System Dynamics Group
Sloan School of Management
Massachusetts Institute of Technology

System Dynamics for Business Policy, 15.874
Prof. James Lyneis

Assignment 9

Understanding Cost & Schedule Overrun on Product Development Projects[*]

Please work on this assignment in a group of three people.

Projects to design and build a new product or service, whether a one-of-a kind effort such as the Channel Tunnel between England and France, or more repetitive efforts such as new car product developments or software systems, are notorious for over-running their original schedule, or budget, or both. Such projects are increasingly important in our economy. For example, almost every new product or service today involves development of new software. Success in the market can depend on being on schedule and budget. This assignment builds your understanding of some of the factors driving success or failure on projects.

Case Background: You have been hired as a consultant to the IT department of a major financial services institution. The department is responsible for development, extension, and maintenance of software to support the growing range of products and services offered by the institution. The department has not been doing a very good job. All of its software development projects over the last few years have significantly exceeded both their original budgets and schedule. The Vice President in charge of the IT Department put it this way: "Our projects start out okay, but about halfway through we begin to discover additional work, often to correct errors made on earlier work. When we realize we are behind we add additional resources as they are freed from other assignments. But we barely keep pace with the workload, and end up throwing lots of staff at the job. And we still don't finish on time. I beginning to wonder if there isn't something to Brooks' Law.[1]"

You interview some of the key project managers from the department, and come up with a long list of factors that might be contributing to the problems in the IT department:

[*] Originally prepared by James Lyneis, April 2000. Last revision, April 2001.

[1] "Adding manpower to a late software project makes it later." Brooks, Frederick P. Jr. The Mythical Man-Month. Reading, MA, Addison Wesley, 1995.

❑ 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.

- Uncertain specs from the operating departments
- Delays in getting additional staff
- Design errors discovered later in the project
- New staff unfamiliar with the project
- Changes in needs of the operating departments
- Overwork and fatigue among the programming staff
- Time spent getting people brought on a project up to speed
- Programmers not doing a thorough job checking their work
- Pressures to promise delivery earlier than is reasonable
- Building design errors into coding, and not discovering the problem until testing

There is a lot of pressure on you and your team to immediately begin developing a comprehensive model of the IT department. However, you know that the most effective way to develop a model, and to engage and educate the client along the way, is to build it in small steps. With this in mind, you work through the four steps below, and at the end have some significant preliminary insights for your client.

ơ Brevity is a virtue in your writeup. Unless specifically requested, it is not necessary to hand in complete sets of output (graphs, tables) for each test and simulation you do. A summary table will suffice. For example, you might construct a table showing the date of project finish and cumulative effort expended. However, as always, you must explain the changes you make in the equations so that an independent third party can replicate your simulations.

A. Step 1 -- The Rework Cycle (1 point)

To begin your analysis, you hypothesize that the "rework cycle" is likely to be at the heart of the IT department's project problems. You therefore construct a rework cycle model of a typical department project, as illustrated in the following figure. Your interviews indicated the following:

A typical project involves 100 tasks
Under optimal conditions, each programmer accomplishes 1 task per month
Normally, programming error rates are 25%
It takes about 4 months to discover design errors
A typical project starts with 4 staff

Download the Rework Cycle model.

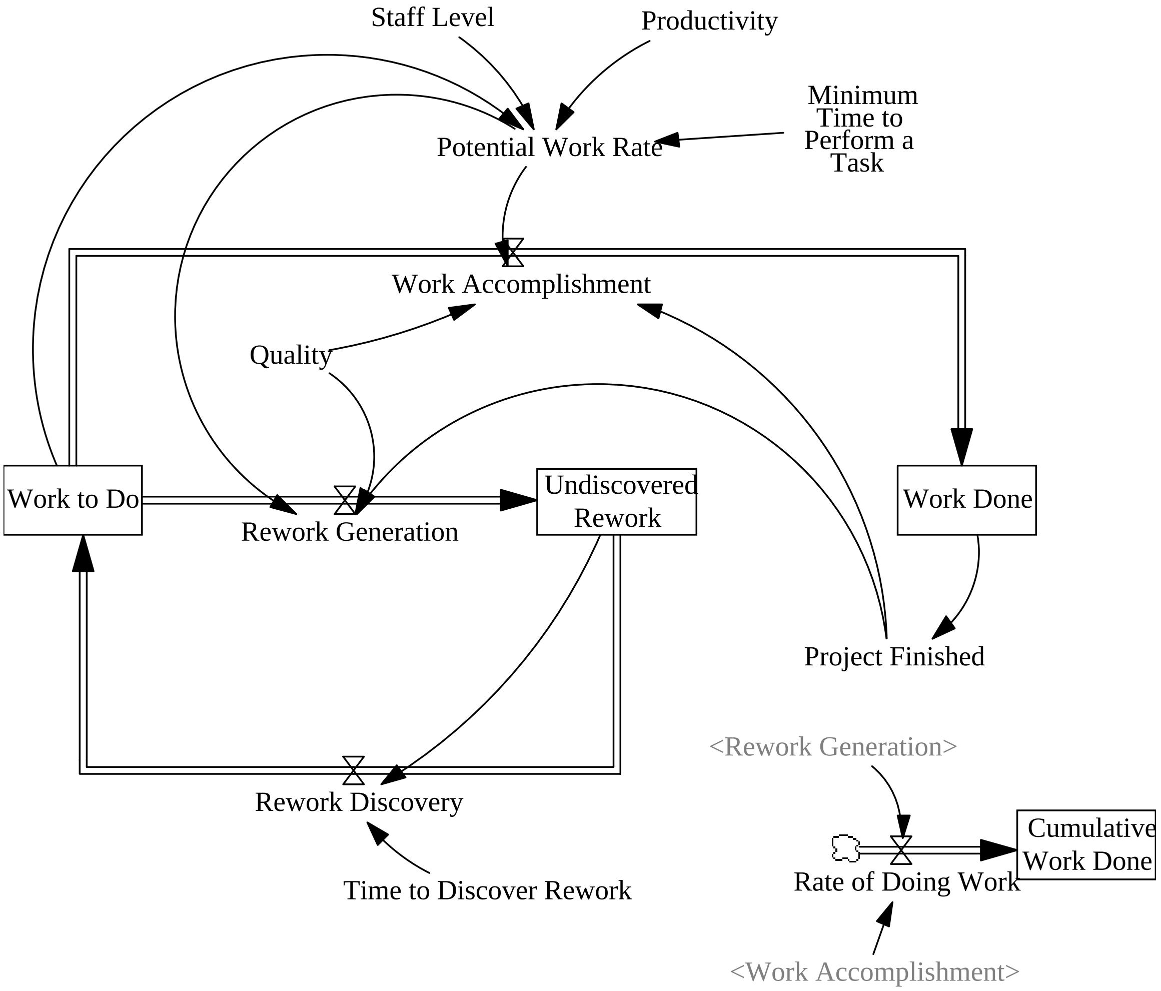
ơ Project Finished is a switch which is 1 normally, but becomes 0 when work done exceeds 99% of original work to do. Vensim’s IF THEN ELSE function is used. Further work stops at this point.

ơ "Cumulative Work Done" is included as a performance measure. This integrates both work accomplishment and rework generation, and therefore keeps track of the total amount of work done and redone.

♂ In trying to understand potential work rate, think about what happens when work to do is 0 or very low. Assume that the minimum time to perform a task is 0.25 months.

♂ Since initial work to do will be used in other variables, the initial value is defined as a separate variable which can be linked to other variables.

The Rework Cycle:



❑ A1. As a test case, if programmers did not make any errors (i.e., quality = 1.0), when would the project finish? What happens to work to do, work done, and undiscovered rework? Confirm that the model behaves as your intuition suggests before proceeding.

♂You will want to create custom graphs to show all the important variables.

❑ A2. Now, set the value for normal quality to that indicated in your interview notes. When does the project finish? What happens to undiscovered rework in this situation?

❑ A3. Which factors do you think are more important in determining project completion -- productivity, quality, or rework discover time? (*Please answer this before simulating your model.* Your grade is not affected by the accuracy of your answer to this question)**.**

❑ A4. Analyze the impact of productivity, work quality, and rework discovery time on project completion date and total work done. Perform this analysis by conducting sensitivity tests varying each parameter by plus-and-minus 33% (this step should require 6 computer runs in addition to the reference or Base Case when the parameters are at there normal values, for a total of 7 runs).

♂Use a quality of 0.75 as the mid-point for your analyses

♂A spreadsheet chart is a useful way to summarize the results of these tests.

B. Step 2 -- Extending the Model: Adding Variable Rework Discovery Time and the Quality on Quality Feedback (2 points)

After exploring the behavior of the rework cycle model with your client, you return to the office to expand the model to better reflect some important feedback effects. Your discussions around the rework cycle indicate that

1. ***Time to Discover Rework*** is not likely to be constant, but falls as progress on the project progresses. Your discussions indicate that rework discovery time is long initially, about 12 months, but once perceived project progress passes 50% complete and testing begins, rework discovery time falls below the initial value of 12 months. In the latter stages of the project, when almost everything is testing and error correction, the discovery time is approximately 1 month. The table below describes their best estimates.

Fraction Complete	Effect on Rework Discovery
0	1
.1	1
.2	1
.3	1
.4	1
.5	1
.6	.95
.7	.8
.8	.45
.9	.2
1.0	.1

2. The phenomena of errors building on errors happens often in software development, and is a particular problem in the IT department. Your discussions indicate that while average quality may be 75% or less, management believes that "normal quality" would be about 85% were it not for these "***quality on quality***" and other effects (to be discussed in Step 3). You probe project personnel on the likely effect of prior quality on current quality. They estimate that if all prior work were done incorrectly, then the effect on current work quality would be 10%. The table below describes their best estimate.

Average Work Quality	Effect on Current Quality
0	.1
.1	.25
.2	.35
.3	.45

.4	.55
.5	.65
.6	.725
.7	.8
.8	.875
.9	.95
1.0	1.0

- ♂Working from the Step A model, *first* add variable rework discovery time, describe the revised behavior, and then add quality on quality and describe the behavior (see questions below).
- ♂Assume that staff and scheduled completion date remain constant (although actual completion date may exceed that scheduled).
- ♂Include as a performance measure "cumulative effort expended". This variable integrates staff (in people) to determine person-months of total effort.
- ♂Fraction perceived to be complete is a key performance measure used by management to take actions. You realize that perceived progress likely includes both work done and undiscovered rework, as management thinks this is done at any point in the project.
- ♂Consider formulating quality as

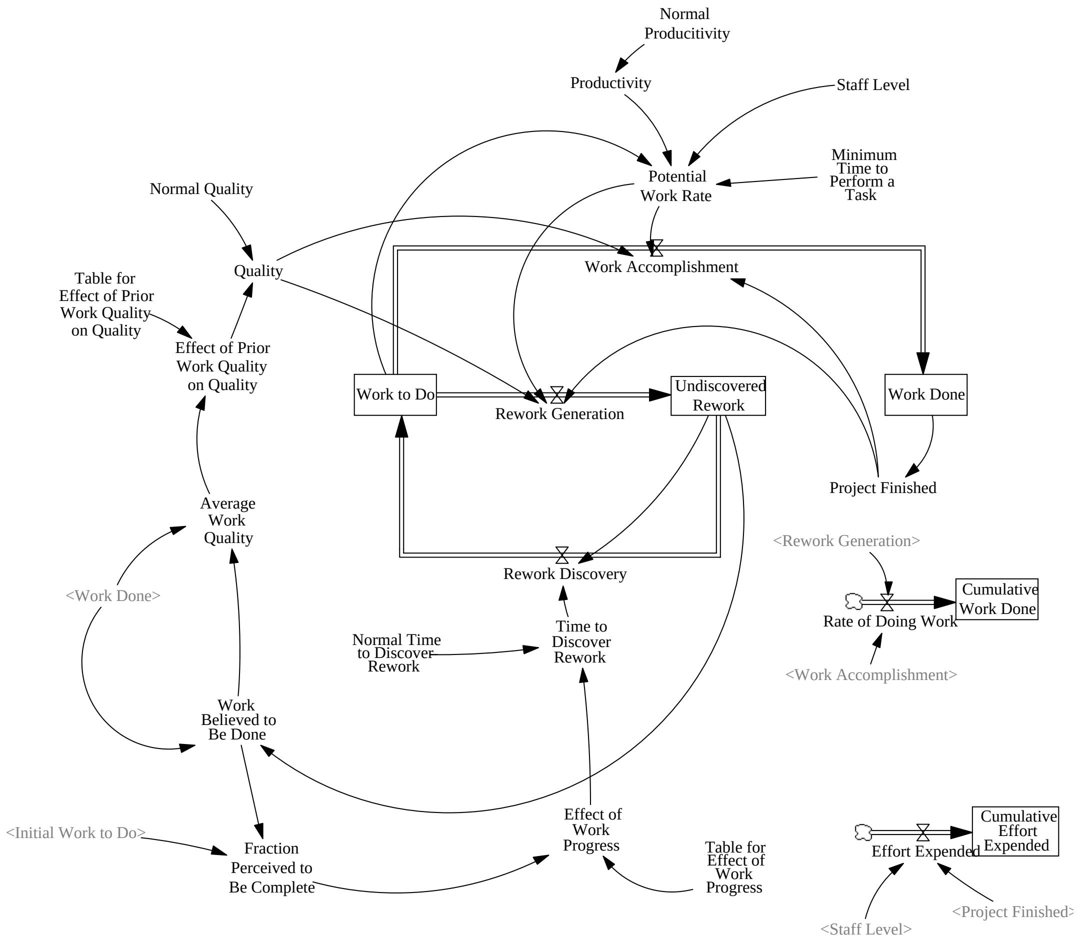
  Quality = Normal Quality * Effect of Prior Work Quality

- ♂Effect of Prior Work Quality should be driven by the average quality of work to date, using VENSIM's Lookup function. Errors embedded in past work products can either cause errors to be made on current work, and/or slow progress on current work as ambiguities need to be sorted out. Some definitions --

  Average Quality of Work done to date
  = WORK DONE/(WORK DONE + UNDISCOVERED REWORK)

You may need to use Vensim's IF THEN ELSE function to prevent division by zero when work done is zero; under that condition, set average quality equal to normal quality.

The revised model diagram is:



□ B1. How does the inclusion of a variable rework discovery delay affect the behavior of the project as simulated in Part A? What happens to completion date and the work backlogs as compared to the case when rework discovery time is constant? [**Checkpoint**: the project should finish 5-6 months sooner that the Step 1 Model.]

□ B2. How does the inclusion of quality on quality feedback affect the behavior of the project simulated in B1? What happens to quality, completion date and the work backlogs as compared to the prior? [**Checkpoint**: the project should finish around month 33, about 1 month sooner than in Step B1.]

□ B3. Hand in a documented listing of the new equations in your model.

C. Step 3 -- Extending the Model: Allowing for Increased Staff (3 points)

Your model is beginning to behave like a real project, and now you can begin to use it to explore the consequences of the IT departments project staffing policies. In order to do this, you need to add the ability to increase and decrease staff on the project. You also need to include the consequences of adding staff on productivity and quality. Your interviews indicate:

- Because staff are fully engaged on other projects, or because hiring takes time, the time to increase staff via hiring or transfer averages 4 months
- New staff coming onto a project require 24 months to gain full experience
- Inexperienced staff work at 50% the productivity and quality of experienced staff

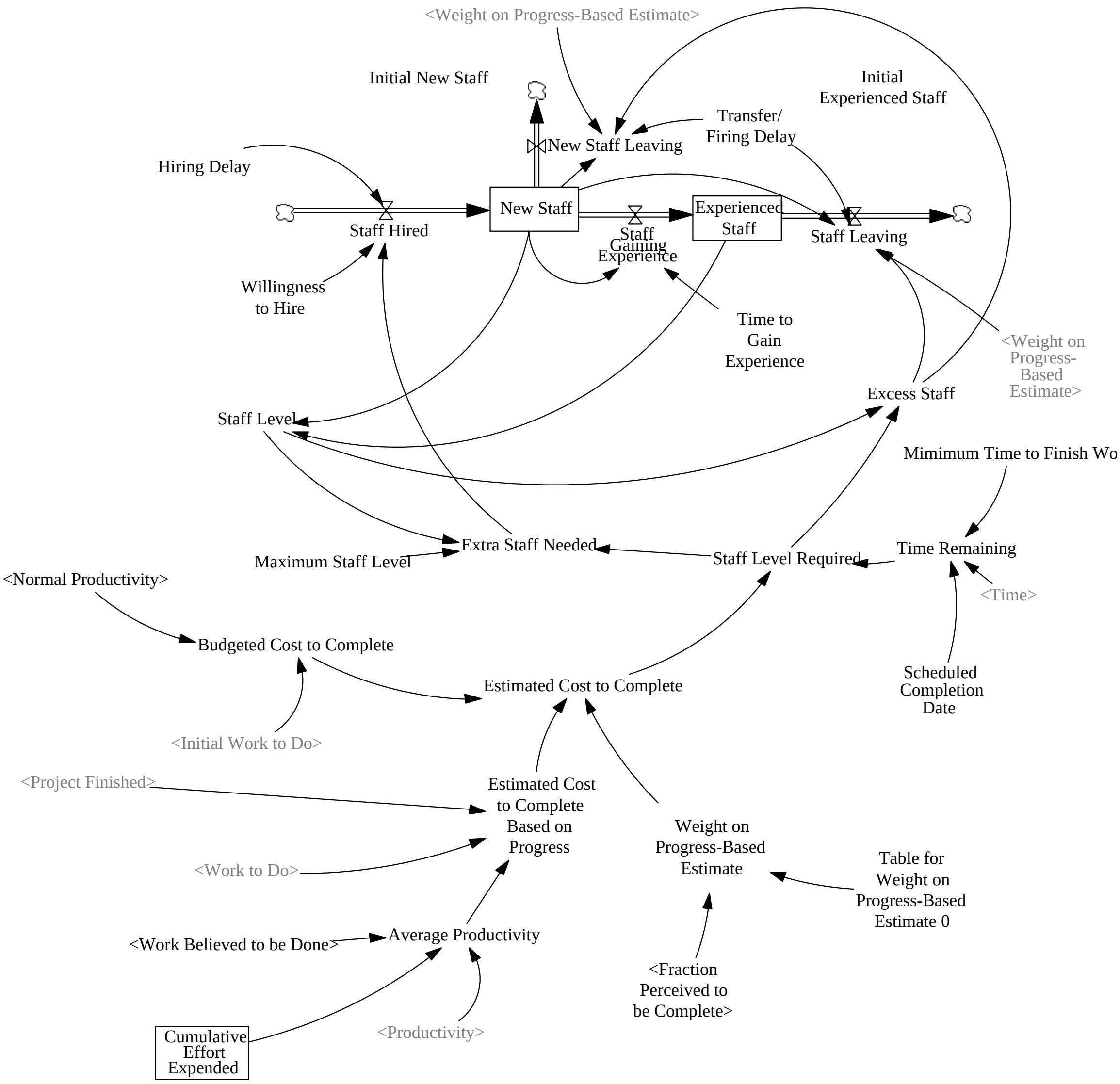
- Inexperienced staff also reduce the productivity of experienced staff as a result of the need to train and coach these new staff; inexperienced staff also reduce the quality of work by experienced staff, because time pressures cause them to make more errors. Management estimates that both of these effects are directly proportional to the fraction of staff which are new
- There is no limit on staff available or allowed to work on the project
- Management estimates the staff level required to finish the project by dividing estimated cost to complete (in person-months) by scheduled time remaining. Since scheduled completion date is fixed, they assume that a minimum of 1 month will be used once the project passes this date. The typical project is scheduled to complete in 30 months. The IT department estimates cost to complete, in person-months, by dividing work remaining to be done by average productivity on the project; average productivity is estimated by dividing cumulative work done to date divided by cumulative person-hours to date. *The structure and equations for computing staff level required and estimated cost to complete are given in the appendix and in the file Staff Required.MDL.*

♂ Create two levels, new staff and experienced staff. Assume that all hires or transfers to the project enter the pool of new staff, and that they take 24 months to gain experience.

♂ A simple mathematical average can be used for effect of experience on productivity and quality, rather than a graphical function as for the other effects.

♂ A diagram of the new staffing section is given below. A tricky formulation in this model regards the behavior early when little work has been accomplished. Because rework discovery takes time to cycle back to work to do, early in the project work believed to be done, and average productivity, appear to be high. This often indicates that fewer staff are required, and transfers off the project would occur. However, a smart manager recognizes this, and until sufficient progress has been made to really determine where things stand, uses budget rather than progress-based estimates for staffing levels, and does not transfer or lay off staff. You will note in the diagram that "weight on progress-based estimate" affects estimated cost to complete and staff leaving. Weight on progress-based estimate is a function of fraction perceived to be complete, as indicated below (and is also given in the file Staff Required.MDL):

Fraction Complete	Weight
0	0
.1	0
.2	0
.3	.1
.4	.25
.5	.5
.6	.75
.7	.9
.8	1
.9	1
1.0	1



- Willingness to hire is a multiplier which reflects management's willingness to hire new staff . In theory, this could range anywhere between 0 and 1. For this assignment, you will probably use either 0 or 1: 0 when you want to turn off hiring, and 1 when you want to hire.
- Some other suggested parameter values: transfer/firing delay of 4 months and hiring delay of 4 months (you may want to test other values); initial new staff of 0, and initial experienced staff of 4 people; maximum staff level -- any large number unless you want to test a smaller value to constrain staff and hiring to a limit.

❑ C1. Describe the behavior of the simulated project when staff are increased in order to meet the originally scheduled date. Does the project finish on time? Does it finish sooner than when staff are not added to the project? Why or why not? (Note: while several computer runs may be necessary to get your model working properly, only one computer run is necessary to answer this question).

♂Look at the behavior of productivity and quality, and the factors which cause them to change over time. How does the addition of staff later in the project affect productivity and quality? What does this do to the total amount of work done and to cumulative person-years spent on the project?

❑ C2. What policies would you recommend at this point?

D. Step 4 -- Policy Analysis: When Does Adding Labor Make Sense (3 points)

While the results thus far seem to be consistent with the behavior of the IT Department, you do not want to end with the recommendation that staff should never be added to a late software project. There must be some conditions under which adding staff makes sense. Conduct a series of analyses to determine when, given the model developed thus far, project performance in improved when staff are added. You might consider assumptions regarding:

Relative productivity of new staff
Relative quality of new staff
Effect of new staff on productivity and quality of experienced staff
Time required for new staff to get up to full productivity and quality
Time required to increase staffing
Initial number of staff

❑ D1. Prepare a summary of the results from your sensitivity experiments, e.g. using a table. Under what conditions does it make sense to add staff to get a project completed on time? Why? (Approximately 10-15 computer runs should be sufficient for this question).

E. Synthesis of Policy Analysis and Recommendations (0.5 points)

❑ E1. Prepare a recommendation for the management of the IT department regarding its project staffing policies. What would you suggest regarding initial staffing and/or schedule? About adding staff as the project progresses?

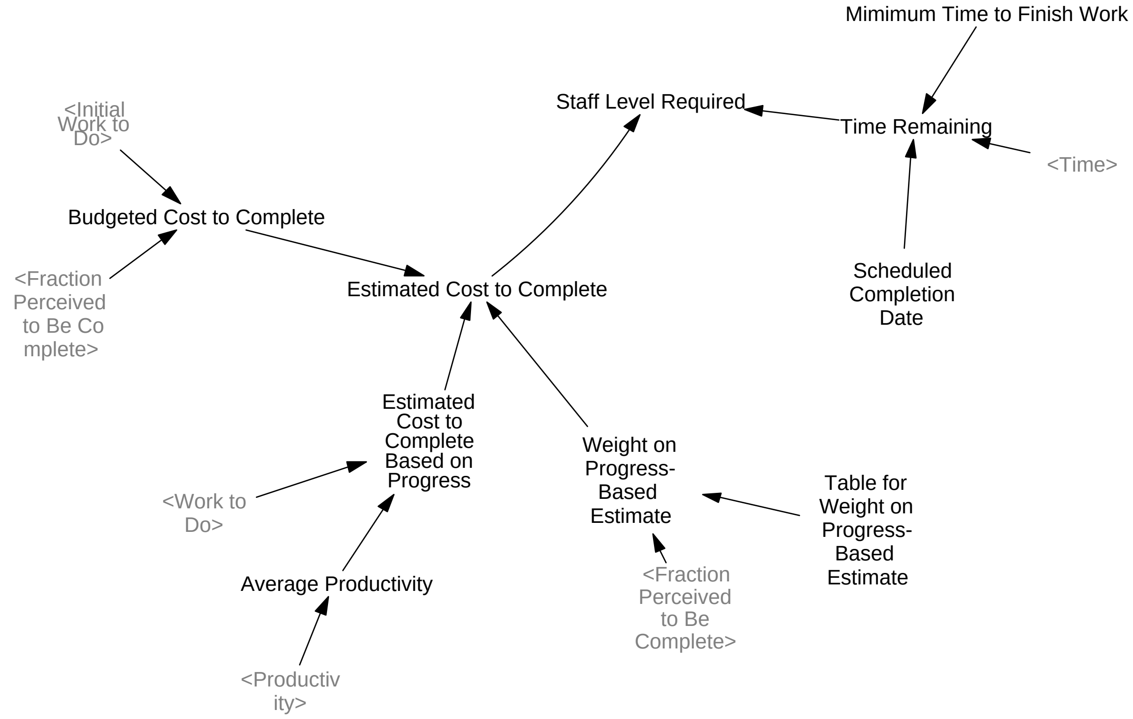
❑ E2. Hand in the diagram and equations for your full, documented model.

F. Assignment Critique (0.5 points)

❑ F1. This is a new assignment and you are the second class to work on it. We would appreciate your candid thoughts about :

- How long it took relative to other assignments (which parts took longest?)
- Whether or not you found it interesting
- How it could be improved

Appendix: Equations for Staff Level Required Contained in File Staff Required.MDL


Mimimum Time to Finish Work
Staff Level Required
Time Remaining
<Time>
<Initial Work to Do>
Budgeted Cost to Complete
<Fraction Perceived to Be Co mplete>
Estimated Cost to Complete
Scheduled Completion Date
Estimated Cost to Complete Based on Progress
Weight on Progress-Based Estimate
Table for Weight on Progress-Based Estimate
<Work to Do>
Average Productivity
<Fraction Perceived to Be Complete>
<Productiv ity>

```
Average Productivity=A FUNCTION OF(Average Productivity,Productivity) ~~|
Average Productivity=
        IF THEN ELSE (Cumulative Effort Expended>0, Work Believed to Be
Done/Cumulative Effort Expended\
                ,Productivity)
        ~       Task/(Month*Person)
        ~       A good estimate of the average productivity on project work to date is \
                given by work believed to be done (tasks) divided by cumulative effort \
                (person-months).  In order to avoid division by 0 and a discontinuity in \
                average productivity, the average is set to actual productivity before \
                work starts.
        |

Budgeted Cost to Complete=A FUNCTION OF(Budgeted Cost to Complete,Fraction Perceived
to Be Complete\
                ,Initial Work to Do) ~~|
Budgeted Cost to Complete=
        (Initial Work to Do/Normal Productivity)*(1-Fraction Perceived to Be Complete)
        ~       Person*Months
        ~       Managment's initial estimate for cost in person-months is given by initial \
                work to do divided by productivity.  As progress is made and fraction \
                complete increases, the estimate of effort remaining based on the budget \
                decreases.
        |
Estimated Cost to Complete=
        Budgeted Cost to Complete*(1-"Weight on Progress-Based Estimate")+Estimated Cost
to Complete Based on Progress
        *"Weight on Progress-Based Estimate"
        ~       Month*Person
        ~       Estimated cost to complete is management's estimate of the person-months \
                of work remaining to complete the project.  Early in the project, \
                management bases its planning on the budget.  As progress is made, \
                however, and management can determine the true scope and productivity on \
                the project, the estimated cost to complete moves toward that based on \
                actual progress.
        |

Estimated Cost to Complete Based on Progress=A FUNCTION OF(Estimated Cost to Complete
Based on Progress\
                ,Average Productivity,Work to Do) ~~|
Estimated Cost to Complete Based on Progress=
        (Work to Do/Average Productivity)*Project Finished
        ~       Month*Person
        ~       Work to do divided by average productivity to date provides an estimate of \
                how many person-months of effort remain to complete the project.  Because \
                actual project scope (including the amount of rework) and actual \
                productivity are hard to determine early in the project, this estimate may \
                not accurately reflect true cost until the project has made some progress.
        |

Fraction Perceived to Be Complete  = A FUNCTION OF( Initial Work to Do) ~~|
Fraction Perceived to Be Complete=
        Work Believed to Be Done/Initial Work to Do
        ~       Fraction
        ~               |

Initial Work to Do=
        100
        ~       Tasks
        ~               |

Mimimum Time to Finish Work=
        1
        ~       Month
        ~       When the project is experiencing a schedule overrun, the minimum time to \
                finish the remaining work specifies, for staffing purposes, over what time \
                duration managment would like to complete that remaining work.
        |
Productivity  = A FUNCTION OF( ) ~~|
Productivity=
        Normal Productivity*Effect of Experience on Productivity
        ~       Task / (Person * Month)
        ~               |

Scheduled Completion Date=
        30
        ~       Month
        ~               |

Staff Level Required=
        Estimated Cost to Complete/Time Remaining
        ~       People
        ~       How many people are required to complete the remaining work (estimated \
                cost to complete is person-months of effort) in the time remaining.
        |

"Table for Weight on Progress-Based Estimate"(
        [(0,0)-(1,1)],(0,0),(0.1,0),(0.2,0),(0.3,0.1),(0.4,0.25),(0.5,0.5),(0.6,0.75),(0.7,0.9\
                ),(0.8,1),(0.9,1),(1,1))
        ~       Fraction
        ~       Early in the project, management uses the original budget to estimate \
                effort remaining.  Therefore, the weight on estimates indicated by real \
                progess is 0.  As progress on the project occurs (fraction complete \
                increases), management progressively increases the weight applied to \
                effort remaining as given by tasks remaining and perceived productivity.
        |

Time Remaining=
        Max(Mimimum Time to Finish Work,Scheduled Completion Date-Time)
        ~       Month
        ~       Time remaining to finish the work given scheduled completion date.  As \
                long as simulated time is less than scheduled completion time, time \
                remaining is simply scheduled completion date minus current simulated \
                time.  However, if the project overruns, time can exceed the schedule and \
                time remaining would be negative.  In this situation, management strives \
                to finish all remaining work in a small minimum time.
        |

"Weight on Progress-Based Estimate"=
        "Table for Weight on Progress-Based Estimate"(Fraction Perceived to Be Complete)
        ~       Fraction
        ~               |
Work to Do  = A FUNCTION OF( -Initial Work to Do) ~~|
Work to Do= INTEG (
        Rework Discovery-Rework Generation-Work Accomplishment,
                Initial Work to Do)
        ~       Task
        ~               |
```