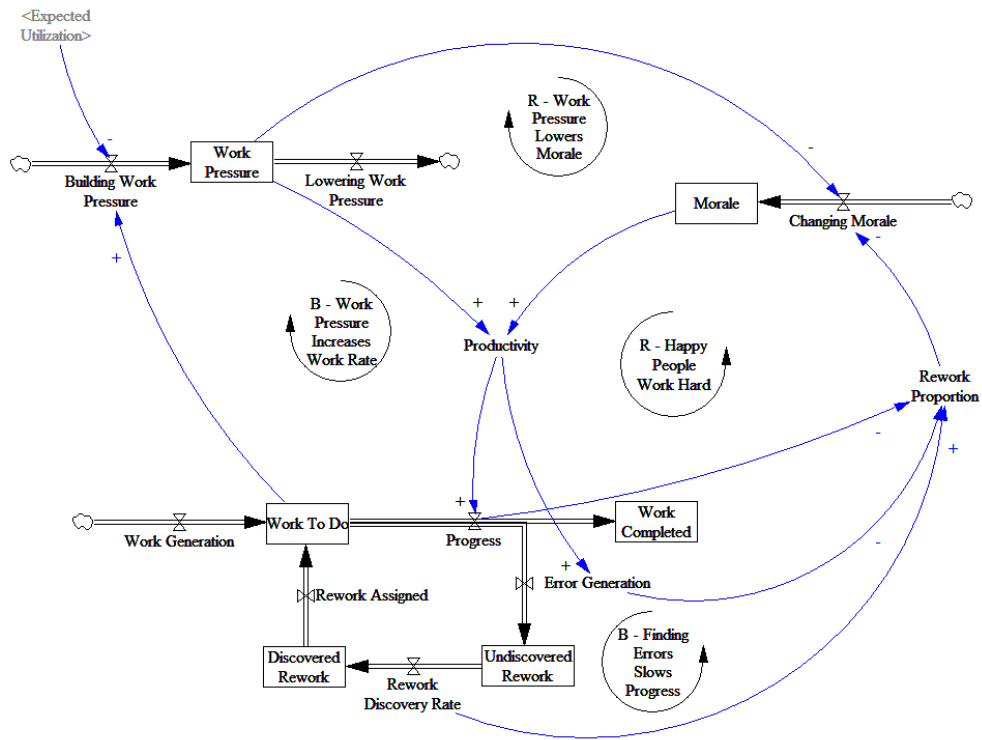


Figure 2.6: Work Pressure

The last factor, *Team Instability*, represents the effect of moving people into and out of these teams. When a team member joins or leaves, the team inevitably adjusts their workflow or reorganizes how they accomplish work. The idea that stable teams generally perform better proliferates much of the team performance literature [20] and often in sports settings [21, 22]. Previous research also suggests that stable teams demonstrate greater learning behaviors within the knowledge tasks [23]. To help capture the cost of organizational restructuring, the number of people joining and leaving the team is compared to the team size. This comparison allows for small changes in large organizations to have relatively low impact while small changes in small organizations should have a much larger effect. This dimensionless figure, *Team Instability*, shown in Figure 2.7, decays with a standard adjustment time and directly impacts the team's *Productivity* negatively.



### 2.4.2. Multi-Tiered Workforce

Within this organizational structure, skilled labor needed to be disaggregated due to their relative differences in hiring pools, an internal promotion bias, and the work that each worker could be assigned. To accomplish this, we first tackled a single-tier workforce creation module to represent the lowest skilled worker (*Skill 1*), see Figure 2.8. *Skill 1* workers can only be assessed into the system by enticing potential hires through a traditional hiring mechanism, B – Hire More People. To help represent the sometimes-sparse number of skilled candidates, potential hires grows and shrinks at some predetermined rates. The keeps the organization from being able to arbitrarily hire large quantities of workers on demand. The organization sets the total number of *Skill 1* workers desired (*Desired Skill 1*), compares that against the current number of *Skill 1* workers (*Total Skill 1*), and hires new workers to fill the gap with some *Hiring Delay*. Once hired, workers enter an inactive onboarding process before joining the active workforce as a new member. This new member is akin to Oliva and Sterman’s Rookie distinction [8]. New workers attrite out of the workforce or gain experience by completing tasks and becoming an experienced worker. Attrition of all workers expresses itself as a function of a standard quit fraction that is scaled by *Morale*. The higher the *Morale* the fewer workers leave.

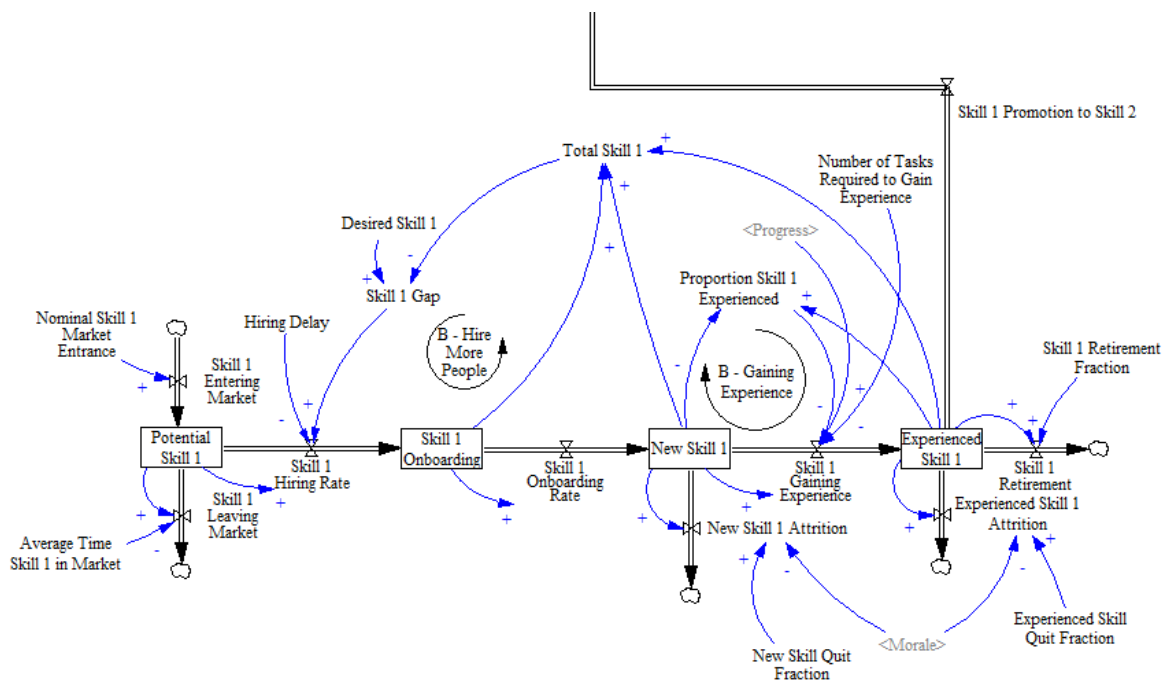


Figure 2.8: Single Tier Workforce Creation

The variable *Proportion Skill 1 Experienced* helps regulate the flow of new workers to experienced workers as both new and experienced workers complete tasks through the *Progress* variable. This convention ensures that only tasks performed by inexperienced workers (i.e. *New Skill 1*) helps contribute to their eventual transition to the experienced stock. Once a worker reaches the experienced distinction, they can attrite, retire, or promote out of the status. This basic structure helps build the basis for the multi-tiered workforce module of the model.



By adding cloned structures with slight modifications to facilitate internal hiring, a multi-tiered workforce emerges, as illustrated in Figure 2.9. To connect the entry tier with the next labor rung, the promotion of experienced lower skilled workers connects to the new higher skilled stock. The choice to insert these newly promoted workers into the new skill bucket reflects the assumption that this transition will still require these workers to gain new talents and knowledge before conversion into the experienced bucket. As currently structured, the organization maintains a heavy internal promotion bias over external hires. This behavior can be adjusted by creating a sensitivity to internal hires variable that adjusts the internal promotion rate to allow for external hires to fill gaps. Currently, internal employees fill all skill gaps with promotions when available.

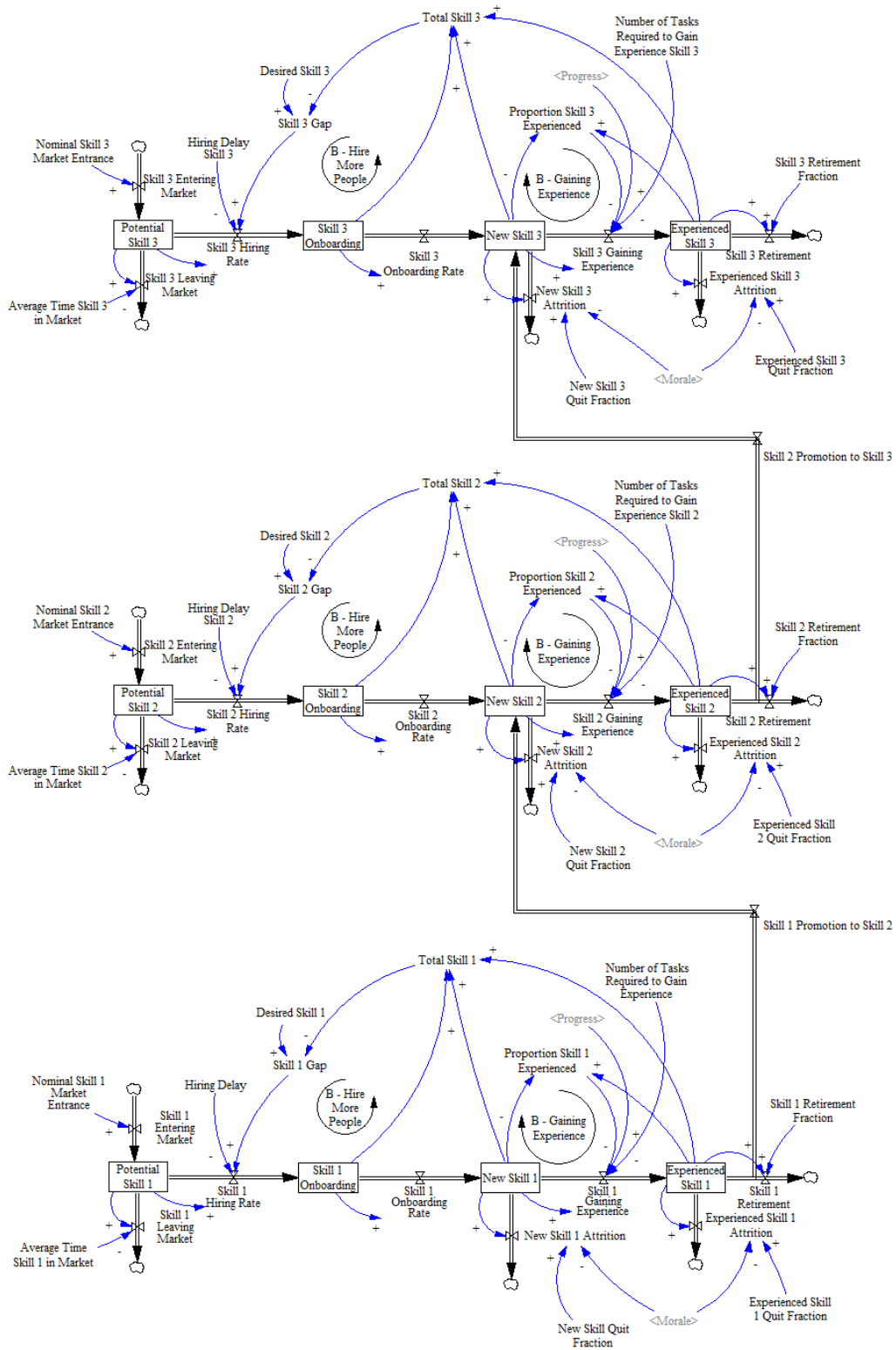


Figure 2.9: Three-Tiered Workforce

### 2.4.3. Dynamic Work Assignment

Because two separate work complexity structures exist, the model must have a mechanism to determine which workers conduct which work. We made several assumptions to help simplify this assignment. In the model, the lowest skilled workers (*Skill 1*) focus solely on the lowest complexity work while the highest skill workers (*Skill 3*) focus solely on the highest complexity work. This leaves the middle tier worker (*Skill 2*) to be divided between two types of work. *Work Pressure* helps us feed the assignment of the middle skill workforce. As *Work Pressure* increases the number of *Skill 2 on Team* increases, adding to the *Workforce*, increasing the rate of *Progress*, and ultimately reducing *Work Pressure* (B – More Pressure More Workers). Through this balancing loop, *Skill 2* workers join the lower complexity work structure with all remaining *Skill 2* works tackling the higher complexity work. The choice to force prioritization of *Skill 2* on higher complexity work reflects a client decision. Conceivably, this tier of worker could be split evenly or biased in the opposite direction with only slight modifications to the allocation formulation.

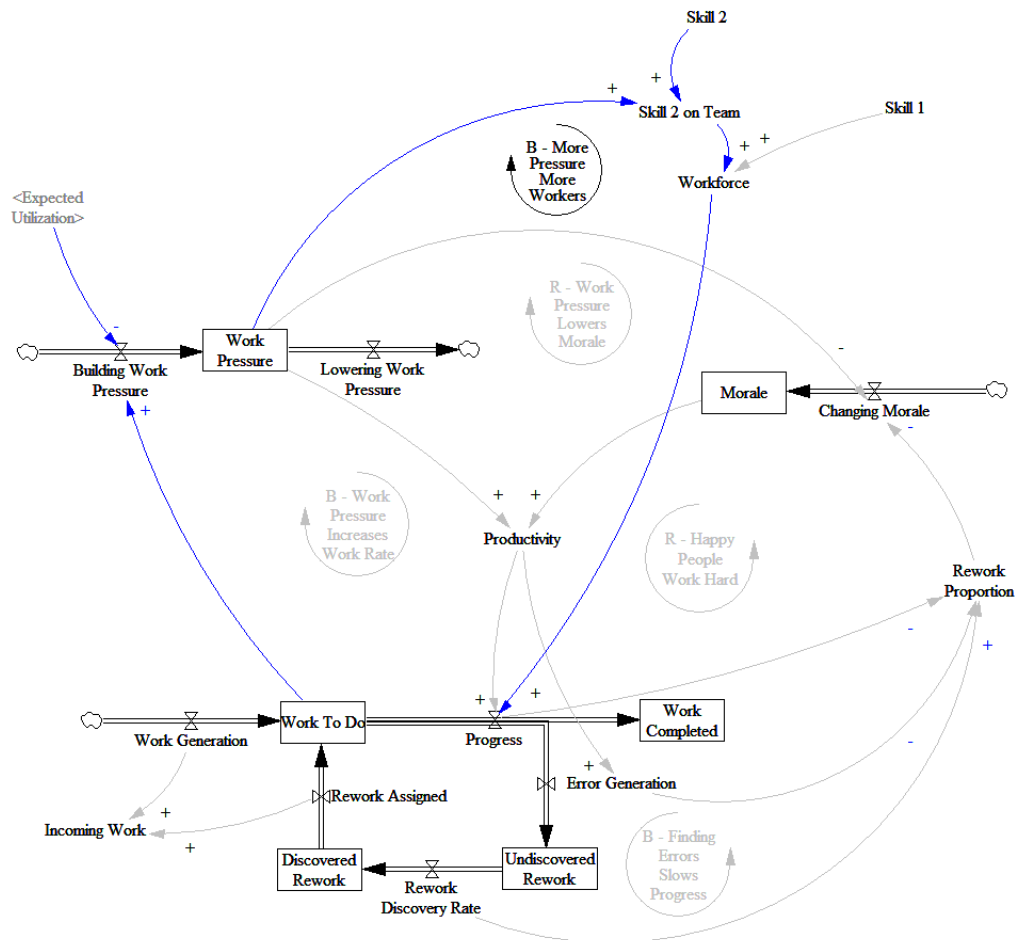


Figure 2.10: Dynamic Work Assignment

Because not all teams contain the same amount of experience, this must be represented in the model where higher skilled teams create less errors. In Figure 2.11, *Team Experience* derives from the proportion of *Skill 1* and *Skill 2* workers, with higher experience reducing the *Error Fraction*, which in turn increases *Progress* and reduces *Error Generation*. In actuality, the final model accounts for the difference between new and experienced staff at each level (i.e. *New Skill 1* and *Experienced Skill 1*) with calibrated error sensitivities for each pairing. This allows the client to scale a nominal error fraction by pairing and use that to calculate the final team *Error Fraction*.

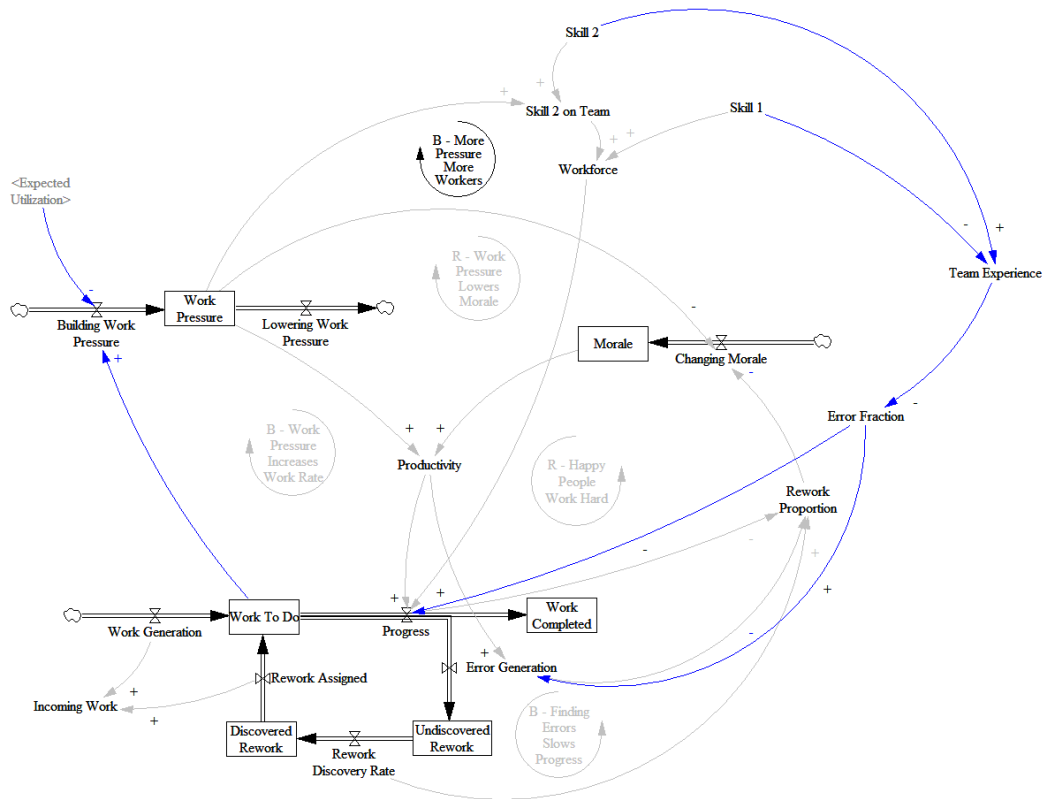


Figure 2.11: Team Experience Level

#### 2.4.4. Model Adjustment to Fit Differing Environments

By decoupling work complexities and skilled workers into tiers, this modeling structure can simulate many differing environments. Changing the error and allocation sensitivities, work arrival rates, and proportion of work complexities allows an organization to simulate several different conditions. Because the structure remains the same, the client does not need additional time to learn about a different model but can simply redefine the work condition they expect to experience. For example, imagine an engineering firm that has both a line and research division. The line division conducts rather simple work (low complexity) at high consistent workloads (steady linear work). The research division designs many of the processes the firm uses and usually works on highly complex work that the leadership dreams up at the quarterly review (pulse input). The model structure can handle each of these situations by simply adjusting

a few exogenous inputs. The importance of this cannot be understated from a consulting perspective. Once the client helps develop, calibrate, and understand the simple structure, it can be leveraged in several use cases within the same organization.

## 2.5. Model Analysis

Elaborating off the example described above, we examine this environment under two simple conditions. For simplicity, all variables remain the constant (i.e. the number of desired workers, amount of work generated, quit fractions...), but the sensitivity to which we assign Skill 2 works varies to explore response. This sensitivity interacts with *Work Pressure* to split Skill 2 works among the two tasks. The equations that dictates the number of *Skill 2 Experienced* workers is outlined below:

$$Skill2ExperiencedOnWork\ 1 = Work1Sensitivity * WorkPressureWork1 * Skill2Experienced \quad (1)$$

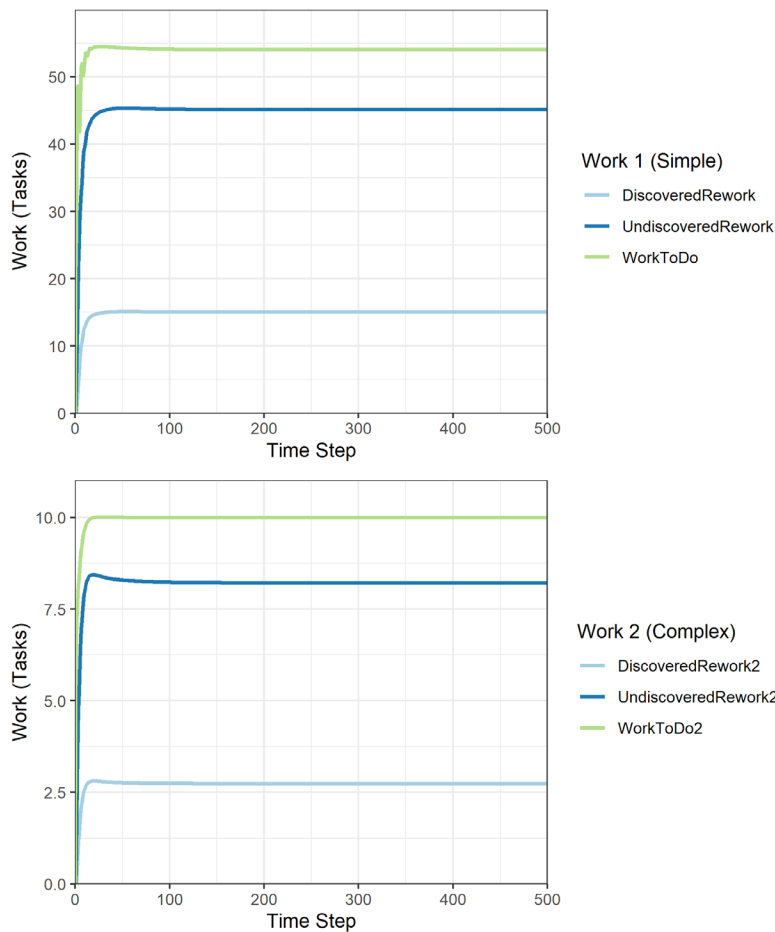
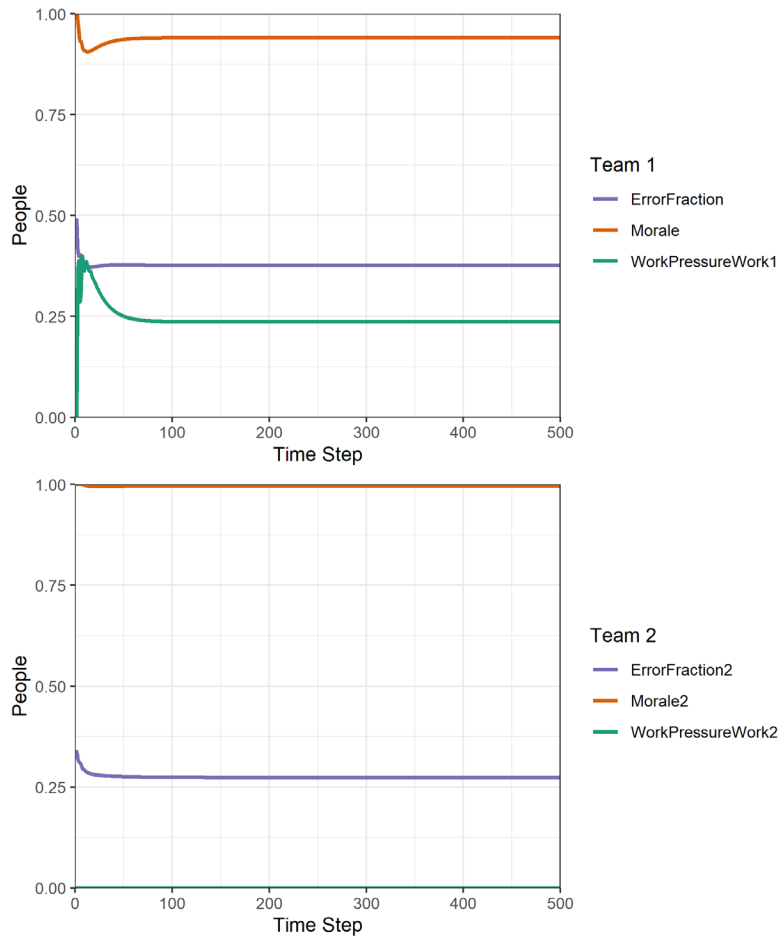


Figure 2.12: All Work Clears and Reach Steady State in Scenario 1

By default, all Skill 2 workers not on Work 1 revert to Work 2 (more complex work). In the first scenario, the client decides to set the sensitivity to 1, suggesting that if *Work Pressure* for the simple tasks reaches its maximum (1), they will allocate all of the Skill 2 workers to those tasks. In this scenario, all teams manage to clear their backlog and reach a steady state between work arrival and work completion, see Figure 2.12. However, the conditions under which these teams operate remain concerning. Because teams can balance their loads thru changes in work pressure, decision makers must explore the conditions of their teams as they operate. Figure 2.13 shows three artifacts of team conditions that help identify these states.



**Figure 2.13:** Team 1 operates under increased pressure when compared to Team 2 in Scenario 1

With this graph, the decision maker can quickly identify that Team 1 operates under increased *Work Pressure* to accomplish their workload, even at steady state. Team 2 remains in excellent condition with no perceived increase in *Work Pressure* during the simulation. This artifact suggests that Team 2 may hold additional capacity that could be leveraged.

To test this assumption, we adjust the work assignment sensitivity from 1 to 2. This action does reduce the *Work Pressure* on Team 1 (from steady state values of ~.24 in Scenario 1 to .14 in Scenario 2) with no negative effects on Team 2. Decision Makers may benefit from seeing the breakdown of workers between the teams as shown in Figure 2.14. This graphic exposes the driving mechanism of increased *Work Pressure* on Team 1. As the organization promotes Skill 1 workers to Skill 2 (to fill a previously identified gap), a large portion of them are shifted to Team 2. The delay involved in recruiting and onboarding new Skill 1 workers increases the pressure on Team 1 until the organization reaches their desired state. Once at the desired state, Team 1 has the capacity (with additional *Work Pressure*) to clear the work backlog but must maintain some residual *Work Pressure* to maintain equilibrium. Possible solutions to this dynamic could include seeding the teams with additional workers up front or utilizing temporary contractors that could fill the worker gap until the organization can hire the appropriate amount of Skill 1 workers.

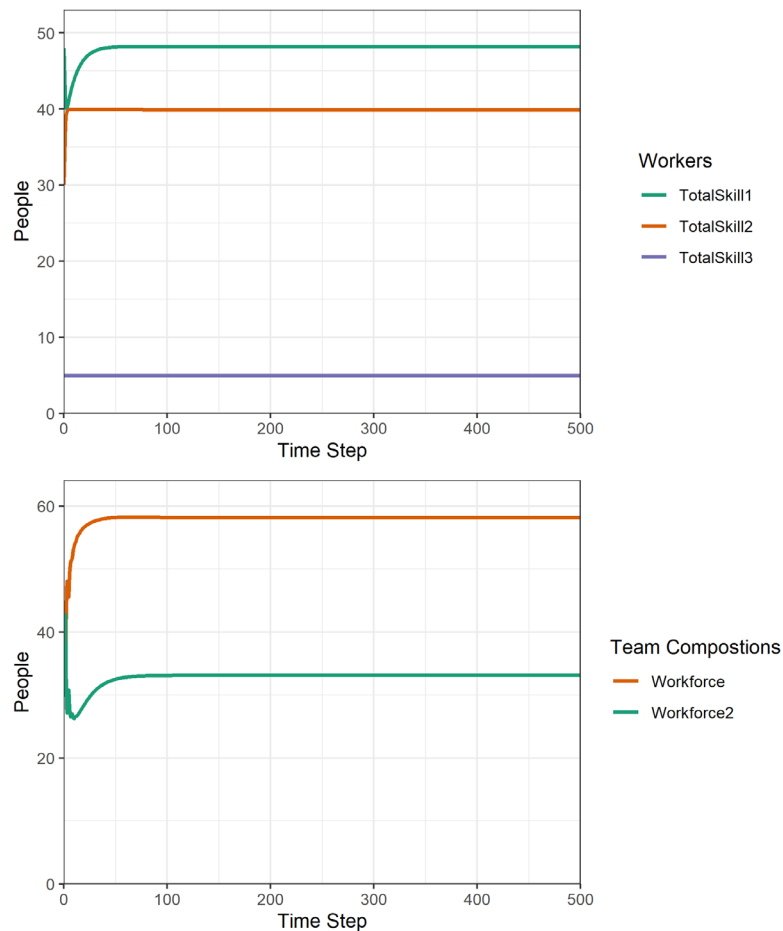


Figure 2.14: Workforce Break down for Scenario 2

## 2.6. Limitations of Model

Once compiled, the multi-tiered workforce and work complexity model remains complicated. It has many moving pieces and will be difficult to quickly explain. This is not unlike many other System Dynamics



models but warrants some consideration. If the client cannot afford the time to work through this level of detail, or the problem does not warrant disaggregation, it may be overly confusing and cumbersome.

Even if the problem and client warrant the disaggregation of work and people, tuning such a model may be extremely difficult or impossible depending on the situation. For example, how much error does a *Skill 1* worker generate? While the behavior and feedback loops of the model remain defensible, this is not a situation that can be optimized. This model still lacks the ability to tell an organization the precise split in workers required. If tuned appropriately it can provide an appropriate estimate and help the organization better think through the tradeoffs, expected behaviors, and anticipated rates of work.

The desire to capture everything, “model reality,” remains high. Once the client starts to explore behavioral and physiological factors, it may be difficult to draw a boundary, further complicating the ability to tune the model. When adapting this model to a particular situation, it may be helpful to adjust and limit these types of mechanisms that affect *Morale* and *Work Pressure* more specifically. Triaging this situation will limit the complexity of formulation and aid in interpretability.

## 2.7. Conclusion and Future Work for Model

This work demonstrates a method to disaggregate previous System Dynamics work force models to enable the design and analysis of more complex work environments. By adding these structures, companies can explore changes to their organizational structure, anticipate the delays and challenges to staff the structure, and predict how the different structures will perform. This provides a more tangible tool as organizations think about transformations. Future work should explore additional changes to the workplace. For example, how would an organization expect performance to change when changing from waterfall to agile methodologies? These types of changes require new roles, staff, and possibly a new work/rework structure. This type of modeling however would be substantially appreciated as more large organizations explore this restructure.



---

## Chapter 3

# Building Interactive System Dynamics Simulations in R<sup>2</sup>

### 3.1. Introduction

System Dynamics provides a mechanism to explore a wide range of phenomenon and contains several features that have made it an endeared modeling methodology. For example, the ability to build interactive models, often called management flight simulators, gives modelers and stakeholders the ability to interactive with and test policy alternatives quickly through interrogations of the model. Several companies and software packages offer tools to create these types of interfaces. This work builds off the recent exploration of system dynamics and R, to elaborate a modeling workflow the creates an interactive system dynamics model with R. The paper starts with a brief literature review, explains the method, discusses the results and limitations, and concludes with future work.

### 3.2. Literature Review

This literature review briefly reviews System Dynamics as a mathematical modeling technique, describes the continued evolution of R, and then outlines recent explorations of system dynamics into the R ecosystem.

#### 3.2.1. System Dynamics

System dynamics enables the understanding of the dynamic behavior of systems using accumulations, flows, and causal relationships within a system. It exposes the underlying structure of a system and models the behavior over time to adjust individuals' mental models of the system and test potential policy alternatives. Forrester's [1] work describes a system as "a grouping of parts that operate together for a common purpose". He further classifies two types of systems: open systems, in which external variables affect the system, or closed systems where all variables are internal to the system [2]. In his book, *World Dynamics*, he presents an example system model of the world that described the interrelations between population, capital investment, geographical space, natural resources, pollution, and food production [3]. The behavior of a system over time is the dynamics of a system, which are often complex and non-linear in nature because of the system's underlying structure [2]. This complexity stems from feedback within the system, time delays between decisions and effects, and the learning process of the system [4]. These attributes of systems make them difficult to understand and identify the cause and effect relationships without an effective model of the system.

Within the management domain, Forrester [2] described the potential for system dynamics to assist decision makers understand the implications of their policies and potentially identify and mitigate unintended consequences of their decisions. Applications of System Dynamics have provided insights across several domains; including corporate policy, infectious disease, commodity markets, and drug addiction, and commodity markets [3]. Companies and consultants have extensively used System Dynamics for managing large, complex projects with a great deal of success. One area where businesses

---

<sup>2</sup> Parts of this chapter were submitted and accepted into the 38<sup>th</sup> International Conference of the System Dynamics Society (Caddell, J., Connell, T., Enos, J., & Driscoll, P. (2020). *Building Interactive System Dynamics Simulations in R*. The 38th International Conference of the System Dynamics Society.).

utilize System Dynamics is in the development of corporate strategy and analysis of business decisions after a crisis or complex problem triggers these shifts in business strategy. System Dynamics assists in determining the root cause of the crisis and identifying potential consequences of alternative courses of action [4]. One of the main benefits of the system dynamics methodology is that it makes explicit causal relationships between variables to understand how the underlying structure and accumulations affect the behavior of the system over time.

Communication often drives whether a modeling effort succeeds or fails. Sometimes this is communicating to other modelers, but often this means explaining a model and results to a client or stakeholder. While several methods help with this task, interactive modeling remains the most appealing manner to engage someone with a model. Interactive modeling builds a more nuanced understanding of the model and helps develop policy buy-in through exploration [24]. There are several tools that help build this experience. Vensim contains the SyntheSim feature where a modeler can build a dashboard within Vensim for the client to investigate the running model. Forio offers another mechanism to build high quality interactive simulations with the ability to use Vensim and other system dynamics software as model baselines. The project SDEverywhere (<https://sdeverywhere.org/>) translates Vensim models into C code and enables web and desktop applications. Each of these mentioned alternatives varies in terms of cost and technical knowledge requirements but offers a viable method to deliver interactive system dynamic models.

### 3.2.2. R Background

R is a computer language that focuses on data manipulation, statistical analysis, and graphical display. Ross Ihaka and Robert Gentleman developed it in 1991 and documented its development in their 1996 article R: A Language for Data Analysis and Graphics [25]. It is free and boasts a strong open source community that prides itself of inclusiveness and a desire to develop new users. The growth of R can be directly linked to the growth in the interest in Data Science and this furious focus on open source development. The basic R language has many appealing features, but data analysts appreciate the language's abundance of packages. An R package is a set of prewritten functions with a set purpose. The R community has amassed a large range of packages that enable a fluent R user to do anything from basic statistics, produce high-fidelity graphs, establish and use databases, build machine learning models, and deploy websites. The Comprehensive R Archive Network (CRAN) hosts and curates most of the stable R packages which enables a single line of code to download a desired package. GitHub acts as the primary host for development versions of R packages. Growth in package developing has continued to accelerate, with over 12,000 packages released in 2019 (Figure 3.15).

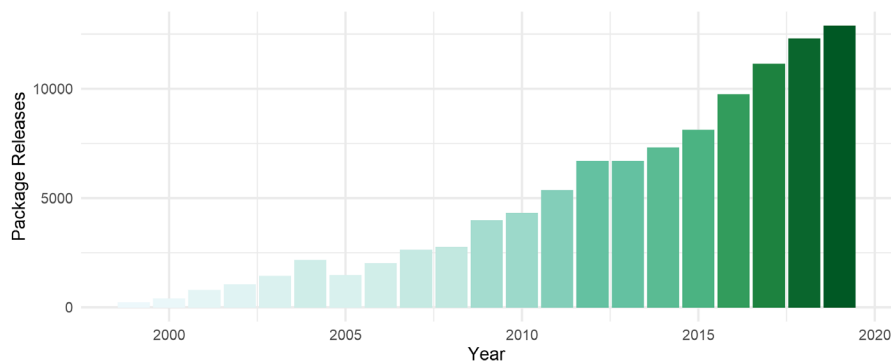


Figure 3.15: CRAN package releases (aggregated by year)

### 3.2.3. System Dynamics in R

Dr. Jim Duggan, a senior lecturer at the National University of Ireland Galway, has made significant contributions to the cross pollination of System Dynamics and R. Dr. Duggan’s 2016 book, *System Dynamics Modeling with R*, outlines the case for R as a framework to conduct system dynamics modeling and provides a straightforward introduction into R’s utilization for such modeling [26]. He expanded on this work to focus on a growing subculture within the R community, the *tidyverse*<sup>3</sup> [27], elaborating on methods to conduct data analysis [28]. His most recent work in this area explored the use of R to conduct sensitivity analysis and calibration on dynamic models [29]. Conducting system dynamic modeling in R, adds the additional benefit of enabling more easily reproducible research, an increasing desire from the system dynamics community [30], as all models and graphics can be executed in a single script. Recently, Sinead Morris published a package (shinySIR) which builds a set of common susceptible infectious recovery (SIR) models in the Shiny framework, providing a readymade interactive system dynamics model [31]. System Dynamics work in R remains a rather new but growing movement with many possibilities.

### 3.3. Methodology

In this article, we will outline the development of an interactive system dynamic model in R, Figure 3.16. We describe how we develop a model in the Vensim modeling software, extracted the output, transformed it into computable R code, and developed it into a Shiny Application that can be hosted online as a system dynamics simulator with customized output.

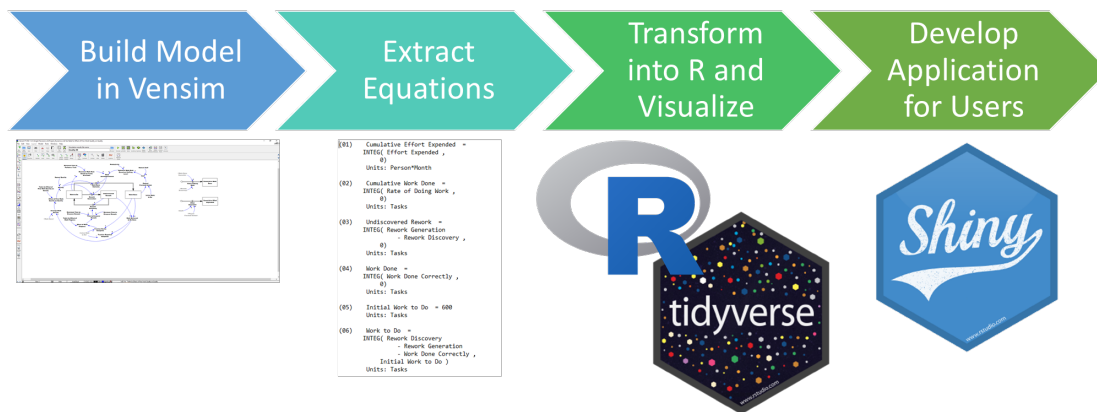


Figure 3.16: Proposed Interactive System Dynamics Modeling Process in R

#### 3.3.1. Modeling in Vensim

Many System Dynamic’s modelers use the software package Vensim to conduct their modeling. Vensim provides many refined functions and a sufficient user interface to quickly establish models in a graphical manner. Vensim also provides a free version of their software for academic and personal use. For these reasons, we feel that leveraging the graphical user interface (GUI) and skills already learned by

<sup>3</sup> The tidyverse is a collection of R packages that are designed to work together to perform Data Science. All packages share an underlying design philosophy and grammar to accelerate learning and increase interoperability ([www.tidyverse.org](http://www.tidyverse.org)).

modelers remains an important aspect of any workflow. For this work, we built a representation of the classic project dynamics models [5, 32] in Vensim as we would expect several modelers to do.<sup>4</sup>

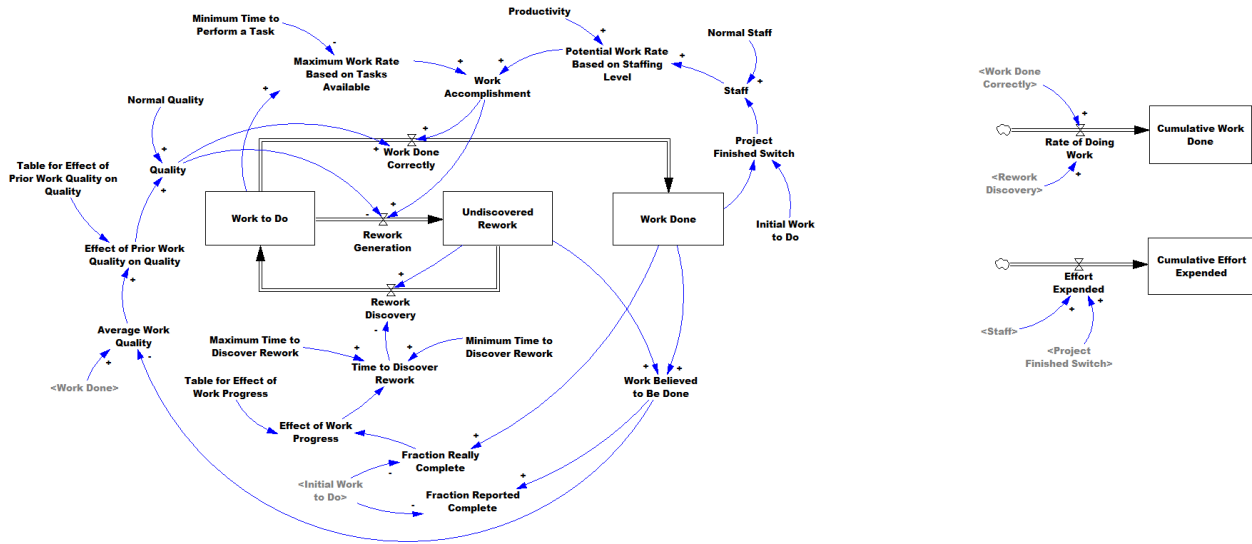


Figure 3.17: Project dynamics model adapted from previous work [5, 32]

The modeling community has grown accustomed to several common features within Vensim, and this model has several of the most popular. These included the functions MAX, MIN, IF THEN ELSE, and the ever-flexible Look Up Table.

### 3.3.2. Conversion from Vensim to R

Once the modeler feels complete with the Vensim model, we can extract the equations from Vensim for use in R. Vensim allows you to save the equations as a text file through the Documentation function on the left side of the tools bar. Dr. Duggan demonstrates this on his GitHub and provides some helper functions for users with the DSS version of Vensim.<sup>5</sup> Users with the free (PLE) version of Vensim will be unable to execute this step without some additional cleaning of text. These functions remove unnecessary characters and annotations from the Vensim documentation and establish an ordinary differential equation based off the Vensim model. Some specialized functions within Vensim will not translate directly into R and will either need manual adjustment or an additional customized regular expression. For example, the lookup tables from the model in Figure 3.17, do not translate directly. We used the *approxfun* function in R to

<sup>4</sup> Other Software might be able to leverage pieces of this workflow but are not addressed in this work for brevity. Some paid versions of system dynamics software may also alleviate several of the gaps we are attempting to address. However, due to their cost, we have found many modelers unaware of these options, unable to purchase them, and/or more comfortable with Vensim.

<sup>5</sup> Code can be found here: <https://github.com/JimDuggan/SDMR/tree/master/reader>



develop this feature within the R version of the model (see Figure 3.18). This function creates a method to conduct linear or piece-wise interpolation akin to Vensim's lookup.

```
#table lookup  
  
Table_for_effect_of_prior_work_quality_on_quality <- approxfun(  
  x = c(0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1),  
  y = c(0.05, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1))
```

**Figure 3.18:** Vensim Table Lookup Function in R

Modelers should always consider the computational bounds of their method. Vensim provides a highly optimized environment that can quickly execute models. When converting to R, we do lose some of this speed. While several methods were attempted, leveraging the deSolve package [33] as demonstrated by Dr. Duggan's work proved the quickest. As the size of models grow, the computation time grows as well. Figure 3.19 examines the changing computation times of two different models as the number of time steps increases. The small model, depicted in Figure 3.17, contains 7 auxiliary variables, 5 stocks, and 23 equations while the large model includes 73 auxiliary variables, 30 stocks, and 114 equations. The models remained unchanged during each simulation, but the number of timesteps varied to show the resulting computational demand from this variance. The relationship depicted in Figure 3.19 demonstrates the expected linear growth behavior. As time steps increase, the time to run the simulation increases. Interestingly, the rate of growth for computation time depends on the model complexity/size. The larger model diverges from the smaller model, increasing in computation time at a faster rate as timesteps increase. Modelers must understand this relationship and make calculated parameter choices for their models to ensure they meet their desired performance. An interactive simulation with 4 second lags between simulation runs should be treated differently than a model that executes in 10th of a second.

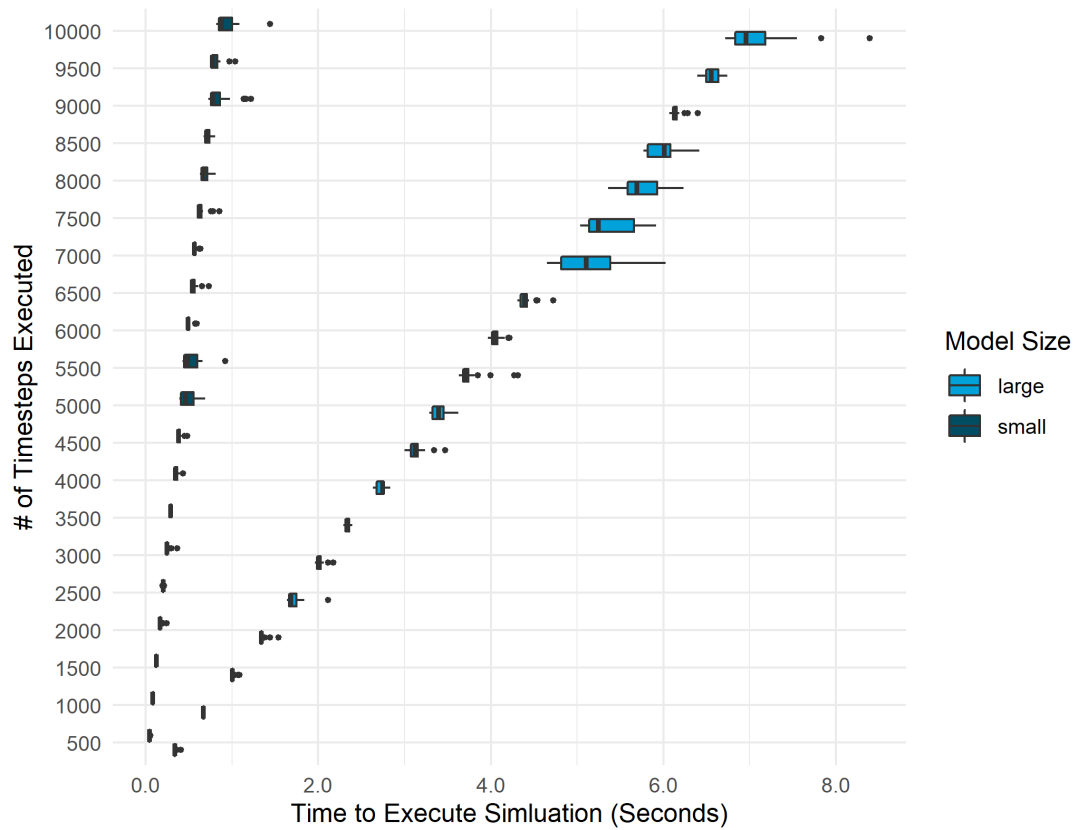


Figure 3.19: R System Dynamics Model Benchmarking

### 3.3.3. Visualizing Model in R

Once the ordinary differential equations are written in R, we can execute the model and begin collecting data. Combining the data generation and visualization portions of the modeling effort greatly improves workflow and enables more nuanced modeling. The graphics from Vensim leave much to desire, but R shines in this area. A modeler can now run their model, testing parameters and policies, exploring interactions, and visualizing them in journal quality graphs in the same environment. Having the ability to model output with high quality graphics and quickly iterate enables modelers to conduct more nuanced investigations. Additionally, the downstream work required due to adjustments of the model or graphics reduces as scripts now produce all data and output. No more running the model in Vensim, extracting the data, and then reworking the graphs in excel. Figure 3.20 shows example output of the Project Dynamics model developed in Figure 3.17. This process also provides the ability to easily share reproducible research, as a single script file could contain all model formulations and output graphics.

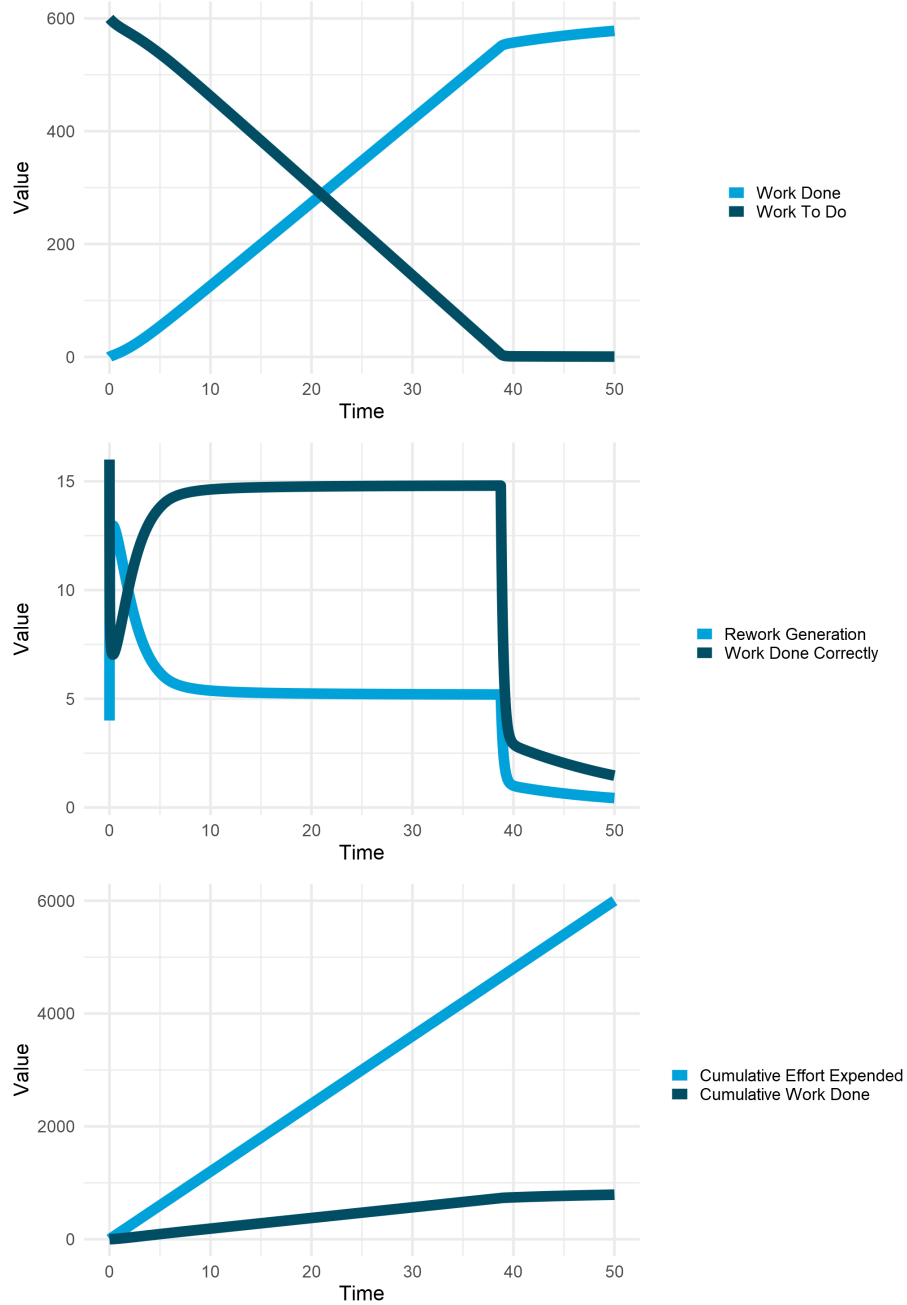


Figure 3.20: Example Project Dynamics Output in R

### 3.3.4. Conversion from R to Shiny Application

Communication and client engagement remain central to the power of System Dynamics. The ability to allow a client to engage with a model enables new insights and builds confidence in the work. Functions

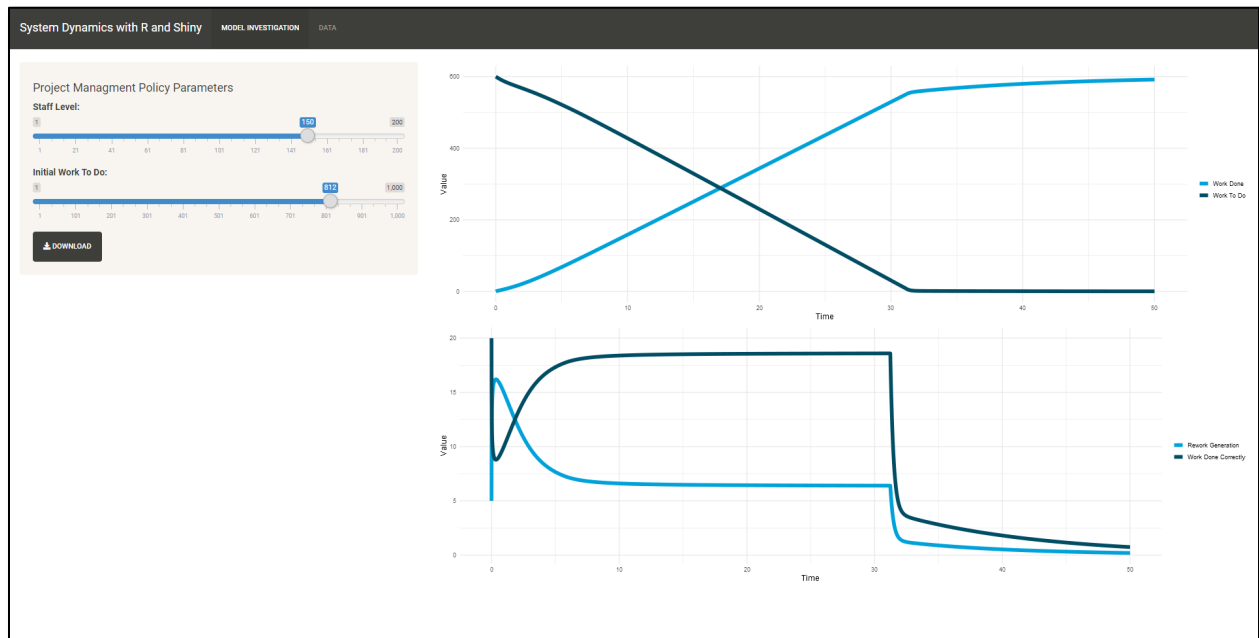
similar to Vensim SyntheSim have provided this capability, allowing users to change input and watch the model output change appropriately. We do not lose this functionality in R. In fact, for certain environments we gain some functionality or access. Shiny applications boast the same high-quality graphics as R, provide flexible display and input formats, and can easily be deployed as websites. The beauty of R rests in its easy software architecture as these Shiny applications can use the same code we previously developed to run a system dynamics model and graphics. The previous code requires little modification before occupying the server side of the Shiny app.

### 3.4. Results

This section describes the resulting Shiny application, features, and limitations.

#### 3.4.1. Model Manipulation and Data Extraction

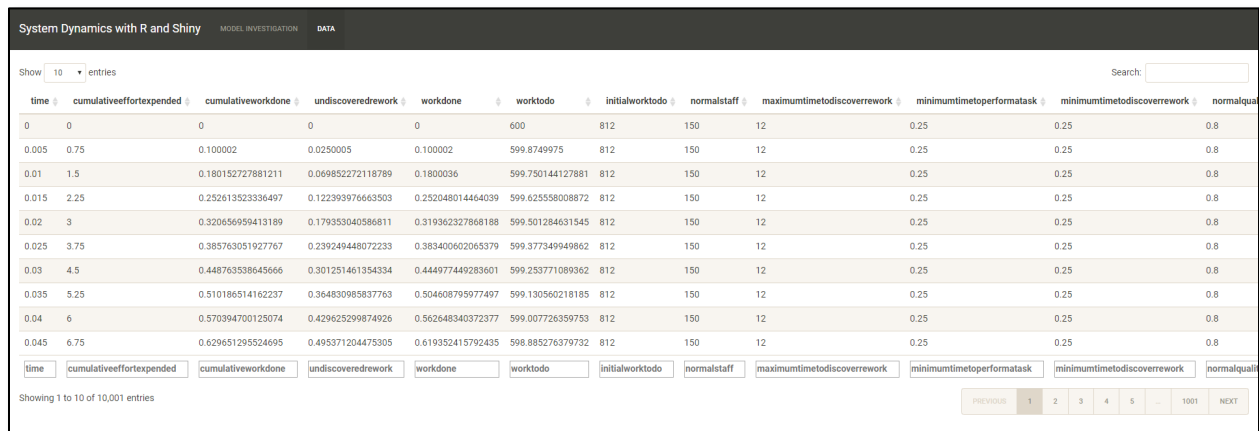
Shiny applications often develop similar feels, a result of good uncomplicated design and built in features. Our application leverages many of these features to make model manipulation intuitive for users. When the application opens, it defaults to the “Model Investigation” panel of the simple two-panel design (Figure 3.21). On this panel, users have slider adjustments for selected exogenous variables on the left and pre-defined variable graphics on the right. The application defaults to set numbers for these variables on instantiation, but adjustment by the user automatically prompts the server to rerun the model and generate graphics based of the new values.



**Figure 3.21:** Shiny Application – Model Investigation Panel

This sort of model manipulation enables quick and intuitive investigations of the model along with high quality graphics to better understand model results. We've also added a "Download" button so that the user could download the data from any simulation run for future reference. If desired, the application could have more variables for the user to adjust, different graphics, baseline policy levels, or even multiple models/output. Shiny provides an impressively flexible format to design user interfaces.

In the second panel, "Data", we display the resulting data from the simulation run (Figure 3.22). Shiny provides some standard filtering and search functionality in its Data Table view as well as pagination. This makes the data more transparent and allows users to explore the model in ways possibly unintended by the developer.



time	cumulativeeffortexpended	cumulativeworkdone	undiscoveredrework	workdone	worktodo	initialworktodo	normalstaff	maximumtimetodiscoverrework	minimumtimetoperformtask	minimumtimetodiscoverrework	normalquality
0	0	0	0	0	600	812	150	12	0.25	0.25	0.8
0.005	0.75	0.100002	0.0250005	0.100002	599.8749975	812	150	12	0.25	0.25	0.8
0.01	1.5	0.180152727881211	0.069952272118789	0.1800036	599.750144127881	812	150	12	0.25	0.25	0.8
0.015	2.25	0.252613523336497	0.122393976663503	0.252048014464039	599.62555808872	812	150	12	0.25	0.25	0.8
0.02	3	0.320656959413189	0.179353040586811	0.319362327868188	599.501284651545	812	150	12	0.25	0.25	0.8
0.025	3.75	0.385763051927767	0.239249448072233	0.383400602065379	599.377349949862	812	150	12	0.25	0.25	0.8
0.03	4.5	0.448763538645666	0.301251461354334	0.444977449283601	599.253771089362	812	150	12	0.25	0.25	0.8
0.035	5.25	0.510186514162237	0.364830985837763	0.504608795977497	599.130560218185	812	150	12	0.25	0.25	0.8
0.04	6	0.570394700125074	0.429625299874926	0.562648340372377	599.007726359753	812	150	12	0.25	0.25	0.8
0.045	6.75	0.629651295524695	0.495371204475305	0.619352415792435	598.885276379732	812	150	12	0.25	0.25	0.8

Figure 3.22: Shiny Application - Data Panel

### 3.4.2. Current Limitations with Method

While simple, this workflow provides some challenges and limitations. First, we recognize that converting System Dynamic models into R requires a certain amount of knowledge about the R language and coding environment. While Dr. Duggan's efforts outline an approach, it does require more expertise than has previously been published in the system dynamics community. Most notably the established regex translators of Vensim to R do not handle custom Vensim functions automatically. This functionality, and many more, can be included but must be defined by the user. Currently, it takes the author about an hour to transition from completed Vensim model to a simple complete Shiny application as demonstrated. Additionally, the hosting of shiny applications does require another set of unique skills. RStudio provides a service that allows a user to host a shiny application with a click of a button but launching at scale may be cost prohibitive.<sup>6</sup> Pricing ranges based on required active application hours between \$9-\$299 a month. Cheaper methods exist (AWS, Azure, etc..) but require advanced skills.

<sup>6</sup> RStudio provides easy Shiny application hosting through <https://www.shinyapps.io/>



### 3.5. Conclusion and Future Work for Method

Building interactive system dynamic models delivers a product that engages stakeholders and clients in significant ways. While there are many alternatives, R and Shiny provide a straightforward, quick, relatively cheap, and high-quality alternative for system dynamic modelers. Additionally, this method provides modelers the opportunity to launch these applications as a website, often circumventing many technical hurdles stakeholders may encounter. Future work may seek to automate many of the common tasks in the process and provide additional functionality for modelers to lower the required technical expertise to execute the workflow. It would also be helpful to have a portfolio of examples that users could reference and replicate for learning.

## Chapter 4

# How to Leverage the Workforce Application

### 4.1. Introduction

This chapter outlines how to leverage the application develop in support of this effort. It details application features as well as provides recommendations for benchmarking and model interpretation.

### 4.2. Application Overview

The application<sup>7</sup> was developed to help the DIA consider different workforce structures and how they might be able to expand or grow analytic capability. The DIA operations under significant hiring constraints and conducts multiple types of work with varying complexity. This application attempts to help answer two primary questions:

- 1) How much and what level of capability do we need to meet anticipated demands?
- 2) How does team performance change as we adapt?

The base model was developed in Vensim (see Chapter 2) and translated into R (see Chapter 3). Several adaptations were made to enable interactivity and generate the completed Shiny application. The following sections outline the application and its features.

#### 4.2.1. Application Navigation

The application consists of three separate pages (Model Investigation, Model Assumptions, and Data). Users switch between the pages by utilizing the navigation toolbar at the top of each page (see Figure 4.23).



*Figure 4.23: Application Navigation Bar*

#### 4.2.2. Model Investigation

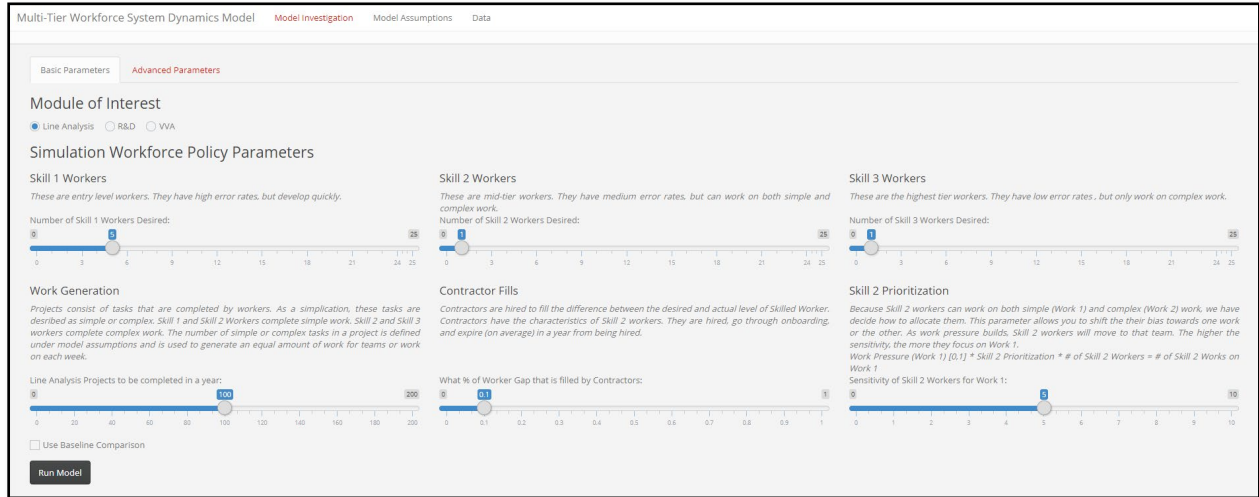
The Model Investigation page is the default and appears when the application is launched. This page consists of two sections: Parameters and Visualization.

##### 4.2.2.1 Parameters

To enable for quick adjustment and exploration by stakeholders, several parameters were exposed with widgets. The parameters were split between basic and advanced to reduce the upfront complexity. Most stakeholders will only utilize the basic parameter set, but some analysts may be interested in exploring some of the advanced parameters. Descriptions and explanations for each variable are provided to help users understand what the parameters represent in the model.

<sup>7</sup> The application is currently hosted at: [https://caddell.shinyapps.io/multitier\\_workforce/](https://caddell.shinyapps.io/multitier_workforce/)

Under the basic parameters tab, user can select the module of interest, set the level of desired workers (at each skill level), change the number of projects or efforts to be attempted in a year, adjust the percentage of gap filled by contractors, and alter the bias assigned to Skill 2 workers for simple work. Each of these parameters is adjusted with either a radio button or slider widget (see Figure 4.24).



**Figure 4.24: Application Basic Parameters**

Because the model and visualizations take a couple seconds to execute and display, the visualizations remain static until the user clicks the “Run Model” button. This prevents any errant executions of the model while attempting to adjust the parameter settings and reduces lag.

Above the “Run Model” button is a check box for Baseline Comparisons. Checking this box reveals a hidden portion of the tab (see Figure 4.25) that enables a user to compare model runs against one another. The difference between the runs can quickly be changed and visualized.

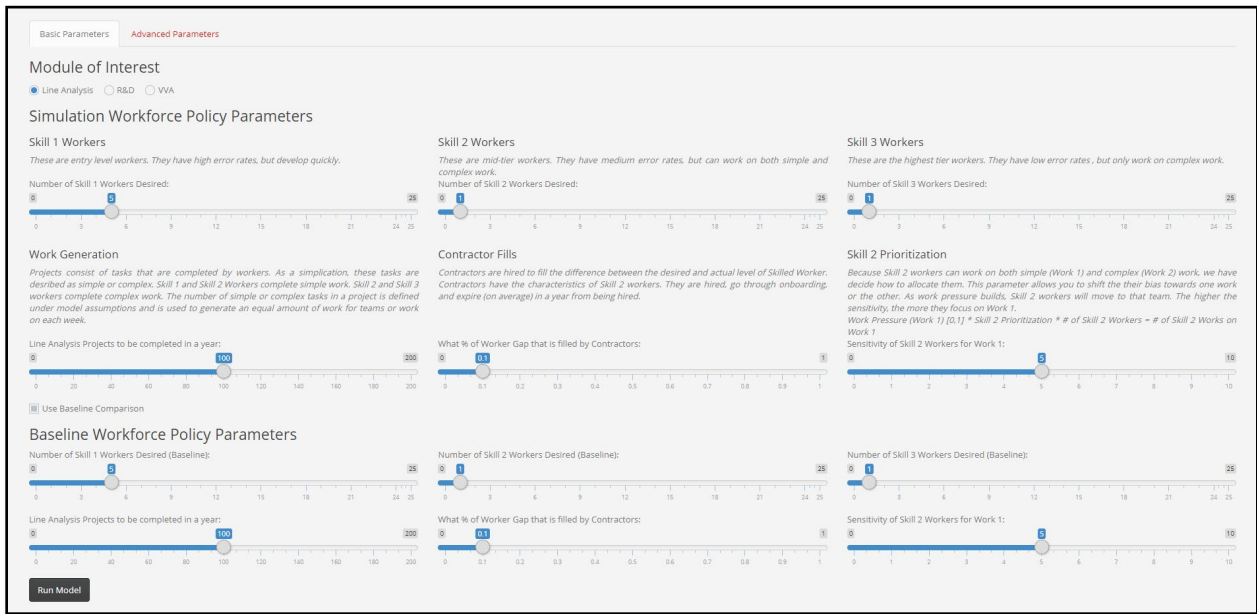


Figure 4.25: Application Baseline Comparison

The advanced parameters tab contains an external hiring switch, sensitivities from Rework and Work Pressure on Morale, simulation length, and on/off switches for Morale, Work Pressure, and Team Stability. By default, external hiring is turned off while Morale, Work Pressure, and Team Stability initialize on (see Figure 4.26).

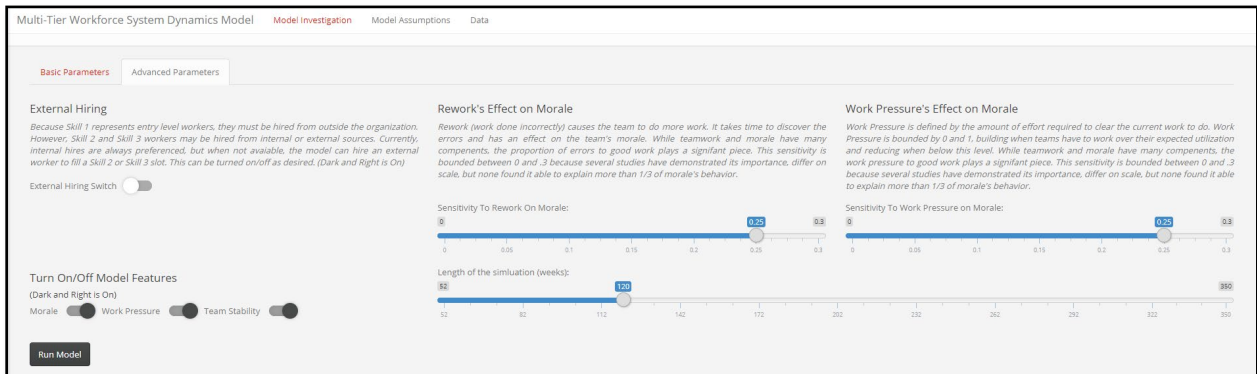


Figure 4.26: Application Advanced Parameters

### 4.2.2.2 Visualizations

Graphics help the user quickly understand the emergent behavior of the selected parameters within the model. Because of the large number of variables, the output was visualized in several graphs split between Simple Work, Complex Work, and Workforce. Each of the graphs are interactive. Users can remove/add variables from the graph by clicking the variable in the legend. Hovering over graph also triggers a tooltip that provides the value of each variable at the timestep of interest.

Because there are two separate types of work in the model (Simple, Complex) they are represented along with the elements of team performance (Work Pressure, Morale, Error Fraction) that correspond to their work group.

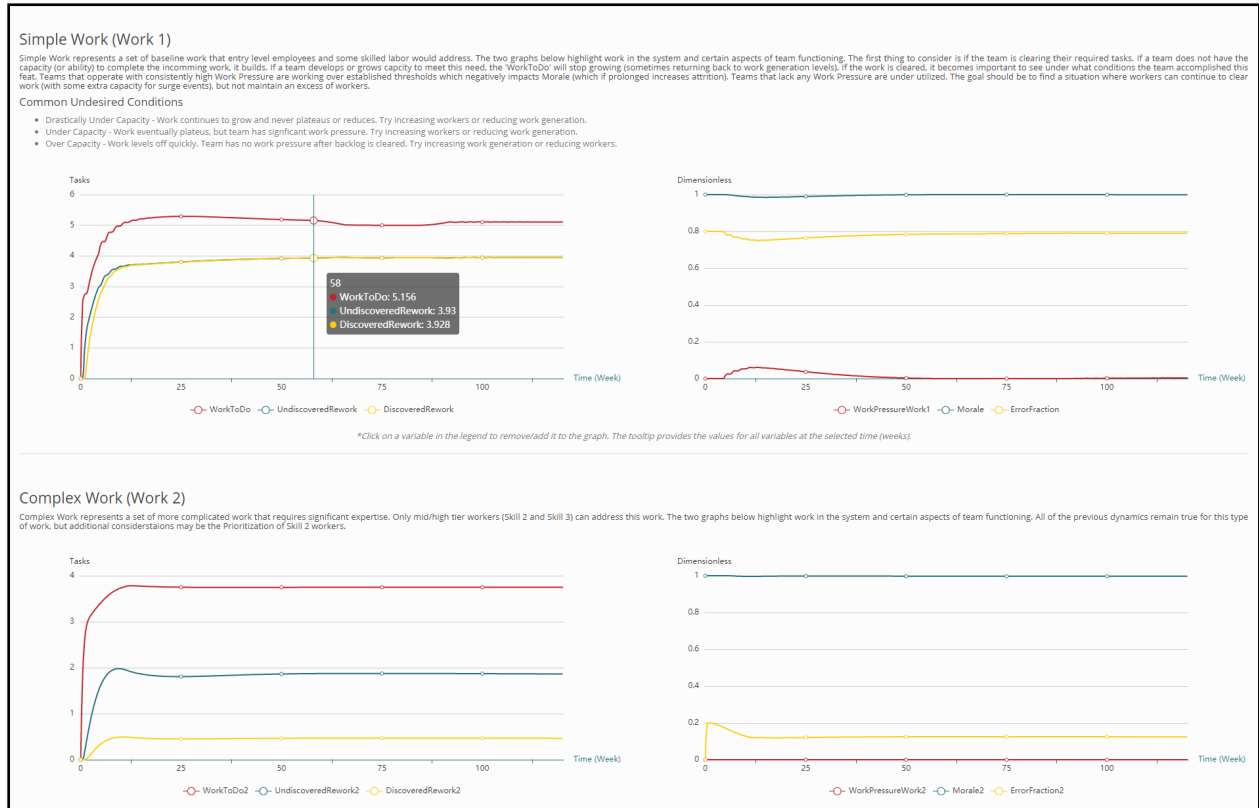


Figure 4.27: Work and Team Function Visualization

Below the work visualizations the user will find a graphic that depicts the number of people in the simulation, Figure 4.28. This visualization contains the same tooltip and remove/add feature as the previous graphs.

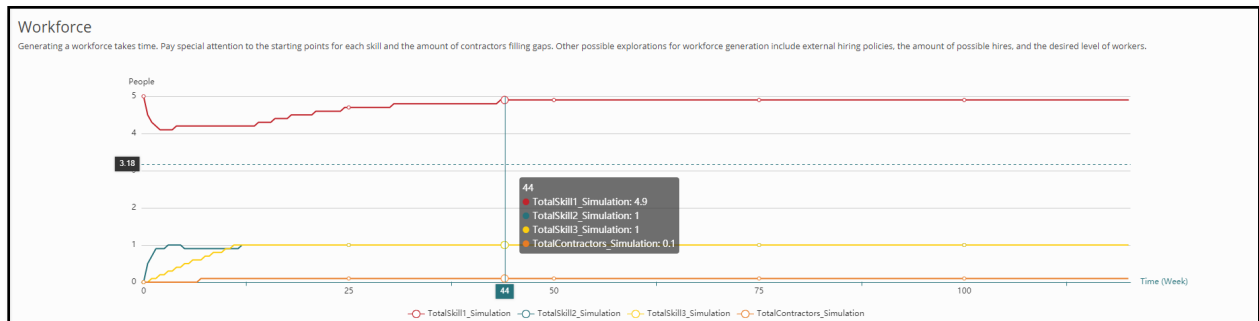
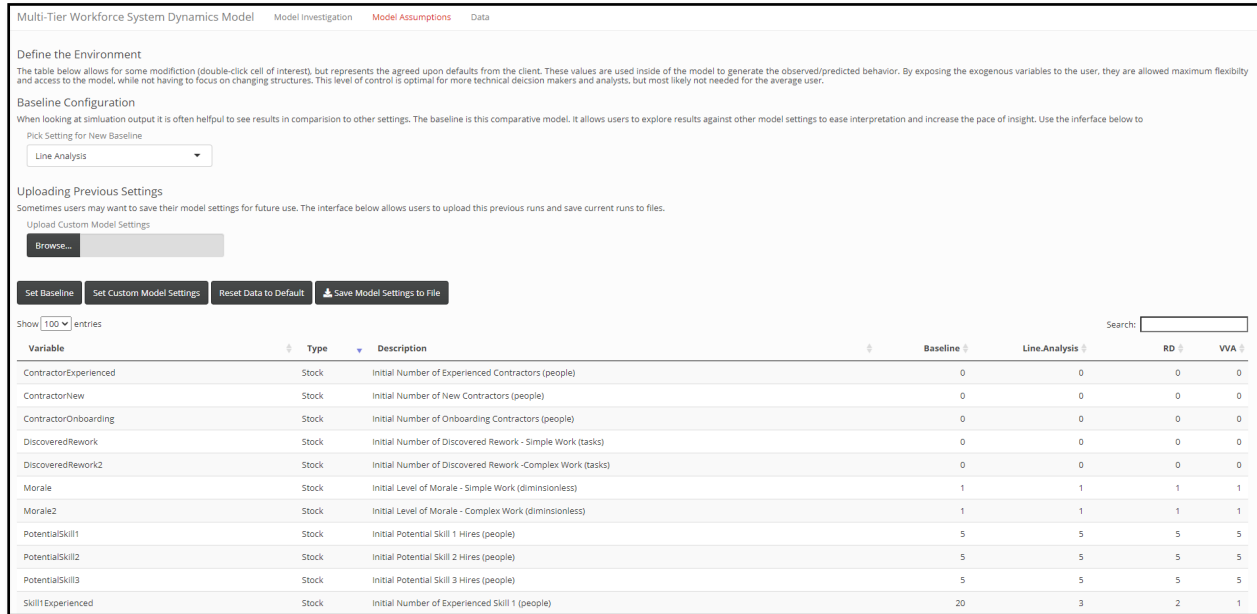


Figure 4.28: Workforce Visualization

### 4.2.3. Model Assumptions

The second page is Model Assumptions. Under this page a user can access all the initial conditions and exogenous variables within the model. This part of the model is intended for users that fully understand the model and want to define special conditions or environments to simulate.



The screenshot shows the 'Model Assumptions' page of the 'Multi-Tier Workforce System Dynamics Model'. It includes sections for 'Define the Environment', 'Baseline Configuration', and 'Uploading Previous Settings'. A table lists various variables with their types and descriptions, and columns for settings in 'Baseline', 'Line Analysis', 'RD', and 'VVA' modules.

Variable	Type	Description	Baseline	Line Analysis	RD	VVA
ContractorExperienced	Stock	Initial Number of Experienced Contractors (people)	0	0	0	0
ContractorNew	Stock	Initial Number of New Contractors (people)	0	0	0	0
ContractorOnboarding	Stock	Initial Number of Onboarding Contractors (people)	0	0	0	0
DiscoveredRework	Stock	Initial Number of Discovered Rework - Simple Work (tasks)	0	0	0	0
DiscoveredRework2	Stock	Initial Number of Discovered Rework - Complex Work (tasks)	0	0	0	0
Morale	Stock	Initial Level of Morale - Simple Work (dimensionless)	1	1	1	1
Morale2	Stock	Initial Level of Morale - Complex Work (dimensionless)	1	1	1	1
PotentialSkill1	Stock	Initial Potential Skill 1 Hires (people)	5	5	5	5
PotentialSkill2	Stock	Initial Potential Skill 2 Hires (people)	5	5	5	5
PotentialSkill3	Stock	Initial Potential Skill 3 Hires (people)	5	5	5	5
Skill1Experienced	Stock	Initial Number of Experienced Skill 1 (people)	20	3	2	1

Figure 4.29: Application Model Assumptions

#### 4.2.3.1 Changing Model Settings

The table seen in Figure 4.29 contains all variables used to define the model. Each variable has a level for the different modules (Baseline, Line Analysis, R&D, and VVA). They can be changed by double-clicking the number the user wants to adjust. This makes the cell active and allows the user to provide new input.

#### 4.2.3.2 Picking and Setting a Baseline

To facilitate the ability to quickly define a baseline model, the application provides a drop-down box labeled "Pick Setting for New Baseline." To set the baseline, a user picks the module of interest (Line Analysis, R&D, or VVA) in the drop-down and then clicks the "Set Baseline" button. They will see the number under the Baseline column update after completing these steps. Individual numbers can still be adjusted for the baseline after this process, but it enables a user to quickly set a new baseline.

#### 4.2.3.3 Saving Model Settings

If the user wants to save the model settings for future use, they can save them by clicking the "Save Model Settings to File" button. This opens up a navigation window to allow the user to decide where to save the file and what to call it. The file format is comma separated value (csv).

#### 4.2.3.4 Uploading Custom Model Settings/Resetting to Default

To upload custom settings, a user clicks "Browse" under the Uploading Previous Settings section of the page and selects the csv of model settings. After uploading the file, the user must click "Set Custom Model Settings." This will finish the process and change all the data in the table to match the uploaded file. If the



user wants to reset the model back to default, they can click “Reset Data to Default” to restore the variable initial conditions.

### 4.2.4. Data

Users can access the data from a simulation run by navigating the Data page of the application. Here they will find a dynamic table that allows them to view all variable output. This data can also be saved for future analysis by clicking the “Download Data” button. This will open a navigation window where the user can decide where to save and what to call the file. The file format is csv.

time	ContractorExperienced	ContractorNew	ContractorOnboarding	DiscoveredRework	DiscoveredRework2	Morale	Morale2	PotentialSkill1	PotentialSkill2	PotentialSkill3	Skill1Experienced	Skill1New	Skill1Old
1	0	0	0	0	0	1	1	5	5	5	3	2	
2	0.5	0	0	0.0041666666666667	0	0	1	1	9.5	4.5	4.458333333333333	2.48741025641026	1.999555555555556
3	1	0	0.0001736111111111111	0.00815500356125356	0	0	1	1	13.5290402421652	4.05	3.972569444444444	2.27303853967666	1.97132704254264
4	1.5	0.0000015914342685192	0.000511811491894811	0.0118837832311375	0.4999999999999998	0.0234374999953125	1	1	17.1455247737843	3.645	3.53747974537028	2.17749445084008	1.94028863072821
5	2	0.00000629117897574923	0.00100224948888997	0.0153451216870566	1.0499831278112	0.0649822195036344	1	1	20.3963256697927	3.2805	3.14817127519476	2.14210987171178	1.91025612468135
6	2.5	0.0000162641693195337	0.00163157792910301	0.0185575300454079	1.47993798469189	0.115344644091236	1	1	23.3206763565353	2.95245	2.80007063462038	2.13614899510506	1.87922084435688
7	3	0.0000339817438782866	0.00238688747098643	0.0215429712199211	1.82285063726561	0.167627436342566	1	1	25.952406665698	2.657205	2.48907436232903	2.14417283039299	1.84704096485088
8	3.5	0.000061587142824245	0.00325648110107203	0.0243245981175242	2.11825701774971	0.217850914919235	1	1	28.321264852672	2.3914845	2.21147755073634	2.1578215746607	1.81458417994171
9	4	0.000100974365782882	0.0042298489603916	0.0269246222808305	2.37685473437914	0.264073180630959	1	1	30.4536939498544	2.15233605	1.96391459670552	2.17290011268994	1.78249746270098
10	4.5	0.000153740487014087	0.00529767992128938	0.0293623978267596	2.60069715067556	0.305536759120319	1	1	32.3733411468601	1.937102445	1.7433362674432	2.18743635110319	1.75124805176455

Figure 4.30: Application Data Page

## 4.3. Benchmarking

Perhaps the most important part of leveraging this model is choosing good benchmarks. Because the model’s structure remains set, the initial conditions generate the changes in behavior between runs. When choosing benchmarks, it is important to understand that you are determining the historical nominal for this effects-based model. Increases or decreases in work pressure, error rates, morale ... should be considered as a deviation from this historical nominal. Picking good benchmarks tunes the model to the environment in question. Bad benchmarks will yield bad data while good benchmarks can yield insights.

### 4.3.1. Workers

The model contains three levels of workers (Skill1, Skill2, and Skill3). Each of them contains individual parameters that can be adjusted. The most important parameters are the error rates and the tasks required to gain experience. The error rate is the fractional error generated by a worker, or how much of their work results in rework. The number of tasks required to gain experience represents how much correctly executed work is required before a worker transitions from inexperienced to experienced. This is important because only experienced workers can be promoted in this model. The other workers parameters remain important, but these two variables significantly impact model outcome and should be considered heavily.

### 4.3.2. Work

Work is broken up into two types (Simple, Complex). To help understand the work environment, it is important to define the ratio of these two types of works when it comes to project. Once a few workers have been selected and benchmarked, this ratio can be calculated. In this example simulation, workers’ error rates were benchmarked and annotated as Good Tasks/All Tasks each work module and type of work. We



kept Tasks/Week and Weeks/Year constant to enable easy communication between modules. We then asked about how many Projects/Year a worker could complete or oversee. This then allowed us to calculate how many Projects/Task, yielding its reciprocal Tasks/Project. With this process we are then able to leverage this information in the simulation in a more meaningful way. The application asks a user what module they are simulating (Line Analysis, R&D, or VVA) and how many projects they would like to accomplish. The model then conducts the calculations to generate constant work (in tasks) that would enable a team (if capable) to complete the desired number of projects.

Simple Work	Good Tasks	Tasks	Weeks	Projects	=	Projects	Tasks/Project
	All Tasks	Week	Year	Task		Year	
Line Analysis	0.2	5	52	0.384615385		20	2.6
R&D	0.8	5	52	0.036057692		7.5	27.7
VVA	0.2	5	52	0.009615385		0.5	104
Complex Work	Good Tasks	Tasks	Weeks	Projects	=	Projects	Tasks/Project
	All Tasks	Week	Year	Task		Year	
Line Analysis	0.9	5	52	0.512820513		120	1.95
R&D	0.9	5	52	0.320512821		75	3.12
VVA	0.9	5	52	0.008547009		2	117

Figure 4.31: Benchmarking Work

## 4.4. Model Execution and Interpretation

### 4.4.1. Scenario Development

Because of the flexibility of the model, it is most helpful to define simple scenarios for interrogation and work on refinements. These simple scenarios are generally easy to define and can garner quick insights. As a user narrows down their trade space they can explore more nuanced alternatives. An example scenario is shown below in Figure 4.32. This scenario will be used to help illustrate the rest of the chapter.

**Simple Scenario 1**

Component: Line Analysis

Unique Variables

- Population
  - Skill 1 (Starting 5, Desired 8)
  - Skill 2 (Starting 2, Desired 3)
  - Skill 3 (Starting 0, Desired 1)
  - Contractor Fill Rate: 0-10% of Skill Gap
  - External Hires (No – Only Skill1)
- Work
  - Project Ratio: 1.3̄  
 -(2.6 Simple Tasks/Project, 1.95 Complex Tasks/Project)
  - Desired Number of Projects a Year: ~100, 120, 140

Figure 4.32: Simple Scenario 1

### 4.4.2. Setting Parameters

Once a scenario has been defined, it must be established in the Model Assumptions page of the application. If an analyst has defined a custom scenario or deviated from the default sufficiently, it may be easier to upload custom model settings as described in 4.2.3.4 as opposed to changing individual numbers.

### 4.4.3. Over Capacity / Under Capacity

Using the Simple Scenario starting conditions (5 x Skill 1, 2 x Skill 2, 0 x Skill 3) we start exploring team performance. These settings yield output graphs seen in Figure 4.33. In this figure we see that the teams manage to complete both sets of work. However, we notice slight turbulence in Work To Do as well as Work Pressure and Error Fraction in Team 1 (Simple Work). These variables interact with each other in the following way – as Work To Do increases, Work Pressure increases, this shifts Skill 2 works from Complex Work (their default) to Simple work, clearing the growing backlog of tasks, and reducing the Team’s weighted Error Fraction. Once the Work To Do is back under control, the Skill 2 workers shift back to Complex work. Because the Skill 1 workers lack the capacity to complete the simple work on their own, we see this emergent sawtooth behavior. Despite this shifting, this team as at or over capacity as Work Pressure for neither team deviates significantly from nominal and Work To Do stabilizes.



Figure 4.33: Scenario 1 Output – Starting Staff at 100 Projects/Year

The same staff faces significant challenges when they attempt to complete 180 projects a year, see Figure 4.34. In these graphs we see that Team 1 copes with the increased workload, but still relies on Skill 2 workers to cover down on occasion. However, Team 2 is unable to maintain the needed pace with Complex work. The Work To Do grows constantly while Work Pressure reaches its maximum level (1) and Morale

drops towards .7 or 70% of nominal. This is a clear situation of under capacity as Complex Work To Do grows without bound and Work Pressure for Team 2 remains at unsustainable levels.

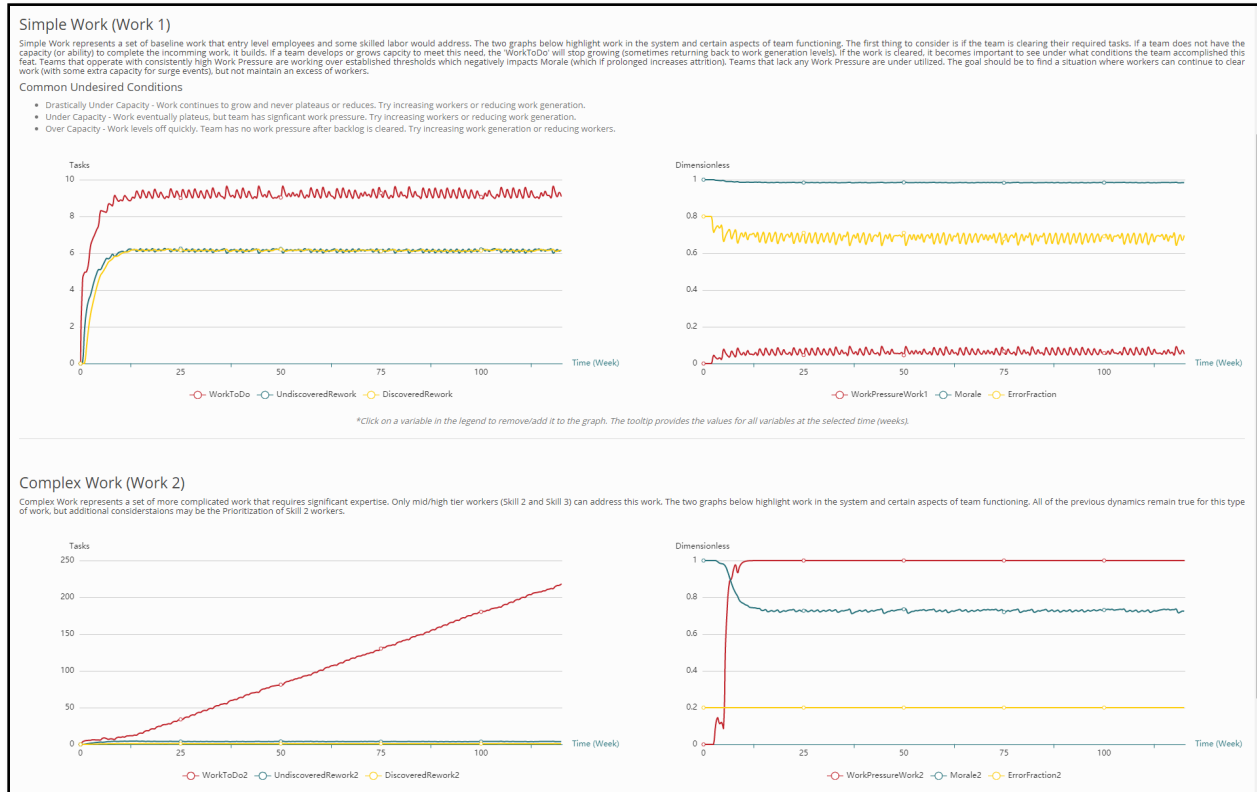


Figure 4.34: Scenario 1 Output - Starting Staff at 180 Projects/Year

#### 4.4.4. Work Prioritization

Under the basic parameter tab of the Model Investigation page, users can define Skill 2 worker's prioritization. This term acts as a sensitivity for work assignment as Work Pressure from Team 1, in conjunction with this term, modulates how much Skill 2 workers focus on Simple Work, see Equation 2.

$$\# \text{ of Skill 2 on Work 1} = \text{Work Pressure}(\text{Team 1}) * \text{Skill 2 Prioritization} * \# \text{ of Skill 2 Workers} \quad (2)$$

This interaction can help balance a team's efforts but results in the turbulence we observed in the previous output. Eventually, this prioritization cannot balance the work and one, or both, of the teams succumb to unsustainable Work Pressure. Figure 4.35 demonstrates the effect of this balancing as the usage of Skill 2 workers controls the work accumulation on Simple Work but requires significant spikes in Work Pressure to clear Complex Work backlogs.

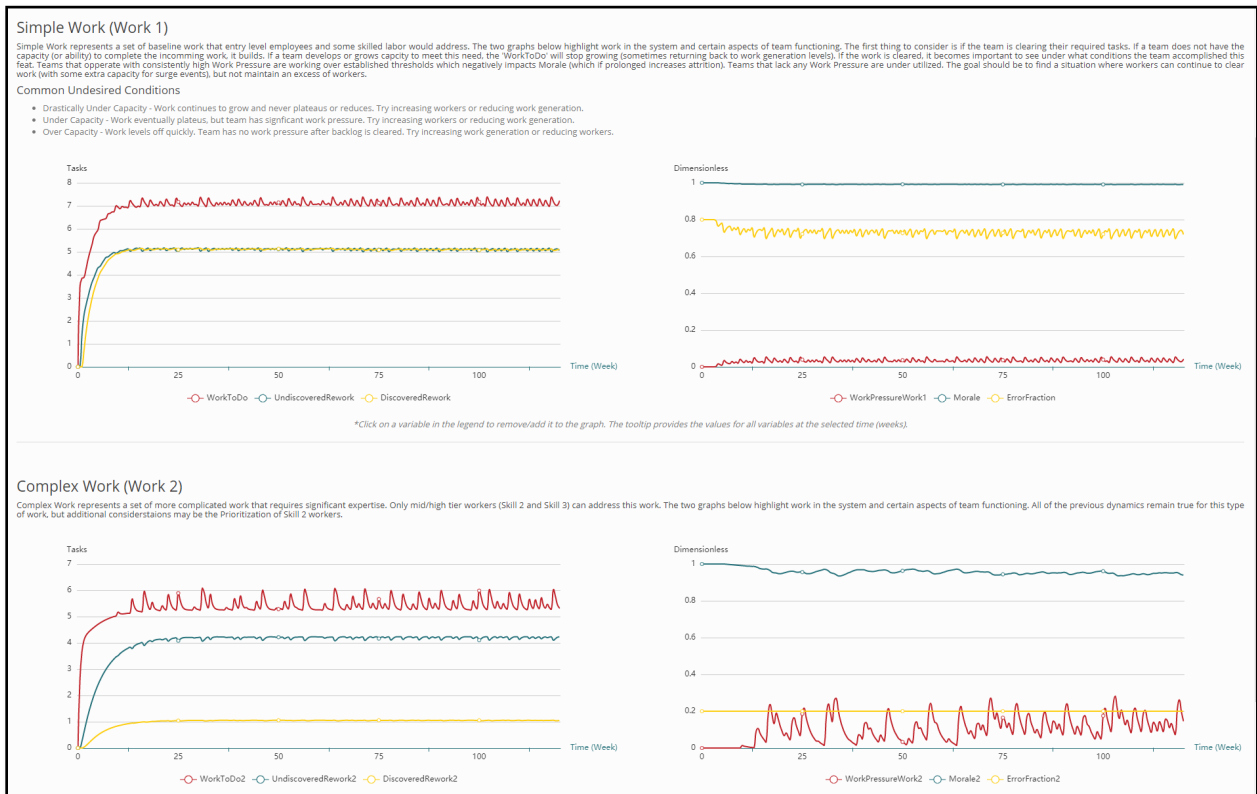
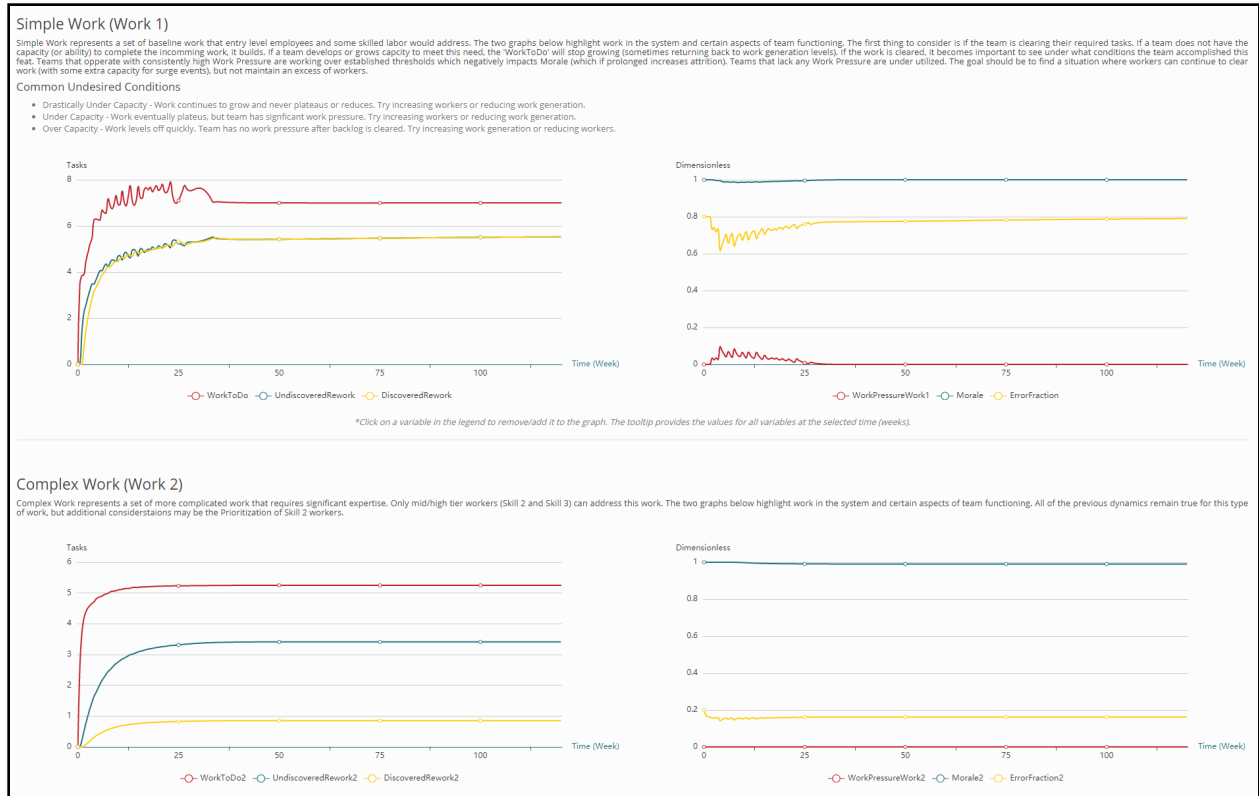


Figure 4.35: Scenario 1 Output - Starting Staff at 140 Projects/Year

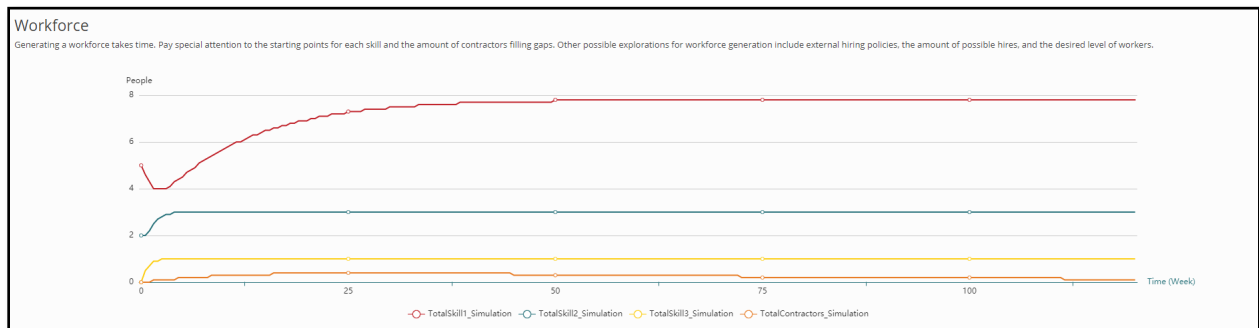
#### 4.4.5. The Effect of Growing the Workforce

The previous examples all explored options that had fixed workforce structures; however, most alternatives look at changes to this structure. Because of the hiring policies and delays in that process, this change in personnel creates emergent behavior that must be considered when designing an organizational adjustment. Consider the impacts of increasing our staffing levels from the starting state (5 x Skill 1, 2 x Skill 2, 0 x Skill 3) to the desired state (8 x Skill 1, 3 x Skill 2, 1 x Skill 3) under the 140 Projects/Year demand illustrated in Figure 4.35. With this new organizational set, we notice a unique anticipated behavior in Figure 4.36. Team 1 (Simple Work) displays the turbulent performance seen previously at the beginning of the change, but eventually reaches steady-state operations with nominal Work Pressure. Team 2 (Complex Work) experiences no increase in Work Pressure and only mild turbulence in Error Fraction from the shifting of Skill 2 workers between tasks. Why does Team 1 experience this turbulence up front and take over 25 weeks to reach steady state when we are just adding staff? Why does Team 1's work pressure actually INCREASE when we try to hire more people? Why doesn't Team 2 experience similar turbulence? The answer can be found in Figure 4.37. To fill the Skill 2 and Skill 3 gap created by the change in desired organizational structure, workers must be promoted from within. Perhaps unsurprisingly, we see a drop in the number of Skill 1 workers (from 5 to 4 people) to at the beginning of the simulation because of the difference in hiring processes. Skill 1 workers must be recruited and onboard the organization while internal hires (if available) have a much quicker pipeline. Because of this difference in hiring processes, only Team

1 – exposed to the delays of recruiting and onboarding – experience the significant increase in Work Pressure.



**Figure 4.36: Scenario 1 Output - Desired Staff at 140 Projects/Year - Work**



**Figure 4.37: Scenario 1 Output - Desired Staff at 140 Projects/Year – People**



## Chapter 5

# Limitations and Future Work

### 5.1. Limitations

Interpreting model output remains challenging. This is an effects-based model that enables comparison of simulations against baselines. The accuracy of the effect remains relative to the accuracy of the parameter values. Users must understand the limitations of benchmarks and proxies on model calibration.

The sensitivities surrounding Morale are difficult to calibrate and can be overpowered by many different factors (finding meaning in work, good leadership, culture ...). Good benchmarks enable the user to encapsulate some of these hard to understand features but cannot be generalized broadly. For example, a team that remains resilient to increased workload/work pressure because they believe in their projects have high trust in their leadership would not serve as a good benchmark for a team with traditionally low morale.

Currently the model only works with point estimates. This creates a condition where tipping points and the dynamics of small teams appear amplified. In truth, we understand that elements like a person's error fraction, productivity, and onboarding are not static numbers. The model does not account for this.

Time inside the model fails to reflect impacts from the calendar (holidays, leave, graduations) or fiscal cycles (possible work impacts). The model treats every week as nominal and does not deviate.

### 5.2. Future Work

Adding distributions to appropriate parameters would help estimate the range of outcomes and behavior. It would probably be sufficient to add distributions to some of the key variables such as work arrival, error fractions, and productivity as these have significant impacts in the model. It would also be beneficial to explore the resiliency of these work structures to common occurrences (work pulses, temporary loss of personnel, changes in assignment policy ...). Developing an additive value model may also help decision makers capture what a 'good' outcome consists of and how to make tradeoffs.



## Chapter 6 Conclusion

This project sought to provide the DIA with a tool to help explore and identify the amount of high skilled labor needed to execute anticipated work demands. It leveraged system dynamics, R, and shiny to create a tailorable multi-tiered workforce and work complexity model that is capable of simulating several different work environments. This method has advantages over other discrete workforce optimization tools because it enables a decision maker to see the changes in performance and capacity across time and as an organization grows into a capacity. Furthermore, this work demonstrates the added benefits of disaggregating workers and work into sub-categories to predict emergent behavior. Because of these things it adds a new perspective to the vibrant research area of organizational change and workforce modeling found within operations research.



## Appendix A References

- [1] J. W. Forrester, *Principles of Systems*, Cambridge, MA: Wright-Allen Press, Inc., 1968.
- [2] J. W. Forrester, *Industrial Dynamics*, Waltham, MA: Pegasus Communications Inc, 1961.
- [3] J. W. Forrester, *World Dynamics*, Cambridge, MA: Wright-Allen Press, Inc., 1971.
- [4] J. Sterman, *Business Dynamics: Systems Thinking and Modeling for a Complex World*, Boston, MA: Irwin McGraw-Hill, 2000.
- [5] J. M. Lyneis and D. N. Ford, "System dynamics applied to project management: a survey, assessment, and directions for future research," *System Dynamics Review*, vol. 23, pp. 157-189, 2007.
- [6] K. G. Cooper, "Naval Ship Production: A Claim Settleed And A Framework Built," *INTERFACES*, vol. 10, December 1980.
- [7] R. Oliva and J. Sterman, "Cutting Corners and Working Overtime: Quality Erosion in the Service Industry," *Management Science*, vol. 47, pp. 894-914, 2001.
- [8] R. Oliva and J. Sterman, "Death Spirals and Virtuous Cycles: Human Resource Dynamics in Knowledge-Based Services," in *Handbook of Service Science*, New York, Springer, 2010, pp. 321-358.
- [9] N. P. Repenning and J. D. Sterman, " Nobody ever gets credit for fixing problems that never happened: Creating and sustaining process improvement.," *California Management Review*, vol. 43, no. 4, pp. 64-88, 2001.
- [10] N. P. Repenning and J. D. Sterman, " Capability Traps and Self-Confirming Attribution Errors in the Dyanmics of Process Improvement," *Administrative Science Quaterly*, vol. 47, no. 2, pp. 265-295, 2002.
- [11] P. Gonçalves, "Balancing provision of relief and recovery with capacity building in humanitarian operations," *Operations Management Research*, vol. 4, no. 1-2, pp. 39-50, 2011.
- [12] J. Guevara, M. J. Garvin and N. Ghaffarzadegan, "Capability trap of the US highway system: Policy and management implications," *Journal of Management in Engineering*, vol. 33, no. 4, 2017.
- [13] E. Landry and J. D. Sterman, "The capability trap: prevalence in human systems," in *35th International Conference of the System Dynamics Society*, 2017.
- [14] S. Armenia, A. Centra and C. Retrosi, "Military workforce dynamics and planning in the Italian airforce," in *30th International Conference of System Dynamics Society*, 2012.
- [15] M. Johnstone, V. Le, A. Novak and B. Khan, "A Multi-level approach to planning and scheduling resources for aviation training," in *21st Internation Congress on Modeling and Simulation*, 2015.



- 
- [16] J. A. Wang, "A System Dynamics Simulation Model for a Four-rank Military Workforce," DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION EDINBURGH (AUSTRALIA) LAND OPERATIONS DIV, 2007.
- [17] E. Lawler and L. Porter, "The Effect of Performance on Job Satisfaction," *Industrial Relations*, vol. 7, no. 1, pp. 20-28, 1967.
- [18] T. Judge, C. Thoresen, J. Bono and G. Patton, "The Job Satisfaction-Job Performance Relationship: A Qualitative and Quantitative Review," *Psychological Bulletin*, vol. 127, no. 3, pp. 376-407, 2001.
- [19] S. DeRue and F. Morgenson, "Stability and Change in Person-Team and Person-Role Fit Over Time: Effects of Growth Satisfaction, Performance, and General Efficacy," *Journal of Applied Psychology*, vol. 92, no. 5, pp. 1242-1253, 2007.
- [20] E. Gonzalez-Mule, B. Cockburn, B. McCormick and P. Zhao, "Team tenure and team performance: A meta-analysis and process model," *Personnel Psychology*, vol. 73, pp. 151-198, 2020.
- [21] C. Harris, G. McMahan and P. Wright, "Talent and time together: The impact of human capital and overlapping tenure on unit performance," *Personnel Review*, vol. 41, no. 4, pp. 408-427, 2012.
- [22] F. Montanari, G. Silvestri and E. Gallo, "Team Performance Between Change and Stability: The Case of the Italian 'Serie A'," *Journal of Sport Management*, vol. 22, pp. 701-716, 2008.
- [23] C. Savelsbergh, R. Poell and B. van der Heijden, "Does team stability mediate the relationship between leadership and team learning? An empirical study among Dutch project teams," *International Journal of Project Management*, vol. 33, no. 2, pp. 406-418, 2015.
- [24] J. Sterman, "Learning for Ourselves: Interactive Simulations to Catalyze Science-Based Environmental Activism," in *Science-Based Activism*, Bergen, Fagbokforlaget, 2015, pp. 253-279.
- [25] R. Ihaka and R. Gentleman, "R: A Language for Data Analysis and Graphics," *Journal of Computational and Graphical Statistics*, vol. 5, no. 3, pp. 299-314, September 1996.
- [26] J. Duggan, *System Dynamics Modeling in R*, Switzerland: Springer, 2016.
- [27] H. Wickham, M. Averick, J. Bryan, W. Chang, L. D'Agostino McGowan, R. François, G. Grolemund, A. Hayes, L. Henry, J. Hester, M. Kuhn, T. Lin Pedersen, E. Miller, S. Milton Bache, K. Müller, J. Ooms, D. Robinson, D. Paige Seidel, V. Spinu, K. Takahashi, D. Vaughan, C. Wilke, K. Woo and H. Yutani, "Welcome to the tidyverse.," *Journal of Open Source Software*, vol. 4, p. 1686, 2019.
- [28] J. Duggan, "Input and output data analysis for system dynamics modelling using the tidyverse libraries of R," *System Dynamics Review*, vol. 34, no. 3, pp. 438-461, 2018.
- [29] J. Duggan, "Using R libraries to facilitate sensitivity analysis and to calibrate system dynamics models," *System Dynamics Review*, vol. 35, no. 3, pp. 255-282, July 2019.
- [30] H. Rahmandad and J. Sterman, "Reporting guidelines for simulation-based research in social sciences," *System Dynamics Review*, pp. 396-411, 2012.



- [31] S. Morris, *shinySIR: Interactive Plotting for Mathematical Models of Infections Disease Spread*, 2020.
- [32] J. Lyneis, K. Cooper and S. Els, "Strategic management of complex projects: a case study using system dynamics," *System Dynamics Review*, vol. 17, no. 3, pp. 237-260, 2001.
- [33] K. Soetaert, T. Petzoldt and R. Setzer, "Solving differential equations in R: package deSolve," *Journal of Statistical Software*, no. 33, 2010.



## Appendix B Workforce Model Equations

This appendix contains the equations utilized to execute the project's model. The application runs two versions of the model for baseline comparisons.

**Table 5.1** Multi-Tier Workforce and Complexity Equations

Variable Name	Equation	Variable Type
LoosingWorkPressureWork1	$\text{WorkPressureWork1} / \text{TimeToRelievePressureWork1}$	Auxiliary
Skill2ExperiencedOnWork1	$\min(\text{Skill2Experienced}, \max(0, \text{Work1Skill2Sensitivity} * (0.9) * (\text{WorkPressureWork1})) * \text{Skill2Experienced})$	Auxiliary
Skill2NewOnWork1	$\min(\text{Skill2New}, \max(0, (\text{WorkPressureWork1} * \text{Work1Skill2Sensitivity})) * \text{Skill2New})$	Auxiliary
Workforce	$\text{Skill1Experienced} + \text{Skill1New} + \text{Skill2ExperiencedOnWork1} + \text{Skill2NewOnWork1} + \text{ContractorExperienced} + \text{ContractorNew}$	Auxiliary
TeamWeightedErrorRate	$(\text{Skill1New} * \text{ErrorRateSkill1new} + \text{Skill1Experienced} * \text{ErrorRateSkill1exp} + (\text{Skill2NewOnWork1} + \text{ContractorNew}) * \text{ErrorRateSkill2new} + (\text{Skill2ExperiencedOnWork1} + \text{ContractorExperienced}) * \text{ErrorRateSkill2exp}) / (\text{Skill1New} + \text{Skill1Experienced} + \text{Skill2NewOnWork1} + \text{Skill2ExperiencedOnWork1} + \text{ContractorNew} + \text{ContractorExperienced} + .0000000001)$	Auxiliary
NominalCapacityWork1	$\text{NominalProductivity} * \text{Workforce} * (1 - \text{TeamWeightedErrorRate})$	Auxiliary
BuildingWorkPressureWork1	$\max(0, \min(\text{MaxWorkPressure} - \text{WorkPressureWork1} + \text{LoosingWorkPressureWork1}, ((\text{WorkToDo}) / (\text{NominalCapacityWork1} + .0000000001)) - \text{ExpectedUtilization})) / \text{TimeToFeelPressureWork1}$	Auxiliary
LoosingWorkPressureWork2	$\text{WorkPressureWork2} / \text{TimeRoRelievePressureWork2}$	Auxiliary
Skill2NewOnWork2	$\text{Skill2New} - \text{Skill2NewOnWork1}$	Auxiliary



Variable Name	Equation	Variable Type
Workforce2	$Skill3New + Skill3Experienced + Skill2NewOnWork2 + Skill2ExperiencedOnWork2$	Auxiliary
TeamWeightedErrorRate2	$(Skill3Experienced * ErrorRateSkill3exp + Skill3New * ErrorRateSkill3new + Skill2NewOnWork2 * ErrorRateSkill2new2 + Skill2ExperiencedOnWork2 * ErrorRateSkill2exp2) / (Skill3New + Skill3Experienced + Skill2NewOnWork2 + Skill2ExperiencedOnWork2 + .0000000001)$	Auxiliary
NominalCapacityWork2	$Workforce2 * NominalProductivity2 * (1 - TeamWeightedErrorRate2)$	Auxiliary
BuildingWorkPressureWork2	$\max(0, \min(\text{MaxWorkPressure2} - \text{WorkPressureWork2} + \text{LoosingWorkPressureWork2}, ((\text{WorkToDo2}) / (\text{NominalCapacityWork2} + .0000000001)) - \text{ExpectedUtilization2})) / \text{TimeToFeelPressureWork2}$	Auxiliary
ReworkDiscoveryRate	$\text{UndiscoveredRework} / \text{NominalTimeToDiscoveryRework}$	Auxiliary
Productivity	$\text{NominalProductivity} * (1 - \text{TeamInstability}) * (1 + \text{WorkPressureWork1})$	Auxiliary
ErrorFraction	$\text{TeamWeightedErrorRate}$	Auxiliary
Progress	$\max(0, \min(\text{Workforce} * \text{Morale} * \text{Productivity} * (1 - \text{ErrorFraction}), \text{WorkToDo} / \text{MinTimeToCompleteTask}))$	Auxiliary
ErrorGeneration	$\text{Progress} * \text{ErrorFraction}$	Auxiliary
ReworkProportion	$\text{ReworkDiscoveryRate} / (\text{Progress} + \text{ErrorGeneration} + .0000000001)$	Auxiliary
ChangingMorale	$(\min(\text{CapacityMorale}, \max(0, (\text{CapacityMorale} - (\text{CapacityMorale} * (\max(0, \text{ReworkProportion} - \text{ExpectedReworkProportion}))) * \text{SensitivityToReworkOnMorale}) - (\text{CapacityMorale} * (\text{WorkPressureWork1}) * \text{SensitivityToWorkPressureOnMorale})))) - \text{Morale}) / \text{MoraleAdjustmentTime}$	Auxiliary
ReworkDiscoveryRate2	$\text{UndiscoveredRework2} / \text{NominalTimeToDiscoveryRework2}$	Auxiliary
Productivity2	$\text{NominalProductivity2} * (1 - \text{TeamInstability}) * (1 + \text{WorkPressureWork2})$	Auxiliary
ErrorFraction2	$\text{TeamWeightedErrorRate2}$	Auxiliary
Progress2	$\max(0, \min(\text{Workforce2} * \text{Morale2} * \text{Productivity2} * (1 - \text{ErrorFraction2}), \text{WorkToDo2} / \text{MinTimeToCompleteTask2}))$	Auxiliary
ErrorGeneration2	$\text{Progress2} * \text{ErrorFraction2}$	Auxiliary
ReworkProportion2	$\text{ReworkDiscoveryRate2} / (\text{Progress2} + \text{ErrorGeneration2} + .0000000001)$	Auxiliary



Variable Name	Equation	Variable Type
ContractExpirationRate	$\text{ContractorExperienced} / \text{AverageTimeLeftOnContract}$	Auxiliary
ProportionOfContractorsInexperience d	$\text{ContractorNew} / (\text{ContractorNew} + \text{ContractorExperienced} + .0000000001)$	Auxiliary
ContractorGainingExperience	$\text{ContractorNew} * \text{ProportionOfContractorsInexperienced} * \text{Progress} / \text{TasksRequiredToGainExperienceContractor}$	Auxiliary
TotalSkill1	$\text{Skill1Experienced} + \text{Skill1New} + \text{Skill1Onboarding}$	Auxiliary
Skill1Gap	$\max(0, \text{DesiredSkill1} - \text{TotalSkill1})$	Auxiliary
TotalSkill2	$\text{Skill2Experienced} + \text{Skill2New} + \text{Skill2Onboarding}$	Auxiliary
Skill2Gap	$\max(0, \text{DesiredSkill2} - \text{TotalSkill2})$	Auxiliary
TotalContractors	$\text{ContractorExperienced} + \text{ContractorNew} + \text{ContractorOnboarding}$	Auxiliary
ContractorsNeeded	$\max(0, \text{Skill1Gap} + \text{Skill2Gap} - \text{TotalContractors}) * \text{ContractorFillPercentage}$	Auxiliary
ContractorHiring	$\text{ContractorsNeeded} / \text{ContractorHiringDelay}$	Auxiliary
ContractorOnboardingRate	$\text{ContractorOnboarding} / \text{ContractorAverageTimeToOnboard}$	Auxiliary
EnteringMarketSkill1	NominalMarketEntranceSkill1	Auxiliary
EnteringMarketSkill2	NominalMarketEntranceSkill2	Auxiliary
EnteringMarketSkill3	NominalMarketEntranceSkill3	Auxiliary
ErrorGeneration2	ErrorGeneration	Auxiliary
ErrorGeneration22	ErrorGeneration2	Auxiliary
LeavingMarketSkill1	$\text{PotentialSkill1} / \text{AverageTimeInMarketSkill1}$	Auxiliary
LeavingMarketSkill2	$\text{PotentialSkill2} / \text{AverageTimeInMarketSkill2}$	Auxiliary
LeavingMarketSkill3	$\text{PotentialSkill3} / \text{AverageTimeInMarketSkill3}$	Auxiliary
Promtions1to2	$\max(0, \min(\text{Skill1Experienced}, \text{Skill2Gap}))$	Auxiliary
NewHiresNeededSkill2	$\text{Skill2Gap} - \text{Promtions1to2}$	Auxiliary
TotalSkill3	$\text{Skill3Experienced} + \text{Skill3New} + \text{Skill3Onboarding}$	Auxiliary
Skill3Gap	$\max(0, \text{DesiredSkill3} - \text{TotalSkill3})$	Auxiliary
Promtions2to3	$\max(0, \min(\text{Skill2Experienced}, \text{Skill3Gap}))$	Auxiliary
NewHiresNeededSkill3	$\text{Skill3Gap} - \text{Promtions2to3}$	Auxiliary
Skill3OnboardingRate	$\text{Skill3Onboarding} / \text{AverageTimeToOnboard}$	Auxiliary
Skill2OnboardingRate	$\text{Skill2Onboarding} / \text{AverageTimeToOnboard}$	Auxiliary
Skill1OnboardingRate	$\text{Skill1Onboarding} / \text{AverageTimeToOnboard}$	Auxiliary



Variable Name	Equation	Variable Type
RetirementSkill1	$Skill1Experienced / AverageSkill1Retirement$	Auxiliary
RetirementSkill2	$Skill2Experienced / AverageSkill2Retirement$	Auxiliary
RetirementSkill3	$Skill3Experienced / AverageSkill3Retirement$	Auxiliary
Skill1ExperiencedAttrition	$Skill1Experienced * Skill1QuitFraction * (1 + ((StandardMorale - Morale) / StandardMorale))$	Auxiliary
Skill1NewAttrition	$Skill1New * Skill1NewQuitFraction * (1 + ((StandardMorale - Morale) / StandardMorale))$	Auxiliary
Skill2ExperiencedAttrition	$Skill2Experienced * Skill2QuitFraction * (1 + ((StandardMorale - Morale) / StandardMorale))$	Auxiliary
Skill2NewAttrition	$Skill2New * Skill2NewQuitFraction * (1 + ((StandardMorale - Morale) / StandardMorale))$	Auxiliary
Skill3ExperiencedAttrition	$Skill3Experienced * Skill3QuitFraction * (1 + ((StandardMorale - Morale2) / StandardMorale))$	Auxiliary
Skill3NewAttrition	$Skill3New * Skill3NewQuitFraction * (1 + ((StandardMorale - Morale2) / StandardMorale))$	Auxiliary
PeopleLeavingTeam	$RetirementSkill1 + RetirementSkill2 + RetirementSkill3 + Skill1ExperiencedAttrition + Skill1NewAttrition + Skill2ExperiencedAttrition + Skill2NewAttrition + Skill3ExperiencedAttrition + Skill3NewAttrition + ContractExpirationRate$	Auxiliary
PeopleMoving	$( PeopleJoiningTeam + PeopleLeavingTeam ) / ( 1 + TotalSkill3 + TotalSkill2 + TotalSkill1 + TotalContractors )$	Auxiliary
Promotion1to2	$Promtions1to2 / AverageTimeToTransition1to2$	Auxiliary
Promotion2to3	$Promtions2to3 / AverageTimeToTransition2to3$	Auxiliary
ProportionOfSkill1Inexperienced	$Skill1New / (Skill1New + Skill1Experienced + .000000001)$	Auxiliary
ProportionOfSkill2Inexperienced	$Skill2New / (Skill2New + Skill2Experienced + .000000001)$	Auxiliary
ProportionOfSkill3Inexperienced	$Skill3New / (Skill3New + Skill3Experienced + .000000001)$	Auxiliary
ReworkAssignment	$DiscoveredRework / ReworkAssignmentTime$	Auxiliary
ReworkAssignment2	$DiscoveredRework2 / ReworkAssignmentTime2$	Auxiliary



Variable Name	Equation	Variable Type
Skill1HiringRate	$\max(0, \min(\text{PotentialSkill1} / \text{Skill1HiringDelay}, \text{Skill1Gap} / \text{Skill1HiringDelay}))$	Auxiliary
Skill2GainingExperience	$\text{Skill2New} * \text{ProportionOfSkill2Inexperienced} * \text{Progress} / \text{TasksRequiredToGainExperienceSkill2}$	Auxiliary
Skill2HiringRate	$\max(0, \min(\text{PotentialSkill2} / \text{Skill2HiringDelay}, \text{NewHiresNeededSkill2} / \text{Skill2HiringDelay}))$	Auxiliary
Skill3GainingExperience	$\text{Skill3New} * \text{ProportionOfSkill3Inexperienced} * \text{Progress2} / \text{TasksRequiredToGainExperience}$	Auxiliary
Skill3HiringRate	$\max(0, \min(\text{NewHiresNeededSkill3} / \text{Skill3HiringDelay}, \text{PotentialSkill3} / \text{Skill3HiringDelay}))$	Auxiliary
TeamAssimilation	$\text{TeamInstability} / \text{TeamAssimilationTime}$	Auxiliary
WorkGeneration	LinearWorkGeneration	Auxiliary
WorkGeneration2	LinearWorkGeneration2	Auxiliary
ContractorExperienced	$\text{ContractorGainingExperience} - \text{ContractExpirationRate}$	Stock
ContractorNew	$\text{ContractorOnboardingRate} - \text{ContractorGainingExperience}$	Stock
ContractorOnboarding	$\text{ContractorHiring} - \text{ContractorOnboardingRate}$	Stock
DiscoveredRework	$\text{ReworkDiscoveryRate} - \text{ReworkAssignment}$	Stock
DiscoveredRework2	$\text{ReworkDiscoveryRate2} - \text{ReworkAssignment2}$	Stock
Morale	ChangingMorale	Stock
Morale2	ChangingMorale2	Stock
PotentialSkill1	$\text{EnteringMarketSkill1} - \text{LeavingMarketSkill1} - \text{Skill1HiringRate}$	Stock
PotentialSkill2	$\text{EnteringMarketSkill2} - \text{LeavingMarketSkill2} - \text{Skill2HiringRate}$	Stock
PotentialSkill3	$\text{EnteringMarketSkill3} - \text{LeavingMarketSkill3} - \text{Skill3HiringRate}$	Stock
Skill1Experienced	$\text{Skill1GainingExperience} - \text{Promotion1to2} - \text{RetirementSkill1} - \text{Skill1ExperiencedAttrition}$	Stock
Skill1New	$\text{Skill1OnboardingRate} - \text{Skill1GainingExperience} - \text{Skill1NewAttrition}$	Stock
Skill1Onboarding	$\text{Skill1HiringRate} - \text{Skill1OnboardingRate}$	Stock
Skill2Experienced	$\text{Skill2GainingExperience} - \text{Promotion2to3} - \text{RetirementSkill2} - \text{Skill2ExperiencedAttrition}$	Stock
Skill2New	$\text{Promotion1to2} + \text{Skill2OnboardingRate} - \text{Skill2GainingExperience} - \text{Skill2NewAttrition}$	Stock



Variable Name	Equation	Variable Type
Skill3Experienced	Skill3GainingExperience - RetirementSkill3 - Skill3ExperiencedAttrition	Stock
Skill3New	Promotion2to3 + Skill3OnboardingRate - Skill3GainingExperience - Skill3NewAttrition	Stock
Skill3Onboarding	Skill3HiringRate - Skill3OnboardingRate	Stock
TeamInstability	PeopleMoving - TeamAssimilation	Stock
TotalRework	ErrorGeneration2	Stock
TotalRework2	ErrorGeneration22	Stock
UndiscoveredRework	ErrorGeneration - ReworkDiscoveryRate	Stock
UndiscoveredRework2	ErrorGeneration2 - ReworkDiscoveryRate2	Stock
WorkDone	Progress	Stock
WorkDone2	Progress2	Stock
WorkPressureWork1	BuildingWorkPressureWork1 - LoosingWorkPressureWork1	Stock
WorkPressureWork2	BuildingWorkPressureWork2 - LoosingWorkPressureWork2	Stock
WorkToDo	ReworkAssignment + WorkGeneration - ErrorGeneration - Progress	Stock
WorkToDo2	ReworkAssignment2 + WorkGeneration2 - ErrorGeneration2 - Progress2	Stock





Department of Systems Engineering  
United States Military Academy  
West Point, New York 10096