System Dynamics Group
Sloan School of Management
Massachusetts Institute of Technology

System Dynamics for Business Policy, 15.874
Professors Brad Morrison and John Sterman

Assignment 8

The Dynamics of Process Improvement[*]

Assigned 18 November; Due 2 December 2003
Please work on this assignment in a group of three people.

As competition has intensified, businesses have sought to improve not only their products but also their processes for producing goods or services. In the search for higher quality and productivity, firms have devised and deployed literally dozens of different improvement programs, including SQC, TQM, JIT, TPM, TBC, MAN, BPR, 6σ and many others.[1] Some firms have achieved tremendous success with these process improvement techniques. Yet despite huge investments of time, money, and training, many firms—perhaps most—have found it difficult to realize sustained improvement from their efforts. Why do so many quality improvement programs fail? How can firms design sustainable improvement programs? What are the dynamics of process improvement?

The model you will develop in this assignment will provide insight into these questions. The data and descriptions upon which you will base the model are real, though disguised. The assignment also develops your modeling and policy analysis skills: You will formulate a model of process improvement from a written description of a real case. You will then use your model to design policies to improve the implementation of improvement programs.

[*] Prepared by Nelson Repenning, April 1997. Revisions by John Sterman, Anjali Sastry, and Brad Morrison (latest in November 2003). For more information, see Repenning, N. and J. Sterman (2000) "Getting Quality the Old Fashioned Way: Self-Confirming Attributions in the Dynamics of Process Improvement" *The Quality Movement and Organizational Theory,* R. Scott and R. Cole, (eds.), Newbury Park, CA, Sage: 201-235; and an unpublished model originally developed by Nelson Repenning, Andrew Jones, Alex Voight, and Adolfo Canovi.

[1] That is, Statistical Quality Control, Total Quality Management, Just-In-Time manufacturing, Total Productive Maintenance, Time-Based Competition, Materials As Needed, Business Process Re-engineering, and Six Sigma.

❑ denotes a question for which you must hand in an answer, a model, or a plot.
• denotes a fact that you will want to include in the model under discussion.
* denotes a tip to help you build the model or answer the question.

Case Background: Based on your recent success with the Widgets corporation (more than $16.95 in documented savings), you recently received a referral for a consulting job at Big Electronics Corp. (Big El, for short), a leading manufacturer of electronic components. In your first meeting with the Vice President for Process Improvement, she explains the problem the firm is facing:

> Over the past few years, we've had tremendous success in improving our manufacturing operations. Almost every process we have tried to improve in manufacturing has shown significant gains. Defects have plummeted, productivity has grown, and the cycle-time for our assembly and manufacturing operations has dropped by a factor of ten.
>
> Since we've been so successful in manufacturing, it seems logical to turn our attention to other areas. Lately we've really focused on improving product engineering, but we've hit a brick wall. We've tried every tool and consultant under the sun, and frankly they haven't done a darn thing. Every effort we have made to reduce cycle time, increase yield, or do anything else in product engineering has failed miserably.
>
> Our CEO is really getting frustrated. His background is in manufacturing, and he was really the person who got all the improvements started in that area. He is where he is today because of those improvements, and he just can't understand how we can be such a success in manufacturing and such a failure in product engineering.
>
> To be perfectly candid, you are really our last hope. If we don't figure this out pretty quickly, our CEO is going to fire a bunch of us and see if somebody else can do a better job. I hope you can help us because we need it. My job may depend on it.

The VP then gives you a quick run-down of the people you'll be meeting during the day and asks you to be sure to tell her if you want to meet anybody else.

Meeting with Crawford

Your first meeting is with the CEO of Big El, Ed Crawford. You are a little nervous getting started (after all, this only your second system dynamics consulting job), so you start with a question you think he will like: "What was the secret of your success in improving manufacturing?" This turns out to be a good move; he leans back in his chair and says:

> It's all about leadership. There is nothing mystical or magical; I follow a few pretty simple rules. Rule #1 is you don't improve a process by adding extra people. Whenever supervisors ask for more people so they can do process improvement, I always say no. Improvements just don't come that way.
>
> Rule #2 is that any improvement effort needs a stretch objective directly related to the bottom line. So many of these things have really fuzzy objectives and never produce anything useful. I always focus on throughput. You can talk to me about process all you want, but if you want me to support an improvement effort you have to commit to making a big improvement in throughput before you are done. I usually ask for something on the order of 50% to 100%.
>
> Rule #3 is, whatever your objective is, it has to happen in two years or less. Before I got here, there were five-year plans to improve this and ten-year plans to improve that, and nothing was really happening. My feeling is that if you can't improve a process in two years or less, you just aren't trying. You have to keep people focused. Telling them that they have to improve throughput by 50% in less than two years tends to have that effect.
>
> Rule #4 is always give people money for training, tools, or consultants if they think it will help them improve. I'll even cut their throughput objective for a year or two to give them time to get their act together. Whenever people ask for resources or slack, however,

> I always say, “Fine, but in two years I am doubling the required throughput of your area.” And I do it. Two years to the day after they start their improvement effort, the honeymoon is over: I require twice as much throughput with no extra heads or capital resources.
>
> You know what, when I ran manufacturing, nobody ever missed. Most of the time, people got far more than 50% increased throughput and were begging me for more work. You’ve got to be tough on stuff like this—otherwise people will spend all sorts of time and money on silly stuff, and it never gets to the bottom line.

Feeling like things are going pretty well, you ask a tougher question: “If it was so easy in manufacturing, why hasn’t it worked in product engineering?”

Crawford’s good mood quickly passes and he is now visibly annoyed. “Those guys just don’t get it,” he says.

> I’ve been hammering on them for years, and they just don’t get it. I’ll give you an example. Right now in our product engineering area only about 20% of the designs are done correctly the first time. The rest have to be reworked, and then only 20% of those get done right, and so on. A few years ago, we spent all this money on computer equipment to increase the yield of our design engineering process. At the beginning people claimed we might be able to increase our yield by as much as a factor of four.
>
> Well, you know what happened? The engineers never took the time to learn how to use the tools. They are just sitting on desks gathering dust. Our engineers have this attitude that they are artists and shouldn’t have to follow a specific process or have any constraints on their creativity. There is just no discipline.

As you’re leaving, Crawford says something else:

> When you talk to everybody else today, they are going to tell you, “Oh, product development is different,” “It’s much harder,” etc. That’s BS. They have forgotten that we used to have all the same problems in manufacturing—long cycle-times, 80% error rates, etc. They seem to think we started from a different place, but it’s not true. The difference is that we were disciplined enough to fix the problems.

Product Development Engineers

Your next stop is with a group of product development engineers and their supervisor. You’ve heard they have become pretty cynical about process improvement, so you are a bit worried that they will not take your meeting seriously. Instead of jumping directly into talking about improvement, you ask them how they regulate their workload. The supervisor immediately answers:

> Well, that’s simple. We have a goal here of always turning our work around in two months or less. So every month or so we look at all the work we have currently outstanding and then determine how much we need to do to hit our cycle-time goal.

What happens, you ask, if there is more work than capacity? The supervisor answers:

> That’s not really a problem. We have a pretty good capacity management system, so things don’t get too far out of whack. And if they do, we get some extra resources or ask people to work some overtime. But, like I said, that almost never happens.

You then ask, “Do people ever need to cut corners or do things faster to get work done?”

The supervisor replies:

> I’m sure that happens once in a while but not very often. Our VP for product engineering has a standing rule that he will fire anybody who doesn’t follow the process. Besides, when our designs are finished, they are always of really high quality.

At this point the supervisor excuses himself and leaves the room.

You noticed that the engineers in the room became increasingly uncomfortable as the supervisor spoke, and none of them offered any additional information. Once the supervisor leaves, you ask them if they have any thoughts. They pretty much repeat what the supervisor said until you ask what they do when they have too much work to do. At this point one of the more senior engineers says, “I can answer that, but you have to turn your tape recorder off.” (You had been recording the interview for future reference, but you oblige and turn it off.)

> It’s really pretty simple. There is a basic rule that we all use. We look at how much work there is to do relative to what we could accomplish if we did what we are supposed to—follow the process, use the process improvement tools, etc. Then we discount from there. Actually in my area we sometimes calculate something we call the “impossibility index.” We take the number of hours we are allocated for the month and divide by the number of hours it would take to do all our work if we followed the process to the letter. Last month was pretty bad for my area. My impossibility index got down to almost 50%. Of course, it takes time to collect and report the data we need to figure out the impossibility index, and it takes folks a little time to respond.

Another engineer jumps in:

> Yeah, we do the same thing in my area. My impossibility index got so bad at one point that I was doing the absolute minimum required to get my work done. I think I was able to do most stuff in about one quarter of the normal time. Of course, I basically ignored every step in the process and did everything by the seat of my pants.

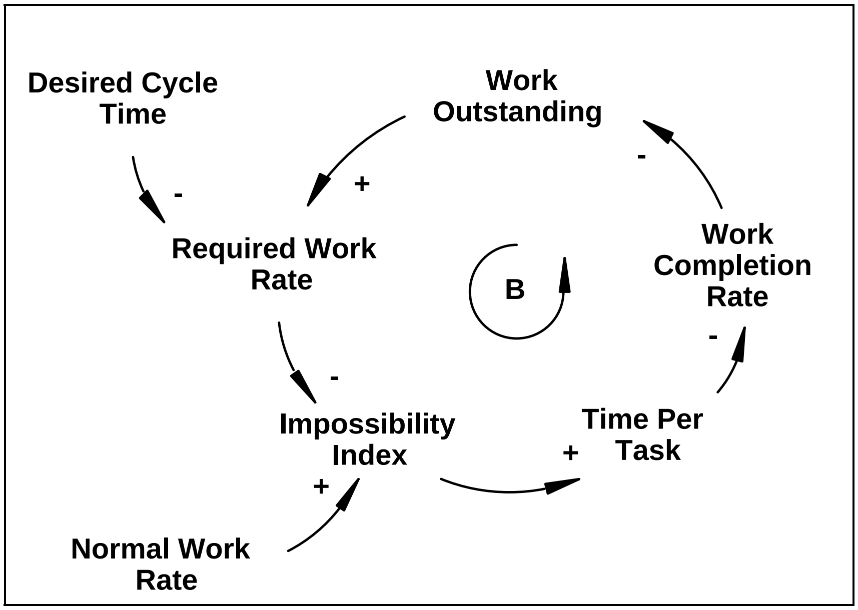
“What about overtime or additional resources?” you ask. “Doesn’t work that way around here,” somebody replies, adding:

> Overtime and extra resources cost money. Crawford believes you’ve got to keep everybody on a short leash when it comes to head-count and overtime, so it’s just not an option. Cutting corners is often the only way to get your work done.

“What about the engineering VP’s policy of firing anybody who doesn’t follow the process?” you inquire. Another engineer replies:

> He hasn’t developed a product or done a design in more than ten years. He has no idea what really goes on down here in cubicle-land. Besides, who’s going to tell him the truth, if it means getting fired? Everybody knows our design work quality is really bad. It’s just that our testing is basically perfect, so if something is wrong, we just keep doing it over again until we get it right. I’ve worked on lots of stuff that gets reworked four or five times before it’s finally released.

Your meeting is almost over, but before you go you quickly draw and explain the following diagram for the group. You ask the group if it plausibly represents how they get their work done.



They agree with your description, and one of them adds:

> The thing that management doesn't understand is that, unlike manufacturing, every time we do a new process thing—computer tools, project management, or whatever—it means that our normal work rate goes down. Every one of those things takes more time, not less. That means we have a lot less time to do real work. But around here real work takes priority. Pretty soon we have to dump the process thing so we don't fall behind. We know that some of that stuff would work if we gave it a chance, but we just don't have the time.

The Vice-President of Manufacturing

The next person you meet with is the head of the manufacturing organization. She had been Crawford's key assistant during the early days of the manufacturing improvements. You ask her how they were able to be so successful. She responds:

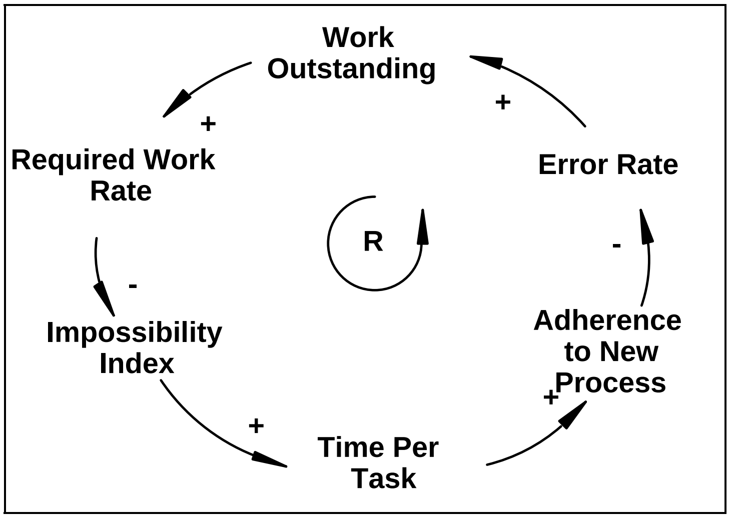
> Well, when we started, things were pretty easy. For example, when we first did cycle-time reduction, you could just walk around on the floor and find pretty big opportunities. That didn't take all that much effort. Later, things got more difficult, and we had to work harder to get the same rate of improvement.

You then ask about Crawford's strategy of stretch objectives.

> Oh, that. Yeah, he thinks he is a great leader because of the famous "big increase." Honestly, it really doesn't make that much difference to us. Usually what happens is that we start following a new process or using a new tool, and, yeah, it takes some more time to learn it and use it. But the early improvements come pretty fast, and that lets us cut the error rate so much that we can take a lot more work, even though each step might take a little longer. By the time the honeymoon ends and the work flow increases, we have

> usually gotten so far ahead that we can take the increase and still have a lot of time to spend on further improvement.

You draw another feedback loop. “So you’re saying that the process looks something like this.”



To test your understanding of her mental model, you say “I hear you saying that when adherence to the new process rises, the error rate drops, so more work gets done right the first time, and the backlog of work outstanding falls. That lowers the impossibility index, so your people can spend more time on each task, making it even easier to improve adherence.”

“Yeah,” she says, “that’s about how it works. There is a big snowball effect. Once we make some early improvements, the thing just takes off on its own.”

As a final question, you ask her why she thinks product development has struggled so much.

> It’s all about discipline. It’s just like in your picture: They never make the early commitment to following the new process. Those folks have gotten way more money and tools than we have, but they just don’t want to follow the process. In the last couple of years, Crawford has cut them a lot of slack, and all along they were telling him, “we’re making big improvements,” “the error rate is falling,” and so on. Well, as soon as their target throughput was cranked up a little bit, the thing just fell apart. I don’t think anything ever changed. They’ve never figured out that it’s more efficient to spend four hours doing something right the first time than to spend two hours doing it and then re-working it three more times.

The Vice-President of Process Improvement (again)

You next meet again with the VP of process improvement. You show her the loops you have developed so far. She is particularly interested in the second one. She says:

> In fact, we have just started trying to measure how much people follow the process. We look at how much more time people should be spending versus the old way of doing things. So if a certain step should take four hours, and it used to take two hours, and our measurement tells us that people are spending on average three hours, then they are following the process about 50%. It looks something like this.

She puts the following equation on her whiteboard:

$$\text{Adherence to the Process} = \frac{\text{(Avg. Time per Task – Time per Task Under Old Process)}}{\text{(Time per Task Under New Process –Time per Task Under Old Process)}}$$

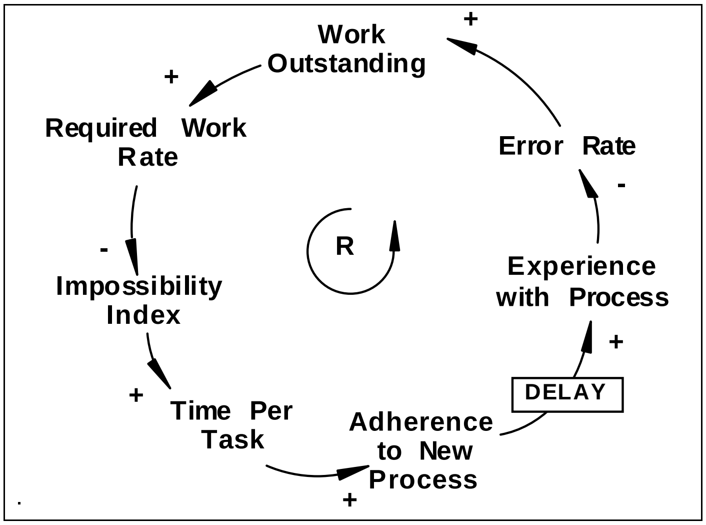
and explains,

> Full deployment of the new product development and process improvement tools would mean each task would take 4 person-hours of effort, because the engineers have to document their work, post it to the library of reusable designs, attend improvement team meetings, and so on. Under the old system, it was about 2 person-hours per task. So if they actually spend four hours per task, the adherence metric would be 100% [she does the math for you, though you brought your calculator and could have followed along]: Four hours spent on average less two hours under the old process, divided by four hours per task required less two required under the old process is 100% adherence to the process.

She adds:

> The big problem we've discovered, though, is that even if people follow the process 100%, it takes some time before it starts to show benefit. People need time to develop experience with the new tools or process.

You quickly change the causal diagram, and ask if it captures her idea.



She says it does and estimates that, on average, it takes about six months for people to develop experience using a new process.

The Process Improvement Consultants

Your next meeting is with a group of internal consultants who are responsible for running various improvement efforts. You show them your diagrams. One responds:

> Yeah, that makes some sense, but I see a problem with the arrows from adherence to experience to error rate. It just doesn't work that way. I see lots of groups following part of the process, but they are getting almost no results. A new process or tool just isn't

> effective until you get over a critical threshold. In one group I'm working with, the adherence metric is at about 20% right now, and I'd say they are getting less than 10% of the total benefit.

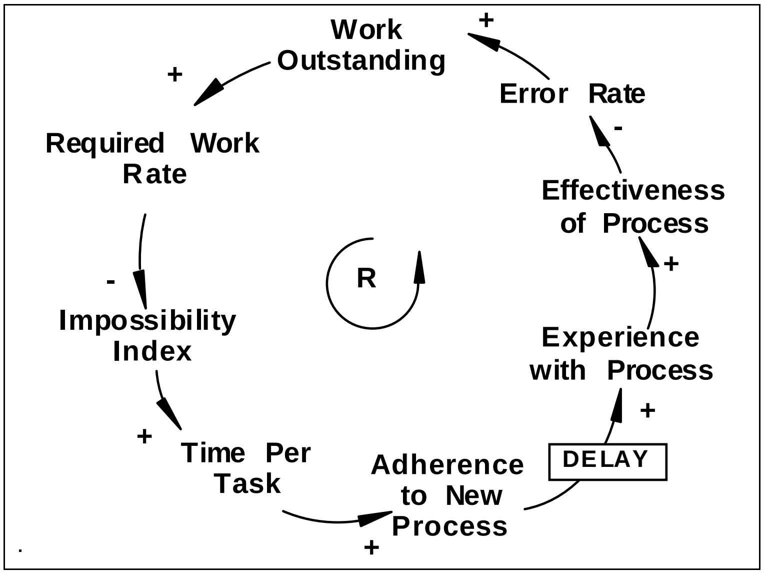
Another one adds:

> In my experience, if they can get that adherence metric past 25%, then things start to get better. If they can get up to 75%, then they've almost made it—they are probably getting 90% to 95% of the benefit.

A third says:

> With my groups, it seems like all the action is in between 30% and 70%. With less than 30% adherence they just don't know enough to get much benefit, and once they get beyond 70% there is a diminishing return.

You change your diagram again to capture this new wrinkle:



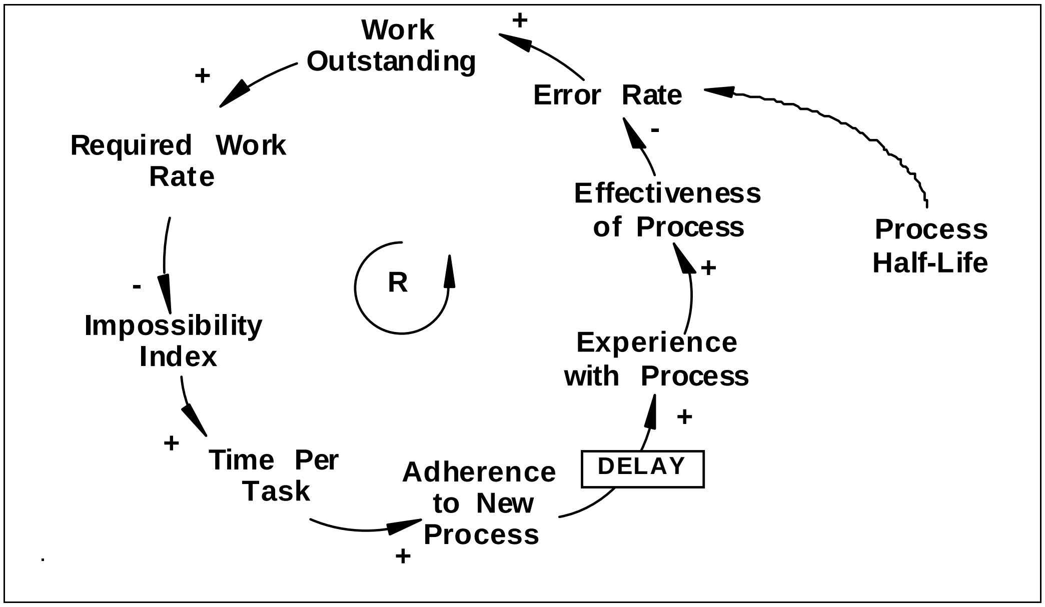
The Vice President for Engineering

Your last stop is with the VP in charge of the product development organization. You describe what you've learned so far and then ask him why he thinks the improvement efforts in PD have been unsuccessful. He replies:

> Product development is harder. The thing that Crawford doesn't understand is that PD and manufacturing are not the same. He thinks you can run a PD organization just like a manufacturing one. You can't. What we do is much more uncertain. I saw a study once where somebody looked at a whole bunch of improvement efforts and tried to figure out statistically how long it takes to cut the error fraction in half. I think he called it the "Process Half-Life." Well, the average for manufacturing was about six months. In product development it took about twenty-four months.[2]

[2] Schneiderman, A. (1988) "Setting Quality Goals," *Quality Progress.* April, 55-57.

To capture this idea you make one final modification to your diagram. The rate at which the error rate declines depends not only on the effectiveness of the process used, but also on the intrinsic difficulty of improvement, captured by the Process Half-Life. Processes that are hard to improve will see slower declines in the error rate than easier ones, even if adherence to the process and the effectiveness of the improvement effort is the same.



Final Meeting with the Modeling Team

At the end of your visit to Big El, you meet with the team of people assembled to supervise your effort. You discuss what you learned during the day. After reviewing the history of their failed attempts to improve the product engineering process, you suggest to the group that there was nothing wrong with the improvement techniques they tried. You've seen them work in other firms. Instead, there must be something about the context of implementation—interactions between the improvement programs and the rest of the organization—that has so far prevented Big El from realizing the potential for improvement. The group is not totally in agreement, but the team finds this an interesting alternative diagnosis to their problems.

You suggest that as a next step you build a model that is quite simple, avoiding the detailed process maps you've often seen in re-engineering efforts. These maps often fill the long walls of conference rooms, showing every step in the process in enormous detail, but don't include a single feedback loop. Your plan is to develop a simple generic model of process improvement. You will keep the model general, so it can be used to represent product development or manufacturing with just a few parameter changes.

As a starting point for the model, you describe the two feedback loops that you developed during the day. Before collecting more data, the team agrees that you will develop a first-cut model that captures the dynamics of just these two loops and present the results at the next meeting.

You ask them to give you a sample process on which to base your model. After some discussion, you arrive at the following specifications for your initial effort.

- Begin with a fixed head-count of 125 people, each working 200 hours per month.
- Since many in the group are focused on product engineering, you choose an initial error rate of 80%.
- The initial workload is 2,500 tasks per month, with each task requiring 2 person-hours of effort.

- The group also decides you should model the introduction of a new process that has the potential to reduce the error rate from 80% to 20%, but increases the time per task to approximately four hours (if the new process is followed 100%).

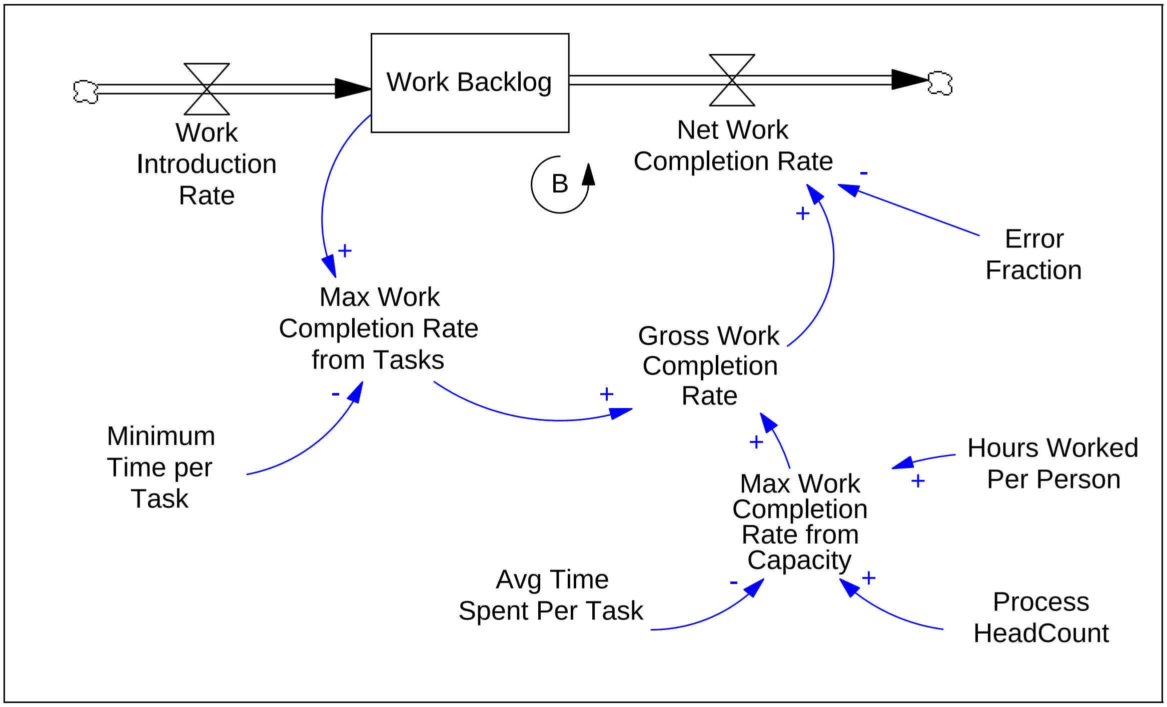
Your task in this assignment is to develop and analyze the model that you promised to your client team. You have two main objectives. First, you want to develop a convincing explanation of Big El's success in improving manufacturing and failure in improving product development. Second, you want to determine which policies and strategies lie at the core of these failures and propose changes that will lead to greater success.

Given the time available and the fact that this will be your client's first exposure to system dynamics, you need to keep your model simple and focus on the two feedback loops you presented at your last meeting.

A. The Basic Model: Closing the Negative Loops

The first thing you need to do is develop a simple model of a work process. At the most basic level, every process can be thought of as receiving a flow of new work, which accumulates in a backlog of work to be done until it is completed and delivered to the customer. In manufacturing, the flow of new work is the customer order rate, and the work completion rate is the shipment rate. In product design, the rate at which new work arrives can be thought of as the flow of new product ideas or projects into the development organization. The work associated with these projects remains in the backlog of products under development until the new designs are completed.

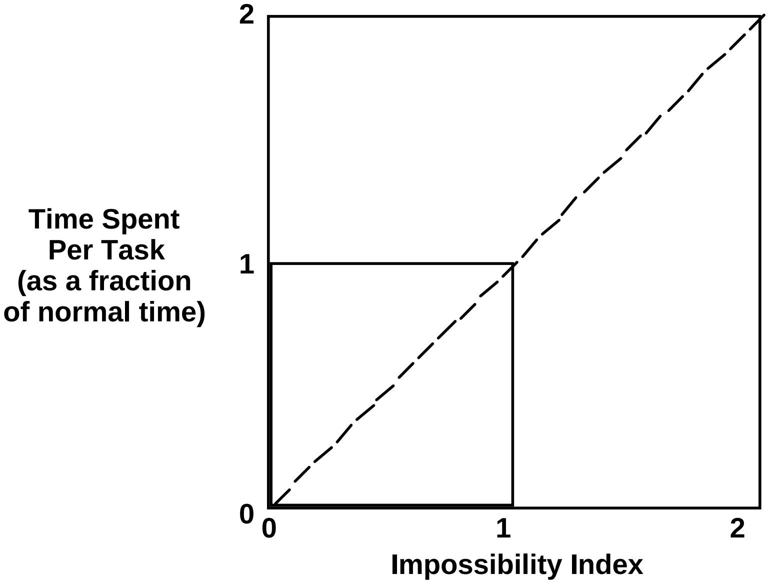
For this model, you decide to consider the rate at which new work arrives to be exogenous (you may decide to use built-in functions such as STEP and PULSE to disturb the system). The work completion rate, however, is endogenous. You observe in your site visits that the development engineers have plenty of computers, testing equipment, and so on. Design completion therefore depends on the number of hours each person spends working, the total number of people working in the process, the average time it takes to complete each task, and the quality of their work. Work quality is measured by the error fraction, the fraction of the effort applied that is erroneous or non-value-added work. To keep things simple, you decide *not* to represent rework or the stock of undiscovered errors explicitly. Instead, you assume that **tasks not completed correctly (those that are defective) remain in the backlog** and must be processed in the future. Thus the *Gross Work Completion Rate* represents the rate at which tasks are completed (per month). The *Net Work Completion Rate* represents the rate at which tasks are done correctly and exit the backlog of work to be done. The *Gross Work Completion Rate* is determined by the lesser of the rate at which work can be done based on the work to do and the rate at which the current staff can work, given their productivity and workweek. The resulting stock and flow structure is:



After developing this structure you go to your notes and decide to start with the negative loop that arose from your discussion with the engineers.[3]

Questions A1, A2, A3, and A4 yield a total of 3 points.

A1. Sketch the relationship between the impossibility index (the ratio of the normal work completion rate to the required work completion rate) and the amount of time an engineer spends on each task. Answer the following questions to develop this relationship as a table function for your model.



[3] You may want to review chapter 14 in *Business Dynamics*, which covers a work management system very similar to the one suggested here and also discusses the formulation of nonlinear relationships.

* You note that the *impossibility index*, defined as the ratio of the normal rate work can be done to the rate required, is the inverse of the *schedule pressure* idea you used in the Widgets model (Schedule pressure was defined as the ratio of the work required to the normal work rate). For example, an impossibility index of 50% means the engineers are asked to do twice as much as they can under normal conditions. This would correspond to schedule pressure of two. While it is tempting to put schedule pressure into the model to be able to reuse some of the formulations you developed in your previous engagement, you have wisely decided to use the terminology and concepts the Big El people use.

❑ a. Give an interpretation of the 45-degree reference line. What policy does it represent?

❑ b. Sketch your best estimate of the relationship between the impossibility index and time per task, given the information you developed in your fieldwork and your intuition. Explain how you arrived at this shape.

❑ c. How does your curve relate to the reference line?

A2. Now formulate the negative feedback loop you discussed with the product engineers.

❑ a. Write equations for the net work completion rate and its inputs starting with the basic process structure diagram presented above.

* Make sure the equations are robust and dimensionally consistent.

* Keep in mind that in any process there are at least two determinants of the work completion rate: (1) the capacity to do work and (2) the outstanding work to be done.

* Assume that if capacity is unlimited, there is still a minimum time required to complete the tasks. This minimum completion time is half a month.

* The minimum time required to do a task (in months) should not be confused with the effort required to do a task (measured in person-hours of effort per task).

❑ b. Using your interview data, develop equations for the rest of the negative loop you discussed with the engineers. Your formulation should match the loop you developed during your visit as closely as possible. You may need to create some additional auxiliary variables.

* Assume that the desired cycle time represents the desired time to complete a task, not the desired time to complete a task correctly (which may take multiple iterations). As a result, you should not need to take the error fraction into account when formulating the required or normal completion rates.

* Remember the desired cycle time and the minimum time required to actually complete a task are not the same thing. The desired cycle time represents how long tasks should take under reasonable, normal conditions; the minimum time represents how long it takes under extreme conditions and with heroic efforts from the people.

* Make sure your stock is initialized in such a way that your model always starts in equilibrium. To do so, remember that for a stock to be in equilibrium its inflows must equal its outflows. Hint: Given the shape of your table function, do you know where the model will be on the curve when it is in equilibrium?

Hand in a copy of the Vensim diagram and a brief verbal description of your formulations. (Do not hand in model documentation for this portion.)

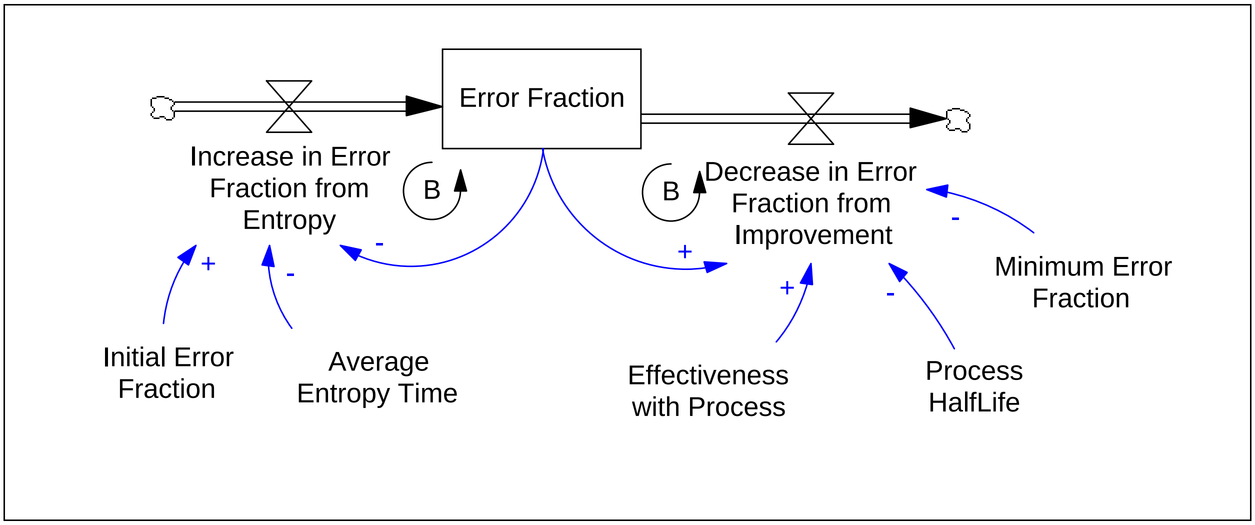
A3. Test your formulation to make sure it works as you expect. While performing these tests, assume that the error fraction is constant.

❑ a. Subject your model to pulses of different sizes (including increases and decreases) in the rate of incoming work. How does it respond? Turn in plots of the work backlog and the time spent per task with all the runs you perform shown on one graph.

❑ b. Does your model return to its original equilibrium for every test? What formulation in your model determines whether or not this happens?

You now need to formulate a model of the improvement process.

The improvement model shown here is based on the "half-life" model of process improvement developed by Schneiderman (1988).[4] Schneiderman showed that, for a wide range of processes in a large sample of firms, "any defect level, subjected to legitimate QIP [quality improvement processes], decreases at a constant [fractional] rate. . . ." The result is an exponential decline in defects characterized by the "improvement half-life"—the time required for defects to fall by 50%.


Error Fraction
Increase in Error Fraction from Entropy
Decrease in Error Fraction from Improvement
B
B
+
-
-
+
+
-
-
Initial Error Fraction
Average Entropy Time
Effectiveness with Process
Process HalfLife
Minimum Error Fraction

The basis for the half-life dynamic is the iterative learning loop at the heart of improvement programs. Participants in quality improvement processes diagnose the root causes of the defects in any process and rank them in order of importance using tools such as Pareto charts and Ishikawa diagrams. They then design, test, evaluate, and implement solutions. The improvement team continues to cycle around the learning loop until the source of defects is corrected, then moves on to the next most important source of defects. In your model, this means the Potential Improvement Rate is given by:

$$(E - E_{min})/\text{Process Improvement Time}$$

where E is the current error fraction, $E_{min} \geq 0$ is the theoretical minimum error fraction, and the Process Improvement Time is the average time required to close the gap between the current and theoretical error fraction. You note that you worked with this structure in an earlier assignment.

[4] Schneiderman, A. (1988) "Setting Quality Goals," *Quality Progress.* April, 55-57.

Schneiderman found that product development processes have long improvement half-lives due to their high technical and organizational complexity (compared, say, to manufacturing). Consistent with the experience of other firms, you estimate that the improvement half-life for product development at Big El is about 24 months. Recalling the "rule of 70," you note that a two-year half-life corresponds to a Process Improvement Time of 24/.70 ≈ 34 months.[5] Assume the theoretical minimum error fraction is 0.2.

The half-life model above is a good starting point for your model of Big El's improvement program. However, it has two problems you must address.

First, in Schneiderman's model, once you make an improvement, it never goes away. Unfortunately, process capability declines over time as equipment ages, workers quit and new ones are hired, the product mix changes, etc. To capture this, assume that the error fraction is always slowly moving towards some "natural" error fraction. Also assume that the model starts at the natural error fraction, so that you can assume the initial error fraction and the "natural" error fraction are equal. The following equation represents one way you might capture this "entropy" in your model (assume the average entropy time is 120 months):

Increase in Error Fraction from Entropy =
(Initial Error Fraction-Error Fraction)/Average Entropy Time

Second, process improvement doesn't just happen. As many of your interviewees mentioned, people need to develop some effectiveness with the new process and the improvement tools before they start generating improvement.

A4. Using the information given above, formulate a model of the improvement process.

❑ a. Write an equation for the decrease in the error fraction from improvement that captures the basic fact that improvement requires some type of human input, and briefly describe your formulation.

* Assume that the improvement rate determined by the current and minimum error fractions and the process improvement half-life represents the *maximum* possible rate of improvement—that is, the rate of improvement that could be achieved if the workers were 100% effective with the process. (What does this suggest about how you might want to formulate Effectiveness of Process?)

❑ b. Assume, for the moment, that the effectiveness parameter is constant at its maximum. Simulate just the structure shown in the diagram above for the two different potential half-lives (6 and 24 months, corresponding to the manufacturing and product development settings). Plot, and turn in, the error fraction for both runs on a single graph.

B. The Basic Model: Closing the Positive Loop

Once you are satisfied with your formulations for the negative loops, turn your attention to the positive loop. You will need to use information from a number of your conversations to develop this loop. Previously, you could treat the normal time per task as constant. In this section you must now make it a variable that changes when a new program is introduced. A common and easy way to do this is to create a variable, sometimes called a "switch," formulated as follows:

[5] Recall that for an exponential decay process with time constant τ, the rate of change of the fraction remaining F is $dF/dt = -F/\tau$, and the behavior of F is given by $F(t) = \exp(-t/\tau)$ where τ is the time constant. The half-life t_h is given by the solution to $F(t_h) = .50 = \exp(-t_h/\tau)$, or $t_h = \tau \ln(2) \approx .70\tau$. Thus if the improvement half-life is 24 months, the time constant is ≈ 34 months. See chapter 8 in *Business Dynamics*.

Program Introduction Switch = Step(1, Introduction Date)

You choose the date you wish the program to start, the *Introduction Date*. Assume that all programs are introduced in month 24. The switch will return a value of zero until the Introduction Date is reached and will return the value of one ever after. To represent the change in the normal time per task caused by the introduction of a new process or tool you can use a formulation similar to the following:

Normal Time Per Task =
Time Per Task On Standard Process *(1-Program Introduction Switch)+
Time Per Task On New Process * Program Introduction Switch

The normal task time will then change from that required by the standard process to that required by the new process whenever the introduction date is reached.

* After adding this formulation to *Normal Time Per Task* you may want to revisit your formulation from part A. Specifically, if *Normal Time Per Task* changes with the introduction of a new tool or process, do engineers immediately factor this change into their decisions about time spent per task? If not, how might you capture this dynamic?

Questions B1, B2, and B3 merit 3 points total.

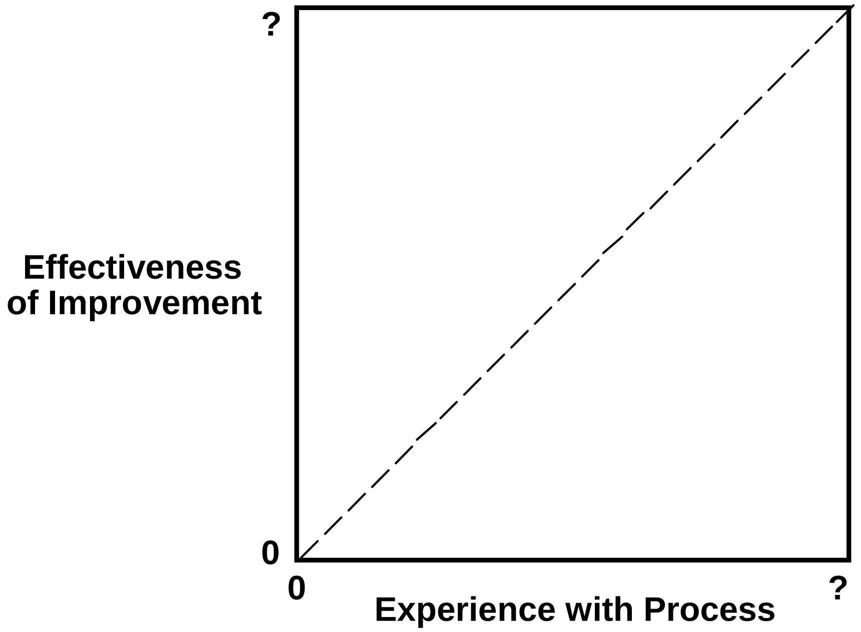
B1.

❑ a. Write an equation for the variable Adherence to New Process. Use the metric suggested to you by the Vice-President for Process Improvement as the basis for adherence. Make sure your formulation is robust under extreme conditions.

❑ b. The diagram you developed with the VP showed a delay between using the new process and becoming an experienced user. Formulate an equation for this delay (the input will be adherence, the output will be experience). Briefly explain your choice of delay type, order, and length.

* Adherence is defined on [0,1]. It may be helpful to define experience on the same interval.

B2. Now you need to determine the relationship between people's experience with improvement and the effectiveness of their improvement effort. Again, use your interview data as the basis for answering the following questions and developing a table function.



❑ a. Give a brief interpretation of the reference line (the 45-degree line).

❑ b. Choose appropriate limits on the X and Y axes (replace the "?" with numbers).

❑ c. Sketch below the relationship you think is most plausible. Explain in two sentences or less why you made this choice and, in one additional sentence, explain its relation to the reference line.

❑ B3. You should now have a complete model that captures both feedback loops but little else. Make sure your model is initialized in equilibrium, and run a base case in which no improvement program is introduced (you can do this by setting the improvement start date to a very large number). Do not turn in the base case, but make sure your model is in equilibrium. Document your model and hand in the Vensim diagram with the main feedback loops clearly labeled. Include a fully documented equation listing. As always, make sure your units balance.

C. Analyzing the Model for Policy Evaluation

You should now have a running model that captures the two feedback loops you discussed with your clients. Prior to presenting your model to them, however, you need to make sure you understand its behavior and can recommend some policies based on what you learned.

Questions C1, C2, C3, and C4 are worth a total of 3 points.

❑ C1. Run the base case of your model using a constant work inflow rate (the 2,500 tasks per month discussed earlier) for both the manufacturing and product development parameters. Run the model for 120 months and assume that the new improvement program is introduced in month 24. Also, using the estimates you received from the Vice-President of Engineering, assume the improvement half-life is 6 months in manufacturing and 24 months in product development. Select and turn in a small number of graphs that compare these two runs, and provide an explanation for why the behavior is different for the two. Keep in mind that your ultimate task is to explain to your clients why they are successful in manufacturing and unsuccessful in product development.

C2. Test Crawford's policy of reducing throughput by 20% for one or two years after the program is introduced and then doubling it after the two-year period ends.

* You can do this by modifying the equation for the rate of work inflow using step functions such as the following:

Work Introduction Rate = Initial Work Introduction Rate
– STEP(Throughput Cut, Introduction Date)
+ STEP(Additional Throughput, Introduction Date + Length of Honeymoon).

where the Initial Work Introduction Rate is 2500 tasks/month and the Throughput Cut is the drop in new work during the honeymoon period Crawford allows. After the honeymoon ends, the rate at which new work arrives rises by the Additional Throughput.

Set the Throughput Cut to 500 tasks/month and Additional Throughput to 2000 tasks/month. Crawford indicated that the Length of Honeymoon is 24 months. With these parameters, the rate at which new work arrives will be 2500 tasks/month up to month 24 (the Introduction Date for the improvement program), then drops to 2000 tasks/month for two years, before finally doubling to 4000 tasks/month.

❑ a. Does Crawford's policy of a stretch objective and honeymoon period work in manufacturing? Why, why not?

❑ b. Does Crawford's policy work in product development? Why, why not?

❑ c. What modifications to Crawford's policies might you recommend to help Big El successfully improve product development? Test these in your model and explain the results.

C3. Think about other policies that might help Big El succeed in its process improvement efforts.

❑ a. Based on your model so far, what other "levers" are there for managers to pull?

❑ b. Test two of these policies in your model. Turn in a small number of graphs that show the results of your effort in the manufacturing and product development cases.

❑ c. Briefly explain why these policies do or do not work and why they may or may not be effective in the different areas.

❑ C4. In your interviews, many of the interviewees pointed to cultural variables and human factors as the main cause of Big El's problems (e.g., engineers are rebel artists who don't like discipline; the PD VP has lost control). Your model does not explicitly capture any of these variables. Briefly explain how the physical structure of the system you are modeling may have led people to incorrect conclusions about the true cause of their process improvement difficulties.

D. Critiquing the Model Boundary and Exploring Next Steps (*1 point*)

Assuming your presentation goes well, Big El will most likely want you to continue your efforts. In thinking about next steps, answer the following questions.

❑ a. Critique the boundary of your model. What are its main strengths and weaknesses?

❑ b. List the important feedback loops that have been omitted from your model (limit your discussion to five loops). Which of these do you think is the most important to add to the model? Why?

E. Optional *Challenge* Question (*2 points extra credit*)

❑ a. Add the feedback loop you just identified to your model. Hand in the Vensim model diagram and a fully documented listing of the new or modified equations.

❑ b. Simulate your model. Present a small number of graphs that show how the new loop changes your model's behavior and explain why.

❑ c. *Briefly* discuss whether and how the new loop might change the policy recommendations you made above.