May 2010

Exploring the Effect of Hiring Rules on Quality in a Large Engineering Company Using System Dynamics

George Michael Raad
Worcester Polytechnic Institute

Nolan Farrall Barrie
Worcester Polytechnic Institute

Follow this and additional works at: https://digitalcommons.wpi.edu/mqp-all

Repository Citation

This Unrestricted is brought to you for free and open access by the Major Qualifying Projects at Digital WPI. It has been accepted for inclusion in Major Qualifying Projects (All Years) by an authorized administrator of Digital WPI. For more information, please contact digitalwpi@wpi.edu.
Exploring the Effect of Hiring Rules on Quality in a Large Engineering Company Using System Dynamics

A Major Qualifying Project Report

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

in System Dynamics

by

Nolan Barrie

George Michael Raad

Submitted: 5/3/2010

Approved:

Prof. Khalid Saeed, Major Advisor
**Introduction**

After perceiving decay in the quality of their products, A Large Engineering Company (ALEC) sought to understand the reasons behind it. Adopting a systemic approach to analyzing the structure and dynamics of the organization was proposed. System dynamics, a novel methodology that relies on feedback loops, causal link, and stocks and flows, clearly is an appropriate tool for shedding light and stimulating insight, from a holistic point of view, on the dynamics that may have caused the quality decay. The model illustrates ALEC’s workforce supply chain and how its structure affects the production’s quality.

This paper presents the steps that WPI and ALEC undertook in this project. After identifying the problem at hand, we explain our reasoning behind why we chose the System Dynamics methodology, describe structure and the process of building of the model, show the results of our simulations, and explore possible policies to remedy the quality decay. Recommendations are made about what are some of the measures that ALEC can take to improve the change of quality over time. Finally, given that our model makes some limiting assumptions about the realities of the organizational dynamics at ALEC, we discuss how to elaborate the model for further research.

**Methodology**

The decision to use System Dynamics as a methodology for the project was not made by our team, but instead was proposed by our contact with A Large Engineering Company (ALEC). Our contact had some experience with the System Dynamics graduate program at WPI, and was interested in applying the methodology to the address quality issues at ALEC. To understand this decision, one must look into the previous attempts by ALEC to identify and control long term quality control issues. Many of ALEC's efforts were driven by a more statistics based approach. Six Sigma has a strong place in the organization, and has demonstrably proven results, but fails to address the “big picture”, and is incapable of processing qualitative feedback. System Dynamics complements Six Sigma well, and is able to address some of the areas that Six Sigma is not associated with.

The modeling process is capable of considering qualitative information as links in the model. This is both a strength and a weakness of the System Dynamics approach. By incorporating these links, the model is able to include important feedback from employees that is unable to be used in a statistics driven approach. These unquantifiable links, however, detach the model from the numerical results that it might produce. A model such as the one we created can only be used to suggest trends or compare policies, but will not be able to predict the numerical changes that might result from a given change. That is to say, our model might be able to produce the result “Policy P1 will have a stronger positive effect on Stock S1 than Policy P2,” but would be unable to create a result such as “Policy P3 will increase Stock S2 by 15% over 3 years.” The strength of a qualitative approach is rooted in the nature of modeling, and is the result of being able to explore the effect of different loops within the model. Complex behavior can be traced to relatively simple structures, and often times it becomes clear where the leverage points in a system are during the modeling process. System Dynamics promotes a more holistic understanding of cause and effect in the model, and allows the user to target areas which will
amplify a potentially positive result. This understanding of the nature of the system allows for
the creation of rules that exploit the system for an advantageous result.

The focus on qualitative links offered a challenge. We were working with an engineering
group at the organization, and most were very hesitant to accept a qualitative approach to the
problem. Many struggled to understand the role that data plays in a System Dynamics model, or
that links might be added to the model that were not directly drawn from the available data.
Engineers are used to working with very concrete problems, tackling a less quantifiable problem
was seen by the group as an interesting challenge, but led to some amount of hesitancy.

Another challenge in this project, and a problem with modeling in general, was the
selection of the boundaries of a model. Through our weekly meetings with ALEC it became clear
that there were several potential areas for study. These included the employee supply chain
(which we eventually chose), the work contracts/changing of specifications provided by the
government, and the union structure. In the final product, many of these are ideas are
incorporated, yet simple rules guide several of the variables. Our model boundary selection was
also guided by literature such as Sterman (2000)\(^1\) and Saeed (1992)\(^2\). In most cases, there is
room to expand the boundaries of the model to create a more robust product, and this idea will be
expanded upon in the write-up of areas for future study.

Model Building Process

Our modeling process followed the guidelines laid out by Akkermans in a paper on a
strategy called Participative Business Modeling, or PBM (1995)\(^3\). Quoting Akkermans:
“Modeling in PBM moves gradually from very informal, qualitative, and conceptual models to
more formal, quantitative simulation models.” This chapter details the model-building process,
starting with the definition of a problem, proceeding with a model conceptualization/brainstorm
phase, and finally refining the ideas conveyed by the team at ALEC into a concrete model.

Recognizing that quality decay in the production has been occurring, ALEC decided to
attempt utilizing a systemic approach to understand the dynamics behind the undesirable
behavior. Meeting on a weekly basis with ALEC, we first had to identify the slice of the problem
we were going to dwell on. ALEC discussed several areas of concern that may have stimulated
the quality decay, including the relationship with the Government (specifically demand in
changes in the product’s specifications), union contract that discourages mentoring new workers,
or the structure of the workforce supply chain. We ultimately decided to focus on how the
workforce supply chain, as it is presently established, relates to the average skill and productivity
of the system, and consequently, to quality behavior. While we will include some of the elements
that had been identified as possible causes of the quality loss in our model, such as the
customer’s change in specifications, we will make the workforce supply chain the central


Dynamics Review. 8(3).

Modelling.
emphasis of our project.

During our first two meetings, our main emphasis was on creating a reference mode of the behavior of the quality over the past several years. Through our discussions, we were ultimately able to illustrate their description of the behavior of quality in the following graph, our reference mode:

![Figure 1.1 -- Reference Mode](image)

**Figure 1.1 -- Reference Mode**

DESCRIBE REFERENCE MODE HERE IN A FEW SENTENCES

After the “golden age”, a period when quality was high and stable, ALEC entered a phase in which quality was decreasing. The intersection of the graph of quality and the “NOW” axis illustrates the state of the quality when ALEC contacted us. Their fear was that quality would continue decreasing (as depicted by the red graph). Their hope, however, was to understand the system, capture the reasons behind the past decrease, and take appropriate action to bring quality back to its stable and high state (as depicted by the green graph).

Our primary task during the first few weeks was to understand as much of the problem and of the organization as possible. In other words, we focused on capturing their mental model of the dynamics behind the workforce supply chain, and translate it into a system dynamics causal loop diagram.

Figure 1.2 illustrates our preliminary view of the system that we were modeling:
Figure 1.2 -- Dynamic Hypothesis

The causal loop diagram captures the basic dynamics of the workforce supply chain, and how they relate to quality. We begin by assuming that when New Workers are increased, the required training time will increase as a result, which in turn increases the fraction of time spent on training. Increasing the fraction of time spent on training increases the total expertise in the system, which increases quality, decreases total rework, decreases the fraction of time spent working, and decreases the fraction of time spent training, hence completing our negative feedback loop, but this takes a while. Expertise decreases in the short run when New Workers are added in the system, but increases when Senior Workers are increased. Note that there is a link between New Workers and Senior Workers because we assume that with time New Workers advance to Senior ranks (hence the dashed lines on the link, which represent a delay). Finally, when quality increases, total rework decreases, fraction of time spent working increases, and quality increases again, completing our positive feedback loop. Note that we have included an exogenous variable, “Complexity of work”, which, when increases, increases the training required, and decreases quality. Moreover, had we initially decreased the amount of new workers (or any other variable), the linked variables would have behaved in the opposite direction, that is, a decrease would have been an increase and vice versa.

The next step was to translate the causal loop diagram into a system dynamics model that is built for simulation. The following section discusses the structure of the model. For information regarding how we refined some of the details of our model, please refer to the meeting minutes. Figure 2.1 shows a high-level map of the model. The model is split into three sectors: the Work Sector, the Skill Sector, and the Employee Sector. Employee sector contains the employee supply chain and shows how employees are hired and progress through different ranks of the organization. The Skill Sector measures the quality of work in the organization by looking at the amount of employees at each skill level in the Employee Sector. The Work Sector
details work additions, work, rework, and schedule compression. All of these will be elaborated on in the following section.

![Sector Relationship Map](image)

**Figure 2.1 -- Sector Relationship Map**

**Structure of the Model**

Each sector of the model in figure 2.1 is described below:

**Workforce Supply Chain**

At A Large Engineering Company, the progression of a worker’s status is structured, and we assume all workers follow the same path. The employee supply chain was constructed on the basis of the interviews with the client and it aggregates several sub-categories of employees into the categories considered in the model and shown in Figure 2.2. It loosely resembles the Human resource Management Subsystem proposed by Abdel-Hamid/Madnick in their Software Project Dynamics book (1991)\(^4\). All new hired workers begin as “learners” (our model assumes that workers cannot enter the system as experienced workers or designers), at which point there are two possible routes to advance to the next position. They can either gain experience through on-the-job learning and still be considered learners, or they can enter a structured training program (usually reserved for the more qualified learners), and be considered apprentices. The rate of entry into the apprenticeship program depends on the amount of learners in the system, and only 5% (found under the parameter “A Rate” in the model) of the learners become apprentices. After graduating from the program, which on average takes a fixed amount of time (5 years in our model), Apprentices become Designers.

---

The other path that the learner can undertake to be promoted to the Designer position is through on-the-job learning. Unlike the apprenticeship program, this path is not structured and its duration may vary, as it depends on the fraction of time experienced workers spend training the learners. That said, we have developed a parameter “Eff. of Designers”, to account for the impact that the time experienced workers spend training the learners has on the learners’ promotion rate. “Eff. of Designers” depends on the number of learners in the system, the number of designers, and the fraction of time that designers spend training the learners.

Once learners become designers, their skills and productivities increase, affecting the average skill (and consequently quality) and total work done, respectively. With time and experience, designers advance to become either design techs or supervisors. Finally, the design techs and supervisors leave the system. While our model does not allow hiring experienced workers, all workers can leave the system according to some attrition rate. The attritions rates are 5% for the learners, 4% for the apprentices, 4% for the designers, and 8.5% for design techs and supervisors.

Further, we developed a variable “New Hires” that will determine the hiring rule, that is, the value of the inflow “Hires”. “New Hires” will vary according to the relationships that we will set it to depend on, and we will experiment with various methods of hiring new workers. For example, the work gap or the attritions rate could determine when to inject new workers into the system. Note that the initial values for the stocks of each type of employee have been given to us by ALEC.
Figure 2.2 -- Employee Sector

Work Cycle Sector

The Work Sector, shown in Figure 2.3, is divided into several smaller parts. The core of the model is composed of the Work Needed to be Done stock and the Completed Work stock. The structure for Work and Rework is based upon the model proposed by Abdel-Hamid/Madnick in Software Project Dynamics(1991)\(^5\). Work is added to Work Needed to be Done via the Work Additions flow, which represents the acquisition of contracts. A complexity variable accounts for the difficulty of the work, and changes in difficulty over time. Complexity modifies work additions proportionally; A complexity of 2 would double the amount of work need to complete a contract. Work flows from the Work Needed to be Done Stock to Completed Work at a particular rate (Total Work Done).

Total Work Done is based on many factors, yet remains a simple equation. It is merely a summation of the amount of each category of worker multiplied by the fraction of time they spend working multiplied by the productivity for each category of worker. Each worker contributes a certain amount of work to the system, and the aggregate value is Total Work Done. This value is then modified by the effect of the employee supervisor ratio.

\[
Total\ Work\ Done = (Apprentices \ast Apprentice\ Productivity \ast A_{Fr\ Time\ Working} \\
+ Designers \ast Designer\ Productivity \ast D_{Fr\ Time\ Working} \\
+ Design_Techs \ast Design_Techs\ Productivity \ast DT_{Fr\ Time\ Working} \\
+ Learners \ast Learners\ Productivity \ast L_{Fr\ Time\ Working}) \\
\ast \ Eff_{of\ Emp\ Sup\ Ratio}
\]

Based on Quality, work may or may not become rework. Rework is divided into two components, there is a base rework rate, representing the portion of rework that cannot be avoided. This rate does not change, and represents the amount of rework caused by change in specifications or other unforeseeable circumstances. The other component of rework varies with Quality. As Quality drops, the amount of work that needs to be redone increases proportionally to the drop in Quality.

\[
Rework\ Rate = \ Change\ in\ Specification + Work\ Rate \ast (1 - Quality)
\]

The last part of the Work Sector is the Schedule Compression Loop. Schedule Gap looks at the difference between Work Additions and the current Work Rate, and calculates how much extra time would be needed to complete the given work on schedule. As this gap grows, employees begin to rush their work. Quality will be lowered, but work rate will be increased as the amount of Schedule Compression grows.

\[
\left(\frac{Work\ Needed\ to\ be\ Done}{Work\ Rate}\right) - Desired\ Project\ Time
\]
The skill sector, shown in Figure 2.4, looks at a weighted average of the skill of all employees to determine the level of Quality in the organization at a given time. Each type of employee has a different level of skill associated with his/her group. These parameters were suggested by the team at ALEC based on their experiences working with members of each group. Supervisors are not included in the average, but have a different avenue of affecting Quality. The ratio of employees to supervisors is calculated, and as the number of employees per supervisor grows, there is a negative impact on Quality. The effect of the Employee supervisor ratio is normalized based on a normal employee to supervisor ratio, the function for which is shown in Figure 2.5. The graph is S-Shaped with an inflection point at the normal value.
Increasing beyond the normal value will have diminishing returns, while any increase from below that point will have an increasing effect. While discussing the graph with the employees of ALEC, we agreed to assume that the effect of supervision is 25% in either direction. Adequately supervised employees are up to 25% more productive, while inadequately supervised employees have a decline in productivity of up to 25%.

\[
(A_{\text{Skill}} * \text{Apprentices} * A_{\text{Fr\_Time\_Working}} \\
+ \text{Designers} * D_{\text{Skill}} * D_{\text{Fr\_Time\_Working}} \\
+ \text{Design\_Tec\_s} * DT_{\text{Skill}} * D_{\text{Fr\_Time\_Working}} \\
+ L_{\text{Skill}} * \text{Learners} * L_{\text{Fr\_Time\_Working}})
\]

\[
\text{Eff\_of\_Emp\_Sup\_Ratio} \quad \text{Supervisors}
\]

\[
(\text{Apprentices} * A_{\text{Fr\_Time\_Working}} \\
+ \text{Designers} * D_{\text{Fr\_Time\_Working}} \\
+ \text{Design\_Tec\_s} * DT_{\text{Fr\_Time\_Working}} \\
+ \text{Learners} * L_{\text{Fr\_Time\_Working}})
\]
**Base Run**

In our base run, our hiring rule is a function of the discrepancy between work needed to be done and average work done. Average work done (productivity) is a weighted average that represents how sum of work that can be done by the workers, divided by the total number of employees. Figures 3.1 and 3.2 show simulations of the base run.
Figure 3.1 -- Base Run

Figure 3.2 -- Base Run Employees

Figure 3.1 illustrates the changes in work quality over time, and how it varies with the hiring process. In the beginning (from \( t = 38 \) to \( t = 52 \)), our system is experiencing the “golden age” identified by ALEC, where quality was consistently high. During the beginning of the golden age, quality is rising because our apprentices and learners are becoming designers and design techs. Note that while the number of designers is decreasing, the proportion of designers to apprentices/learners is high. At \( t = 50 \), because desired workforce is higher than actual workforce, our hiring policy is implemented, and thus, an influx of learners is added to the system. Consequently, quality decreases. With time, these learners/apprentices become designers, and quality rises again, thus completing the cycle.
Extreme Value Testing

No Hires

If no hires are allowed into the system, Work builds up over time and quality eventually decreases. The original growth of quality can be attributed to the fact that most employees will become designers or design techs before they leave the company.

![Figure 4.1 -- Behavior of Employees in a test when hiring is stopped](image1)

![Figure 4.2 -- Behavior of Quality in a test when hiring is stopped](image2)

Without any work in the system, there is no need to hire employees. Employees eventually decline to zero as those that started initially in the model leave over time.
Thus, the results reassure us that the model is robust; the model behaves as it would be expected to under these extreme conditions. This is not proof of the validity of the model, but it suggests that the model reasonably represents the system.

**Policy Experimentation**

**Hire Adjustment Time**
In our first policy experiment, we change the hire adjustment time, which represents the number of years it takes to respond to work force discrepancy. This simulation is shown in figure 5.1.

**Figure 5.1 -- Hire Adjustment Time**

Lines 1 through 6 in Figure 5.1 represent hire adjustment times of 1 through 6 years respectively. Clearly, as the time needed increases, the period of the oscillations and the magnitude of the oscillations in quality increases. A low adjustment time is desirable. Having a lower adjustment time is better since not only do we note a decreasing periodicity, but also, the magnitude of the troughs is lower.

**Hiring people to replace attritions**

In this experiment, we will modify our model such that every time a worker leaves, a new worker (i.e. a learner) will be added to compensate for the attrition. Recall that in our base run, hiring is based on the work gap only. While Figure 1.1 illustrates the original behavior of the system, Figure 5.2 illustrates what happens if we only hire to compensate for the workers who leave and do not take into account the work gap.
Figure 5.2 -- Hiring for Attrition (Hires/Quality)

Figure 5.3 -- Hiring for Attrition (Employees)
Figure 5.4 -- Hiring for Attrition (Quality/Schedule Compression)

Clearly, eliminating hiring gap for gap removes the cycles in the model, which is desirable. However, as is discernable from figure 5.4, schedule compression is the driving force behind the work. As work rate is reduced, schedule compression is used to compensate for the gap, and quality suffers. Since our total number of workers is constant, and a percentage of our work done goes into rework, the total work done will increase. Therefore, in order to compensate for the increase in total work (i.e. work + rework), we must compress schedules.

**Hiring for attrition and for work gap**

Now, in order to remedy the increase in schedule compression, we will set up the model such that it includes the hiring rule for both attrition and work gap. As we expected, some oscillation occurs, but the system eventually finds equilibrium. In both cases, hiring for attrition and hiring for attrition and work gap, equilibrium is eventually found. The latter case has a significantly higher equilibrium quality value. Figure 5.5 demonstrates this new rule and shows that a higher equilibrium value for quality is attained.
Another policy to test is hiring at a fixed rate, regardless of whether or not workload is increasing or decreasing. The following graph depicts how quality changes as result over time.

As shown in Figure 5.6, quality increases for a short amount of time, but a relatively low equilibrium quality is attained in the long run. This behavior is expected since there is a large amount of new workers at the start of the run, and hence, when this group advances in the workforce supply chain, quality increases. With time, this initial group that is relatively large (because this group of workers that we are tracking consists of our the workers initially in the
system as well as a added group of workers at time zero due to constant hiring) will attrite, and consequently the proportion of workers in each stock will become stable. Moreover, as we have previously established the total work done increases due to rework. Consequently, quality decreases, because schedule compression kicks in. Note that this behavior is similar to that of hiring for attrition. In fact, hiring for attrition in our model is a type of constant hiring, for, the same proportion of workers is leaving the system with time, and thus the hiring rule is constant.

Therefore, for a certain value of “constant hiring rate”, the constant hiring rule is equivalent to hiring for attrition. However, what would happen to our system’s behavior if we changed the magnitude of the constant rate? Let us test our system for various values of constant hiring rates.

To do so, we develop the following comparative graph, shown in Figure 5.7:

![Figure 5.7 -- Constant Hiring Rate (1, 10, 15, 20, 25 employees per year)](image)

Lines 1-5 depict how quality changes when the “constant hiring rate” is 1, 10, 15, 20, and 25 respectively. Note that the general behavioral mode of graphs 1, 2 and 3 changes significantly from that of graphs 4 and 5. In fact, in graphs 1, 2 and 3, quality first increases, reaches a pinnacle, and then decreases to reach a low equilibrium point. On the other hand, graphs 4 and 5 first decrease, and then reach a high equilibrium point. Note that both 4 and 5 eventually merge and experience the same behavior.

This behavior is understandable. In graphs 1, 2 and 3, the hiring rule does not provide enough workers to complete the work without a compression in schedule, and consequently quality decays (as explained previously). In graphs 4 and 5, due to the initially high number of learners, quality decreases. Clearly, quality decays more in 5 than in 4 because our system carries a greater number of learners. This decrease is followed by a rise in quality because the initial
group of learners has advanced and has become more skilled. The system ultimately reaches equilibrium when the initial group leaves the system, and the proportion of skilled to unskilled workers does not change.

Let us now probe into why graphs 4 and 5 eventually reach the same equilibrium point. The answer, quite simply, is that because the amount of work that has to be done is constant, only a specific number of workers are required. Our model assumes that having more workers than is necessary does not increase quality, for there is only a specific number of workers needed to complete the work. Finding this value, which would be the lowest number of workers such that the system reaches the high equilibrium point, will optimize our system, because quality will decrease the least at first and still achieve the same equilibrium. Furthermore, if our model were to be expanded to include the cost and wage of workers, the value of the constant hiring rate will be of great importance, and we will still want to choose the lowest number of workers to constantly hire such that the reference mode is that of graphs 4 and 5. In our model, we find that this value is 20 workers.

Given that the work addition rate is constant and that a fraction of work (that depends on quality) will go into rework, the total work done will increase over time. For example, if 20 workers are able to maintain a quality of 0.90, then our rework will be 10 percent of our completed work (changes in specifications exist will cause rework whether or not our workers are skilled). If we begin with 100 as a workload, and assume that there are no changes in specifications, then rework will increase our total work done to 110. It is important to note that when using a constant hiring rule, there must be enough workers to cover both the work, and the rework. If there are not enough workers, rework will continue to grow, and quality will begin to decay.

**Ramping Work Rate**

In the previous experiments, it has been assumed that the work inflow rate remains constant. That is to say, the company receives new work contracts at a consistent rate. While those experiments are valuable in terms of identifying how the system reacts to different hiring rules, they are hardly realistic. In this run, it is assumed that the inflow of work ramps up periodically. Every 20 years, the company receives an additional unit of work, as can be seen by the Work Ramp Function shown in figure 6.1.
While hiring at a rate of 20 employees per year may have been the optimal solution when work was added at a constant rate, there is a severe decay in quality if this policy is applied to a ramping rate. This serves to logic, as a constant rate is unable to address the increasing amount of work, and has no structure to compensate or add employees to the system to handle the additional work. Shown in Figure 5.9 is a trial run with a constant hiring rate and ramping work rate.

Figure 5.8 -- Ramping Hiring Graphical Function
Our base run already has shown that hiring for gap only is a weak strategy, yet applying the gap strategy to the ramping work rate shows that it performs drastically worse when applied to a situation with a dynamic work rate. Whereas the base run with gap only hiring demonstrated sinusoidal behavior with a trough quality of ~.5, a run with ramping work additions expressed a similar shape, but with a trough quality of ~.32. This is shown in figure 5.10.
Figure 5.10 -- Ramping Hiring

With Hire for Attrition and Ramping Work Rate

As with a steady work rate, hiring for attrition serves to eliminate the sinusoidal behavior. Quality reaches a plateau, and remains at that level after the system has reached an equilibrium where hires=attrition and the amount of employees at each experience level remains constant. This run is shown in Figure 5.11.

Figure 5.11 -- Ramping Hiring

Random Work Rate (Normal Distribution with mean 1 and a standard deviation of 0.5)

We now assume that the work is added randomly, following a normal distribution with mean 1 and standard deviation of 0.5. That said, the equation for “Work Addition Rate” is NORMAL(1, .5). Then, we implement various hiring rules, and see how quality changes as a result.

Constant Hiring Rate

We begin by setting up a comparative graph of the quality with various hiring values. We get the following graph:
Figure 5.12 -- Random Work Rate

Graphs 1 through 5 depict hiring values of 1, 5, 10, 15, 20, and 25 respectively. Eventually, with 20 workers, quality stabilizes, because we are no longer relying on schedule compression to get the job done. We note that this behavior is very similar to the system where work rate is constant. Since we are most interested in the general behavior, rather than specific outputs, we can confidently rely on the same hiring policy as that of the constant work rate when confronted with random fluctuations in work rate.

The following experiment illustrates an important point regarding the dynamics that govern a system. As noted previously, the behavior of quality with randomness is generally the same as with constant work addition rates. Noise accompanies randomness (because of the variance), and one would expect our system to be disturbed. However, quality approaches an equilibrium value on the long run, suggesting that the noise has no significant effect on our system on the long run. Clearly, the effects of the endogenous dynamics of the system are greater than that of the noise. Thus, we witness how significant the structure of a system is in determining the nature of the governing dynamics.

Recommendations for future research

Given that our model is currently making several limiting assumptions, ameliorating the model by including the details that we did not take into account due to time constraints is a good starting point. Our most fundamental assumption, and that is due to the systemic approach of system dynamics, is that the model illustrates the dynamics at hand from an aggregate point of view. That said, our model takes into account averages and cannot tell us anything about specific cases within the variables and stocks. While, as mentioned, this is a feature of system dynamics,
researchers in the future can include the various subcategories within each stock of workers. Figure 6.1 illustrates the types of workers that fall under each stock.

![Employee Aggregate Chart](image)

**Figure 6.1 – Employee Aggregate Chart**

Furthermore, our model does not take into account that design techs and apprentices can either follow the design or the engineering track. Including the natural progression of workers within the subcategories in the workforce supply chain is an area of interest that may reveal further insight, and yield new policy space.

A second assumption our model makes is that the time an experienced worker spends with a learner does not change and remains the same, because of aggregation. Consequently, the sole way to increase the fraction of time spent training new workers is by increasing the number of designers. We have assessed that there are two ways to address this assumption: either by having supervisors mandate the time as a policy, and hence let it be an exogenous factor, or by having internal dynamics determine the level of commitment, and let it be an endogenous factor, such as supervisor ratio. Including this assumption in our model may also allow for new policy space. For instance, giving incentive for the designers to train the learners will accelerate the new workers’ transition from the learner stage to the experienced level, and potentially have a desirable impact on quality.

A third area of expansion is to model the workers’ ability to handle schedule compression. Our model currently considers that there is a general limit on how much schedule
compression can increase, however, further research should address in more depth how much schedule compression can be utilized as the driving force behind the work done.

Another assumption that our model makes is that a decrease in quality does not lead to a change in the rate of work additions. However, one may argue that if quality decreases the company would lose business because the customer would rely on another service provider. This depends largely on the type of market that the company is in, but given that competitors exist, to be more complete, our model must include the implications of a decrease in quality with respect to the customer’s willingness to contract the company.

Furthermore, future work can develop a model that allows hiring of experienced personnel, and examine the effect of the policy on quality. It will be crucial to also include how feasible such a policy is when applied to the organization. In other words, our model would doubtless choose to hire experienced workers over learners if hiring the two types of workers lead to the same costs for the organization; however, this is not the case, and not only are experienced workers harder to find, but their inflow at a certain point in time is most likely less than that of new workers.

During one of the later meetings with ALEC, the rate of flow for the apprentice program was discussed. Currently, 5% of people in the Learner stock become apprentices. A more accurate way to describe the apprenticeship program, however, would be to have 30% of all new hires go into the apprentice program. Although this change in formulation has very little impact on the base run of the model, as is shown in Figures 6.2 through 6.4, More study would be needed to figure out the exact implications of changing this rule, and the sensitivity of the parameters involved.

Figure 6.2 – Quality graph With 30% apprentice rate
Figure 6.3: Quality graph with current apprentice program rate

Figure 6.4 – Employee graph with 30% apprentice rate
Finally, modeling the implications of our hiring rule on the company is of significant interest and value. We have attempted various hiring rules, and have examined their effect on quality. However, any company, ALEC included, would be keenly interested in knowing what the costs of the hiring strategy are, and see whether it is economically feasible to implement the policy. Modeling the financial implications will be a stepping stone to going from the “modeling/simulation world” to the “real-world”, as we study how to effectively implement the policies in the company. Needless to say there are several other implications of the hiring rule, and future researches must closely coordinate with ALEC to understand the system more fully, and explore additional consequences of the hiring rule.

**Conclusion**

The base run of our model did not exactly depict our initial reference mode; however, we realized that what ALEC perceived as being the behavior of quality was simply part of the larger cycle seen in the base run’s simulations. The “golden age” in quality (the period when quality was consistently high) was merely the crest of an oscillating system that is due to the structure of the organization. As work is added to the system, employees needed to be brought on to handle the additional amount of work, which caused decay in quality due to an increase level of inexperience in the organization. However, with time, the learners become more advanced, and progress in the ranks of the workforce supply chain, to become designers, design techs, and supervisors, hence increasing the level of experience in the system, and consequently quality. After the experienced people begin to leave, new workers are hired, and quality falls as a result. And the cycle repeats, leading to the oscillatory behavior seen in the base run. We have noted that the reference mode described by ALEC depicts a part of the oscillations that our simulations
have shown. System Dynamics provides us with the advantage of perceiving the long-term behavior of the system, whereas the mental models of the ALEC employees solely address the short-term picture.

Shown in Figure 7.1 is a small section of the base run from time 50 to 60. This section depicts the mental model of the employees, which we were able to reproduce. This shows that the “golden age” of quality was from time 50 to time 57, at which point quality began to decay. Time 60 would approximately correspond to the year 2010.

![Figure 7.1](image)

**Figure 7.1 -- Conclusion**

Our experiments were targeted at dampening the oscillations in the system. Oscillations in the system occurred because the hiring inflow occurred in bursts, that is, new workers would be injected into the system, in large amounts, after long periods of no hiring. To address this issue, the bursts must be smoothened out, and we achieved this by hiring for attrition.

In a system where work is added at a constant rate, or follows some kind of normal distribution, a linear hiring rate is the preferred rule since it removes the oscillations. Clearly, however, the company must assess the strategy to see whether it is financially feasible.

In a system where work is added at an increasing rate (ramping work rate), hiring for attrition or at a constant rate are no longer beneficial rules, as is previously seen in the simulations. To address the new type of work inflow, we have found that combining both rules, hiring for attrition and for work gap, is the most beneficial strategy. Implementing this rule in the company requires constant and thorough investigation of the necessary amount of workers for a given amount of work inflow.
Although this model was designed specifically for ALEC, it can be noted that the findings of this model could be applicable to a vast array of organizations whose basic structure is similar. This model applies to any organization that must deal with long term contracts arriving at an inconsistent rate, with a long training time for employees.

We recommend that, if ALEC chooses to further pursue the project, the company develops the model to address the limitations of the assumptions listed previously. They can be strong starting points for future research.

Appendix

The Interface

The goal of the interface was to deliver a product that allowed ALEC to experiment with different scenarios while assuming that the user would have little, if any, knowledge of using System Dynamics software. To accomplish this goal, the Interface creation tools of iThink were used. From the Interface, many parameters can be easily accessed and tweaked. The table under the graph gives access to most of the important parameters used by the model. These include the initial employees (shown), attrition rates, productivity of each type of employee, skill of each type of employee, promotion rates, and the amount of time spent working for each category of employee. The goal was to gather all of these important numbers to one location, so that a user would not be required to dig through the model.

Other variables such as Work Addition Rate and Complexity could not be modeled by simply changing a parameter. These were added as graphical functions, and can be seen in the lower left hand corner. Instead of writing an equation to model these, a user could simple “draw” an equation in the box. In Figure 8.1, Work Addition Rate is set to ramp over time.

The last part of the interface is the Policy Switch Section. This section allows the user to turn hiring rules on and off, and allows for the combination of different hiring rules. These switches can be used to easily recreate the experiments demonstrated in this paper, and could allow for some amount of further exploration.
Figure 8.1 -- The Interface

Equations

Apprentices(t) = Apprentices(t - dt) + (A_Program - A_Promotions - A_Attrition) * dt

INIT Apprentices = Initial_Apprentices

INFLOWS:

A_Program = A_Rate*Learners

OUTFLOWS:

A_Promotions = Apprentices/5

A_Attrition = A_Attr_Rate*Apprentices
\[ \text{Completed}_\text{Work}(t) = \text{Completed}_\text{Work}(t - dt) + (\text{Work}_\text{Rate} - \text{Rework}_\text{Rate}) \times dt \]

INIT \text{Completed}_\text{Work} = 0

INFLOWS:

\[ \text{Work}_\text{Rate} = \text{Total}_\text{Work}_\text{Done} \times (\text{Schedule}_\text{Compression}) \]

OUTFLOWS:

\[ \text{Rework}_\text{Rate} = \text{Change}_\text{in}_\text{Specification} + \text{Work}_\text{Rate} \times (1 - \text{Quality}) \]

\[ \text{Designers}(t) = \text{Designers}(t - dt) + (\text{L}_\text{promotions} + \text{A}_\text{Promotions} - \text{D}_\text{Promotions} - \text{D}_\text{Attrition} - \text{D}_\text{to}_\text{S}_\text{Promotions}) \times dt \]

INIT \text{Designers} = \text{Initial}_\text{Designers}

INFLOWS:

\[ \text{L}_\text{promotions} = (\text{Learners}/8.5) \times \text{Eff}_\text{of}_\text{Designers} \]

\[ \text{A}_\text{Promotions} = \text{Apprentices}/5 \]

OUTFLOWS:

\[ \text{D}_\text{Promotions} = \text{New}_\text{DT}_\text{Hires} \]

\[ \text{D}_\text{Attrition} = \text{D}_\text{Attr}_\text{Rate} \times \text{Designers} \]

\[ \text{D}_\text{to}_\text{S}_\text{Promotions} = \text{Supervisor}_\text{Fr} \times \text{Designers} \]

\[ \text{Design}_\text{Techs}(t) = \text{Design}_\text{Techs}(t - dt) + (\text{D}_\text{Promotions} - \text{DT}_\text{Attrition}) \times dt \]

INIT \text{Design}_\text{Techs} = \text{Initial}_\text{Design}_\text{Techs}

INFLOWS:

\[ \text{D}_\text{Promotions} = \text{New}_\text{DT}_\text{Hires} \]

OUTFLOWS:

\[ \text{DT}_\text{Attrition} = \text{DT}_\text{Attr}_\text{Rate} \times \text{Design}_\text{Techs} \]

\[ \text{Learners}(t) = \text{Learners}(t - dt) + (\text{Hires} - \text{L}_\text{promotions} - \text{L}_\text{Attrition} - \text{A}_\text{Program}) \times dt \]

INIT \text{Learners} = \text{Initial}_\text{Learners}

INFLOWS:
Hires = New_Hires

OUTFLOWS:

L_promotions = (Learners/8.5)*Eff_of_Designers

L_Attrition = L_Attr_Rate*Learners

A_Program = A_Rate*Learners

Schedule_Compression(t) = Schedule_Compression(t - dt) + (Chg_in_SC) * dt

INIT Schedule_Compression = 1

INFLOWS:

Chg_in_SC = (Eff_of_Schedule_Gap-Schedule_Compression)/Time_to_change_SC

Supervisors(t) = Supervisors(t - dt) + (D_to_S_Promotions - S_Attrition) * dt

INIT Supervisors = Initial_Supervisors

INFLOWS:

D_to_S_Promotions = Supervisor_Fr*Designers

OUTFLOWS:

S_Attrition = S_Attr_Rate*Supervisors

Work_Needed_to_be_Done(t) = Work_Needed_to_be_Done(t - dt) + (Rework_Rate + Work_Additions - Work_Rate) * dt

INIT Work_Needed_to_be_Done = Normal_Work

INFLOWS:

Rework_Rate = Change_in_Specification+Work_Rate*(1-Quality)

Work_Additions = Work_Addition_Rate*Normal_Work*Complexity

OUTFLOWS:

Work_Rate = Total_Work.Done*(Schedule_Compression)

Apprentice_Productivity = 0.6

Avg_Skill = (A_Skill*Apprentices*A_Fr_Time__Working+Designers*D_Skill*D_Fr_Time__Working+Desi
\[ \text{Avg \_ Work \_ Done} = \frac{\text{Total \_ Work \_ Done}}{\text{Total \_ Employees}} \times 0 + 1 \]

\[ A \_ Attr \_ Rate = 0.04 \]

\[ A \_ Fr \_ Time \_ Working = 0.60 \]

\[ A \_ Rate = 0.05 \]

\[ A \_ Skill = 0.4 \]

\[ \text{Change \_ in \_ Specification} = 0.1 \]

\[ \text{Constant \_ Hires} = 0 \]

\[ \text{Constant \_ Hires \_ Rate} = 1 \]

\[ \text{Designer \_ Productivity} = 1 \]

\[ \text{Design \_ Techs \_ Productivity} = 1.2 \]

\[ \text{Desired \_ Project \_ Time} = 1 \]

\[ \text{Desired \_ Workforce} = \frac{\text{Work \_ Needed \_ to \_ be \_ Done}}{\text{Avg \_ Work \_ Done}} \]

\[ DT \_ Attr \_ Rate = 0.085 \]

\[ DT \_ Fr \_ Time \_ Working = 0.90 \]

\[ DT \_ Skill = 1.2 \]

\[ D \_ Attr \_ Rate = 0.04 \]

\[ D \_ Fr \_ Time \_ Spent \_ Training = 0.4 \]

\[ D \_ Fr \_ Time \_ Working = 0.60 \]

\[ D \_ Skill = 1 \]

\[ \text{Eff \_ of \_ Designers} = (0 \times \text{Designers} \times \text{Learners} + 1) \times D \_ Fr \_ Time \_ Spent \_ Training \]

\[ \text{Emp \_ Sup \_ Ratio} = \frac{\text{Total \_ Employees}}{\text{Supervisors}} \]

\[ \text{Hire \_ Adjustment \_ Time} = 2 \]

\[ \text{Hire \_ for \_ Attrition} = 0 \]
Hire_for_Gap = 1
Initial_Apprentices = 29
Initial_Designers = 177
Initial_Design_Techs = 14
Initial_Learners = 66
Initial_Supervisors = 20
Learners_Productivity = 0.4
L_Attr_Rate = .05
L_Fr_Time__Working = 0.75
L_Skill = .3
New_DT_Hires = .05
New_Hires = MAX(((Desired_Workforce-Total_Employees)/Hire_Adjustment_Time, 0)*Hire_for_Gap + (A_Attrition+DT_Attrition+D_Attrition+L_Attrition+S_Attrition)*Hire_for_Attrition + Constant_Hires_Rate*Constant_Hires
Normal_Emp_Sup_Ratio = 15
Normal_Work = 154
Quality = Avg_Skill*(2/(1+Schedule_Compression))
Schedule_Gap = (Work_Needed_to_be_Done/Work_Rate) - Desired_Project_Time
Supervisor_Fr = .025
S_Attr_Rate = 0.085
Time_to_change_SC = 6
Total_Employees = Apprentices+Designers+Design_Techs+Learners
Total_Work_Done =
+Learners*Learners_Productivity*L_Fr_Time__Working)*Eff_of_Emp_Sup_Ratio

Complexity = GRAPH(TIME)
(1.00, 1.00), (10.9, 1.00), (20.8, 1.00), (30.7, 1.00), (40.6, 1.00), (50.5, 1.00), (60.4, 1.00), (70.3, 1.00), (80.2, 1.00), (90.1, 1.00), (100, 1.00)

Eff_of_Emp_Sup_Ratio = GRAPH(Normal_Emp_Sup_Ratio/Emp_Sup_Ratio)
(0.00, 0.76), (0.2, 0.765), (0.4, 0.782), (0.6, 0.823), (0.8, 0.905), (1.00, 1.00), (1.20, 1.06), (1.40, 1.11), (1.60, 1.15), (1.80, 1.19), (2.00, 1.22), (2.20, 1.24), (2.40, 1.25), (2.60, 1.25), (2.80, 1.25), (3.00, 1.25)

Eff_of_Schedule_Gap = GRAPH(Schedule_Gap)
(0.00, 1.00), (1.00, 1.11), (2.00, 1.19), (3.00, 1.29), (4.00, 1.41), (5.00, 1.50), (6.00, 1.60), (7.00, 1.70), (8.00, 1.79), (9.00, 1.90), (10.0, 1.99)

Work_Addition_Rate = GRAPH(TIME)
(1.00, 1.00), (10.9, 1.00), (20.8, 1.00), (30.7, 1.00), (40.6, 1.00), (50.5, 1.00), (60.4, 1.00), (70.3, 1.00), (80.2, 1.00), (90.1, 1.00), (100, 1.00)
Works Cited


