Jason C. Ho
214-41-9077
12/20/1999

USC CS599
Term Report

Xerox SPI Model Study

Chapter

1

Introduction

USC CS599 course

As a continuous effort of offering best study topics for graduate students. The University of Southern California's (USC) Center for Software Engineering has created a new course, CS599: Software Process Modeling, for the semester of Fall 1999. Dr. Ray Madachy is responsible for the course, with support from Dr. Barry Boehm, Director of Center for Software Engineering.

The course overviewed the field of software process modeling, and address current research issues with student simulation projects. Process modeling techniques for both continuous systems and discrete systems were covered, with a concentration in system dynamics modeling (continuous systems). The course is designed for students and software engineering professionals who are interested in understanding the dynamics of software development and assessing process strategies. Examples of process and project dynamics covered are Rapid Application Development (RAD), the effects of schedule pressure, experience, work methods such as reviews and quality assurance activities, task underestimation, bureaucratic delays, demotivating events, process concurrence, other socio-technical phenomena and the feedback therein. These complexes and interacting process effects are modeled with system dynamics using continuous quantities interconnected in loops of information feedback and circular causality. Knowledge of the interrelated technical and social factors coupled with simulation tools can provide a means for software process improvement.

The objectives of the course are to:

- Review the field of software process modeling.
- Develop simulation term projects that address critical software issues.
- Describe the systems thinking paradigm for developing increasingly deep understandings of software process structures.
- Show basic building blocks and model infrastructures for software development processes.
- Provide sufficient introductory material including exercises and executable models on the Internet.
- Describe the modeling process, including calibration of models to software metrics data.

- Show recent developments on how others have used the principles of system dynamics to analyze and improve their processes.

- Provide detail of critical implementation issues and future research motivations.

The course project for the semester is a simulation study. Each student needs to define his/her research topic to be addressed by software process simulation.

Problem statement / Purpose of study

Since the author is a full time employee of Xerox Corporation with a software development group, Common Client & Server Development (CCSD), working on its Software Process Improvement based on SEI's Software CMM model. The course project picked is to research and produce a system dynamics model for Software Process Improvement based on the CMM model. With a system boundary based on the scenario of a Xerox S/W development group working from just assessed as a Level 2 organization moving towards achieving Level 3 under CMM model.

Purpose of model building

Reuse or build a model to help provides insights into complex process behavior. And support planning, tracking and prediction of software process improvement activities for the Xerox group.

- Help evaluate different approaches

Chapter 2

Background

About System Dynamics

System dynamics was originally created to help better understand and control industrial systems. System dynamics is used to model problems, to address problems being experienced by any system that changes over time.

The "iThink Analyst" product by High Performance Systems was chosen as the tool to use for crating system dynamic models for the class.

The following quote from "Radical Improvements Require Radical Actions: Simulating a High-Maturity Software Organization" by Steven Burke, published in June 1997. Further explain Systems Dynamics and iThink:

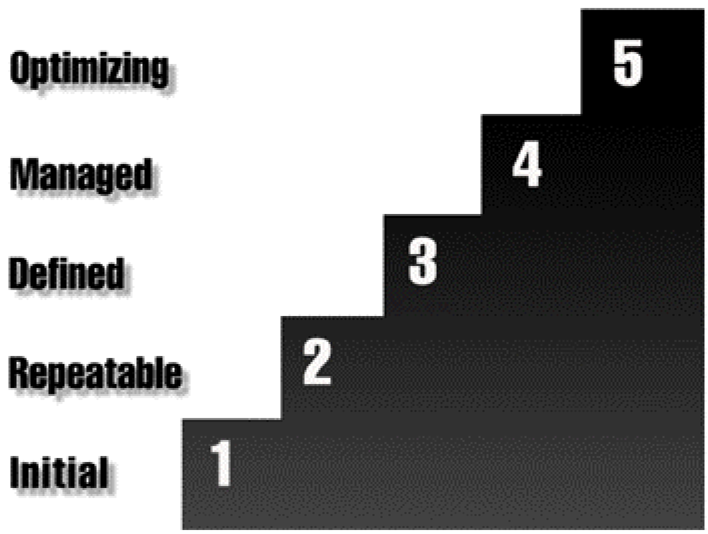
> The following description of Systems Dynamics and iThink is taken from the iThink reference manuals (HPS 1994). Systems Dynamics and iThink (the Systems Dynamics tool used in this study) are concerned with the dynamics, or the change of state over time, of closed loop processes. Systems Dynamics and iThink base their fundamental, process-simulation building blocks on the following definition of process: "A process is a sequence of activities through which material flows for purposes of undergoing some sort of transformation which adds value." The essence of a process is flow. Therefore, a key question when creating an accurate simulation is, "What is flowing?" Over time, it has been observed that material does not flow much of the time. It is often waiting for some activity to be performed on it. Systems Dynamics and iThink, therefore, make a distinction between stocks (amount of resources) and flows (the rate of change of the stock). Flows directly control the transfer, consumption, and transformation of stocks and, therefore, directly affect the level or amount, of stock.
>
> Stocks and flows are two of the three building blocks used by iThink to represent processes. The third building block, information feedback links, provides indirect control of the stocks and flows over time. Feedback is information, not the actual flow of resources, that is provided to the decision rules that control the amount of flow over time. To illustrate this concept, take cutting a bagel as an example. The bagel is the stock being transformed. Cutting the bagel with a knife at a certain rate (e.g., one inch per second) is the flow. Feedback is the form of information from your eyes to your brain that the bagel is cut in two. This information is used to affect the decision rule that controls the cutting rate. Without this feedback, you may cut off your finger.
>
> People try to improve processes by using a static process map or diagram to analyze the resources (stocks) and activities (flows) in isolation. Their goal is to see which ones can be removed or rearranged to speed up the flows. Removing stocks and flows that do not add value is a typical example of such a process improvement methodology. Systems Dynamics and iThink help improve this methodology and help prevent tampering with a process by giving proper focus to the feedback and decision rules elements of a process. It has been found that people are able to analyze the static structure of a process fairly well, but they have difficulty analyzing the dynamics of a process. That is why computer

simulation can be effective. The state of a process can be defined as the amount of stock at any given point in time.

About CMM

The Capability Maturity Model for Software (CMM or SW-CMM) is a model for judging the maturity of the software processes of an organization and for identifying the key practices that are required to increase the maturity of these processes.

The Software CMM has become a de facto standard for assessing and improving software processes. Through the SW-CMM, the SEI and community have put in place an effective means for modeling, defining, and measuring the maturity of the processes used by software professionals.



The Software Engineering Institute (SEI) is a federally funded research and development center established in 1984 by the U.S. Department of Defense with a broad charter to address the transition of software engineering technology. The SEI is an integral component of Carnegie Mellon University and is sponsored by the Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics [OUSD (AT&L)].

Approach of the study

Given the complexity of the CMM model of SPI (Software Process Improvement), and with the limited resource (one person in one semester time frame). The approach was not to create from scratch a CMM SPI model (which might be worthy of a Ph.D. study topic), but to look for an existing SPI model for adoption and reuse.

Limited choice out there of existing Systems Dynamics models on SPI. The Burke's model, from his "Radical Improvements Require Radical Actions: Simulating a High-

Maturity Software Organization” technical report (CMU/SEI-96-TR-024) was chosen for study, because of its close relationship of a SPI model to the CMM concept.

Chapter 3

Model Development

Modeling process

Burke's study as stated in his technical report, starting with the original goal of: find the quantitative value of improving from CMM Level 3 to Level 5. However, since the organization chosen by Burke for study, NASA Goddard Space Flight Center (GSFC) Software Engineering Lab (SEL), uses a different process improvement mechanism than the CMM, which it calls the "Experience Factory." Burke then redefined the new goal: Determining the impact of various management-level policy decisions were discovered and then modeled. The final model produced was not conforming to the CMM model. Quote from Burke's report:

> The original intent of the study was to determine the value of high CMM maturity. The SEL, however, did not pattern their Experience Factory directly after the CMM. Therefore, this study shows the value of high maturity in general, the value of the Experience Factory in particular, and how the Experience Factory can relate to the CMM in a very generalized sense.

Since the approach of the study is to reuse Burke's model to satisfy the purpose of helping Xerox 's software group does CMM SPI activities. The challenges are to resolve the following issues in adopting Burke's model for this study:

- "Experience Factory" vs. CMM
- Simulate high-maturity then working backward vs. Simulate lower-maturity then working forward
- Two types of SPIs : major & minor vs. KPAs in CMM

Chapter 4

Model Description

Burke's study and model

The following causal loop description is the basis of the simulation described in Burke's report. It represents the software development and process improvement processes used by the NASA Goddard Space Flight Center (GSFC) Software Engineering Lab (SEL):

1) Major software process improvement (SPI) efforts are piloted and deployed based on project cycle time (i.e., pilot on one project, tailor and deploy on the subsequent project).

2) Major SPIs increase maturity (the probability of successfully achieving your goals).

3) Increased maturity attracts new hires and retains experienced staff that are "pro SPI" (i.e., they support and participate in SPI activities and are attracted to success and innovation).

4) Pro-SPI staff make minor SPI suggestions.

5) Major and minor SPIs decrease cycle time.

6) Decreased cycle time enables more major and minor SPIs to be accomplished.

7) Go back to 1 and repeat the cycle.

Figure 1 shows a high-level diagram of Burke's simulation and the key feedback relationships.

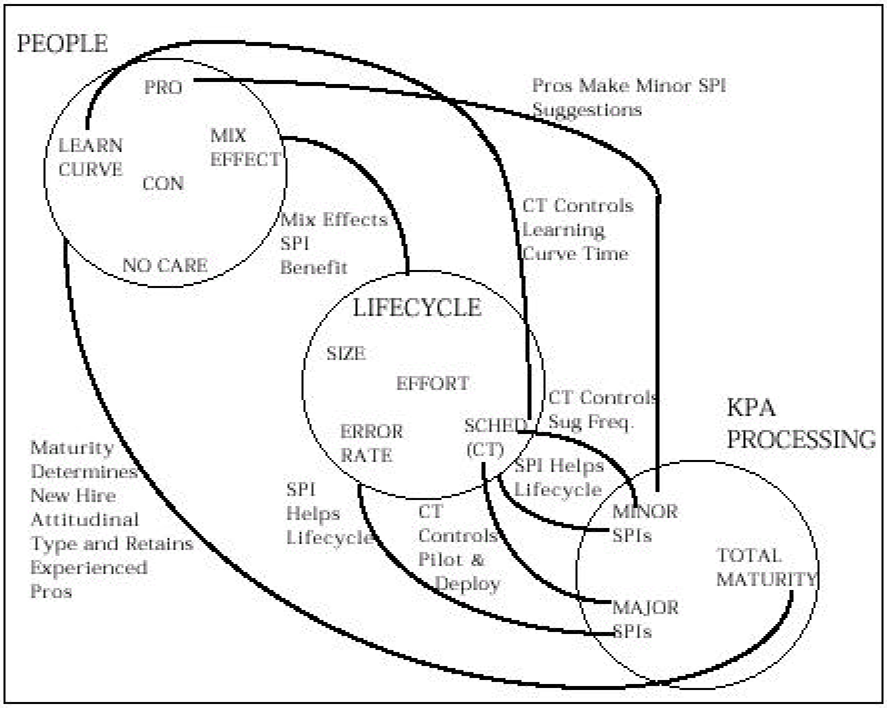


Figure 1. High-Level Diagram of SEL Feedback Relationships

Life-Cycle process

Burke's model life-cycle process modeled how software size, effort, quality, and schedule related to each other in order to produce a product. They could be changed by process improvements or by each other as one process improvement rippled through the four attributes.

SPI benefits are modeled as percent reductions in size, effort, error rate or schedule.

The people process

Burke's model simulated three attitudes of project staff that affected the potential benefit of process improvement:: pro-SPI people, con-SPI people, and no-care people.

The attitudinal mix and the pro/con ratio can affect the overall potential benefit realized by a SPI effort. This was defined as the attitude impact.

It also assumed that it takes one project cycle for a new hire to become considered experienced. That is, the project cycle time is the learning curve.

Processing of maturity Level 4 and 5 KPAs

Modeled the timing of the flow of process improvements into the life-cycle and people subsystems.

There are two types of SPIs in Burke's model: major and minor.(This maturity variable does not directly map to the five levels of the CMM, but uses adoption of SPIs and suggestions as surrogates for maturity levels.)

The total maturity was defined as being a composite of major SPI maturity and minor SPI maturity. Each of these was set on a scale of zero to one.

For more detail description of Burke's study and model, please read Burke's "Radical Improvements Require Radical Actions: Simulating a High-Maturity Software Organization" technical report (CMU/SEI-96-TR-024).

Model modifications

In order to reuse Burke's model for simulating Xerox software group's CMM SPI activities, the following assumptions and modifications were made:

Major assumptions

- Burke's model basic structure can be adopted as reflecting a general software development organization's product life-cycle and process improvement behaviors. Though it was specifically simulating and using NASA GSFS SEL's data.
- Xerox software organization data, later on can be used in Life-Cycle and The People processes with calibration to replace SEL data used in Burke's model.
- The definition of major SPIs in Burke's model can be equated to KPAs in CMM model.

Life-Cycle process

Currently there are no data available from Xerox for recalibrate the percentage (reduced) effects for counting the benefit of CMM KPAs to project life-cycle elements (size, effort, error rate and schedule) in the model. For now, NASA data (major SPI benefits) used in Burke's model will be adopted for reference. Therefore, the results produced now based on NASA data will be for reference only to Xerox organization. Later on when Xerox data are available, the model can be then recalibrate for using Xerox data.

The people process

Burke's model on simulating the people effect reflects some general phenomenon experienced by software development organization undergoing the SPI activities, evidenced by my own personal experience with the Xerox software development group.

> The SEL said that there were three attitudes of staff that affected the potential benefit of process improvement: pro-SPI people, con-SPI people, and no-care people. Pros made almost all of the minor SPI suggestions. Based on a subjective survey conducted on an Ada/reuse major SPI effort, about 30% of the staff said they were pro-Ada, 20% were con-Ada, and 50% did not care. Both the pros and the cons were vocal in their opinions about this major improvement effort.
>
> The attitudinal mix and the pro/con ratio can affect the overall potential benefit realized by a SPI effort. This was defined as the attitude impact. If there are more pros than cons, then more no-cares will also

adopt the effort. This higher penetration and adoption will realize a higher overall benefit. If there are a lot of cons, then a lot of people may not adopt the SPI effort. Interviewees agreed that attitude affected SPI adoption and that staff members with strong attitudes affected other staff members' adoption of SPI.

In System Dynamics, finding the correct relationships among variables is more important than waiting for extensive empirically validated data. This is because the simulation can run a wide range of values to see what values are critical and if the overall results are insensitive to the particular values. The following "soft" variable relationship between pro/con ratio and personality mix effectiveness was derived using 30%, 20%, and 50% values:

1. At a 30/20 ratio, assume the 50% no-cares go along with the pros.

2. So 80% of the total staff are virtual pros. Since this was the documented case, we set their personality mix effectiveness to 1. That is, if a SPI effort claims to give a benefit of 50%, then the personality mix effectiveness given this personality mix is 50% times 1, or 50%.

3. If we want to find the effectiveness for a 25/25 pro/con ratio, then we can use the following: Assume the no-cares are also split since the pro/con ratio was even. If 80% virtual pros produced an effectiveness of 1, then having 50% virtual pros produces an effectiveness of 50%/80% = 0.625.

4. If we have a 20/30 pro/con ratio, assume the no-cares are virtual cons (poisoned). So having only 20% pros produces an effectiveness of 20/80 = 0.25.

Quick polls over the Xerox software development group under study, the nominal case of a 30/20 pro-con ratio and 50% no-cares, seems to told the truth for the organization.

Also in Burke's report the SEL reported that turnover (quits) started at about 10% per year, then decreased to 2-3% as maturity increased, then increased to about 6% toward the end of the 10-year period. For the past five years, the turnover rate at the Xerox software development group under study, has been flat at 5% year by year regardless of process maturity. In the test run, "quit rate" will be modified to reflect Xerox's situation.

KPA process

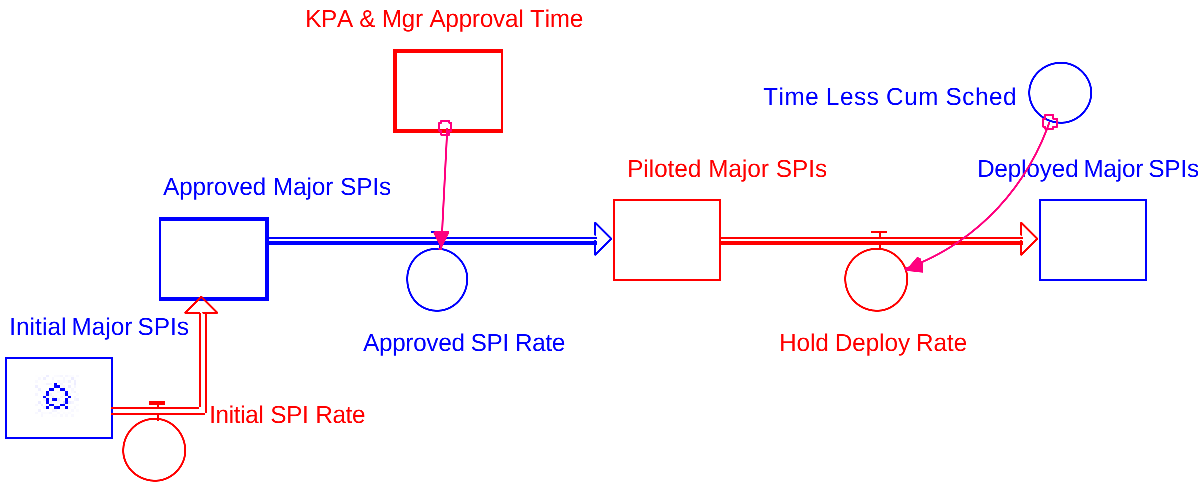
This area is the major focus of the study, in order to fulfill the purpose of helping Xerox CCSD decide on its commitment and approach of moving from CMM Level 2 to Level 3. Given one of the major assumptions: the definition of major SPIs in Burke's model can be equated to KPAs in CMM model.

So reusing "major SPI" concept in Burke's model to simulate CMM's KPAs. Then from now on major SPI and CMM KPA can be used interchangeably in the model. There are 7 major SPIs (7 level 3 KPAs under CMM model) has to be piloted and deployed by projects, in order to reach the next level of maturity. Based on CMM assessment criteria, all KPAs must be implemented (piloted), plus some need to be repeated (deployed) to satisfy the institutionalization requirements. In which the pilot/deploy scenario simulated in Burke's model can be adopted to reflect the CMM assessment requirements.

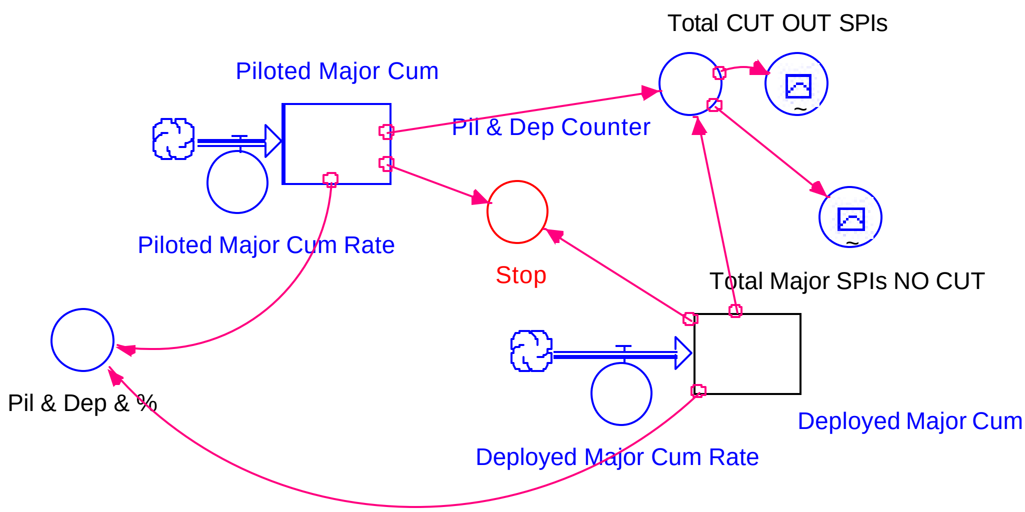
In Burke's model, it assumes that per cycle time (project) can only pilot or deploy one signal major SPI. Which in CMM's case, it will take too long to implement all 7 KPAs with the original model (see test run results in Chapter 6). Validated by Xerox CCSD's experience on achieving CMM Level 2, on average, 2 KPAs can be implemented per project.

To simulate average 2 KPAs roll out per project, the rate of the following elements (as shown in the picture) in the model, has been changed from 1 to 2 for the nominal case of Xerox CCSD:

- Initial SPI Rate
- KPA & Mgr Approval Time
- Piloted Major SPIs
- Hold Deploy Rate



Also in order to capture the simulated lapsed time in implementation 7 level 3 KPAs, an element needs to be added to stop the model running after 7 major SPI roll out. In this particular Xerox CCSD case, the already implemented Level 3 activities is assessed as worth of 1 full KPA. So the modification is to stop the model after 6 major SPIs have been piloted. The following picture shows the element added.



Concept of Minor SPIs in Burke's model has little to do with reaching the next level of CMM maturity.

> Minor SPIs were generated either by the pro-SPI people within one of the five projects (intra-project SPIs) or received as a lesson from one of the outside four projects (inter-project SPIs).
>
> …
>
> Minor SPI maturity was different. The basic premise was that an organization was not considered mature until the project staff actively participated in process improvement. That is, an external staff, such as the Experience Factory staff, can impose a major SPI on a resistant project staff. But if the project staff does not participate in continuous process improvement, the organization should not be considered mature. It was decided that when the frequency of making minor SPI suggestions approaches one per week for a sustained period of half a project cycle, then the minor SPI maturity reaches its highest value (of 1). These decisions may seem arbitrary, but their content represents the spirit of the CMM. The spirit of the CMM is to have all of the people participate in technological and process improvement as the best means to successfully achieve your goals. This maturity variable does not directly map to the five levels of the CMM, but uses adoption of SPIs and suggestions as surrogates for maturity levels.

However, it does simulate in general, benefits of process improvement on size, effort, quality and schedule. Which will be preserved until further research can be done to more correlates it to the CMM model.

One of the major issue faced by Xerox CCSD group, while taking on the CMM SPI journey, was that there isn't a group of full time CMM process people to help on process activities. So the group has to use the same resource (people who do software development) to work on process related activities, such as define organizational process and plan for the roll out, etc. In a general sense, it will prolong the process journey in reaching the next level of maturity on the CMM ladder. Moreover, in many cases, it is not the general approach taken by other major companies working on their CMM SPI.

Fortunately enough there is a similar condition being simulated in Burke's model, it's called Cut-out vs. Non-Cut-Out cases.

> "Cut-out" versus non "cut-out" cases. In the cut-out cases, projects accomplish SPI activities in isolation with no sharing of lessons. The non cut-out case has a fully staffed group that would accomplish organization-level SPI activities across projects using the CMM Maturity Level 4 and 5 key process area activities as a guide to pilot and deploy major SPIs across projects, extract lessons from projects, communicate them to other projects, and staff an organization-level metrics program. The SEL calls this staff the "Experience Factory" staff. The subsequent tables of run results will have "CUT OUT" labeled on the appropriate runs.

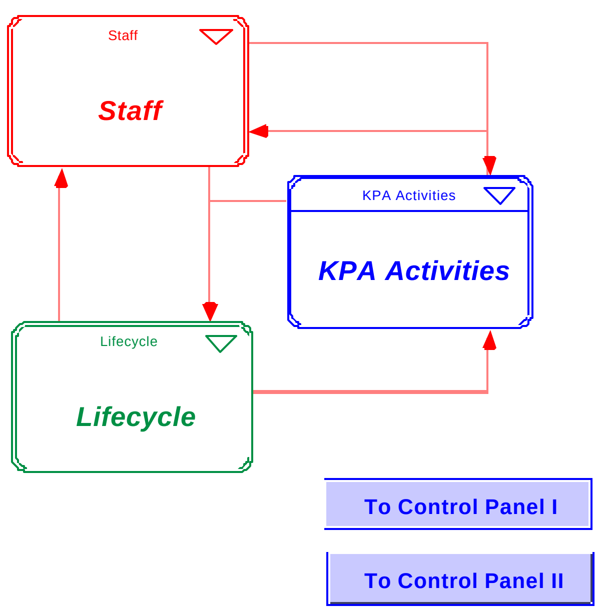
So in the case of Xerox CCSD group, it is a similar situation of Cut Out case, which will be nominal case for the modified model. In addition, the comparison with the Non Cut Out case would warrant the attention of management for the approach on SPI.

User interaction

The original Burke's model didn't build a user interface for changing parameters and display results, most likely because of an earlier version of iThink software tool, without much support on that area.

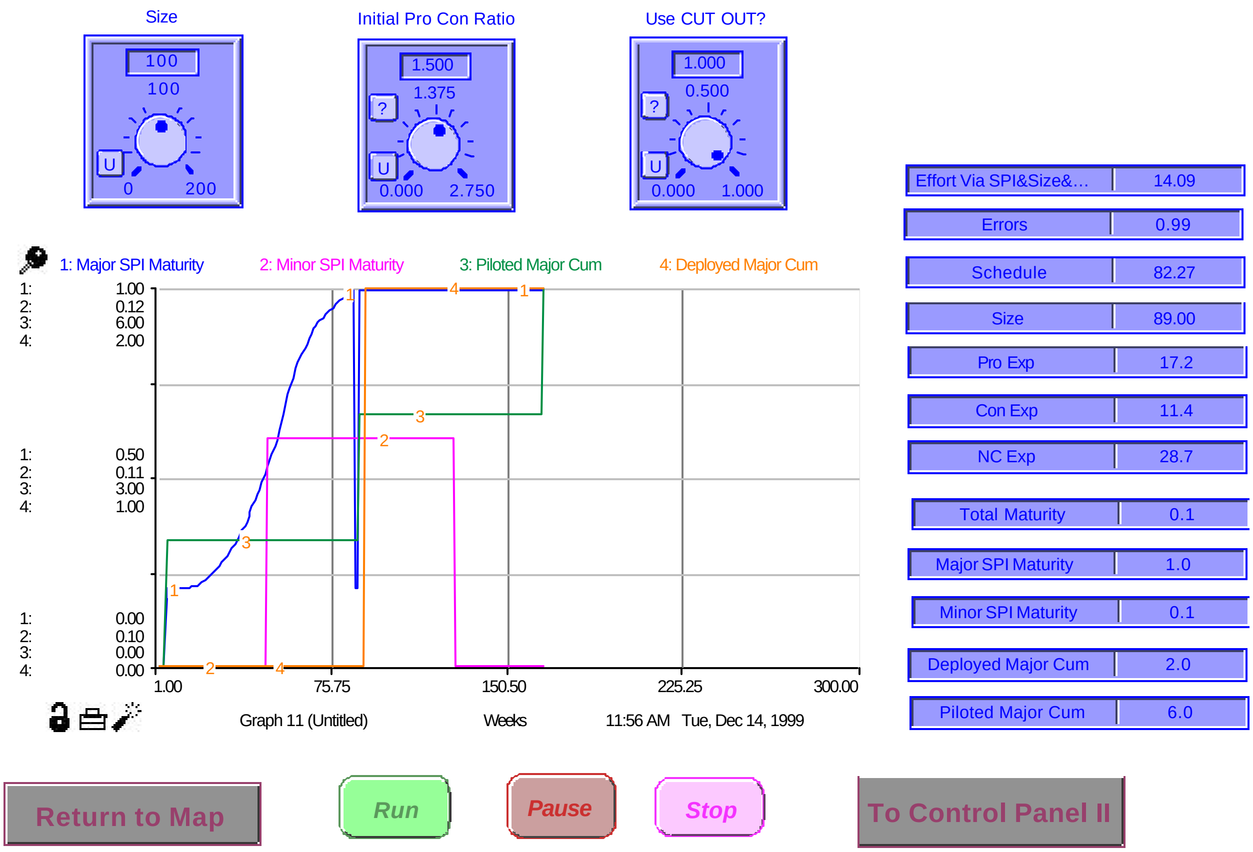
Part of the effort of this study is to build some user interaction for the model. Followings are what been added:

Model High Level Diagram



Using the Process Frame building block from iThink tool created the high-level diagram. This High-level mapping layer shows the concentration of the most important aspects of the system. Notice the arrows linking the Process Frames in the high-level diagram. These show the relationships between the various pieces of the diagram.

Control Panel I



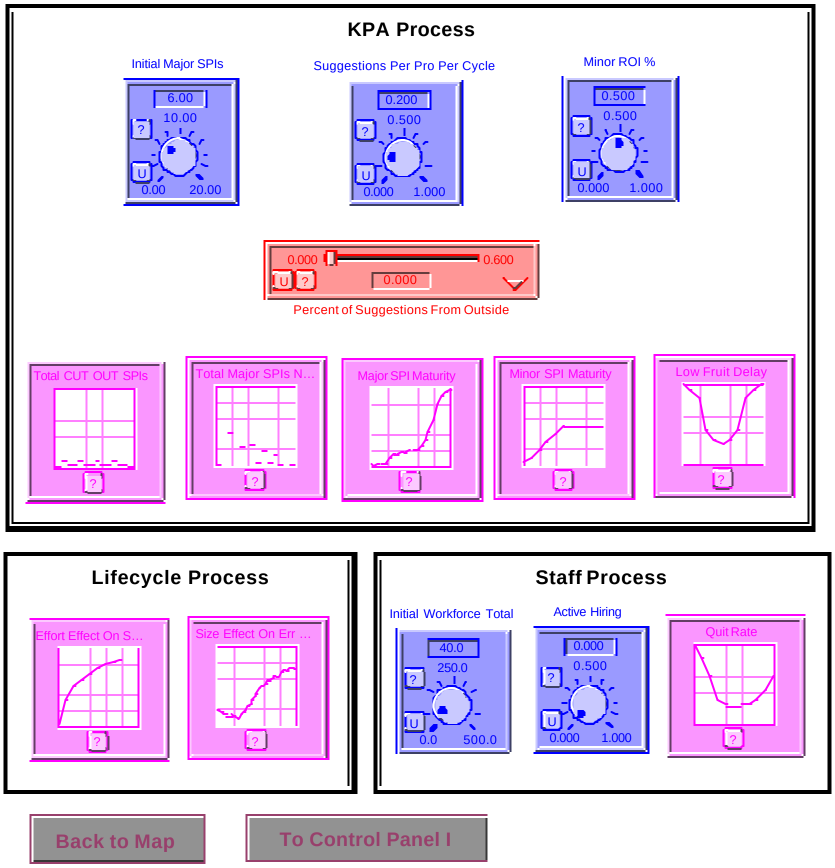
From the high-level diagram, click once on the button titled “To Control Panel I” and the screen will look like the above picture. This is the “dashboard” or “cockpit” from which to perform some “Flight Simulator” like test runs. A special feature of the iThink tool.

This control panel is the major user interaction area for this particular study, which shows the following parameters available for changes:

- Size of the project in KSLOC, using a Knob input device.
- Initial organization pro/con ratio, using a Knob input device.
- Cut Out or Non Cut Out, using a Knob input device.

It also shows a Sketchable Graph output on displaying cumulative major SPI being piloted or deployed as the model run in real time. As well as various Numeric Display bars on interesting outputs from the model, such as improved size and schedule numbers and many others.

Control Panel II



This particular control panel was created for grouping all the various input parameters being used by different sections of the model together. As a reminder and convenience for future research and expansion on the model.

As well as easy adjustment of the parameters for the test runs.

Chapter

5

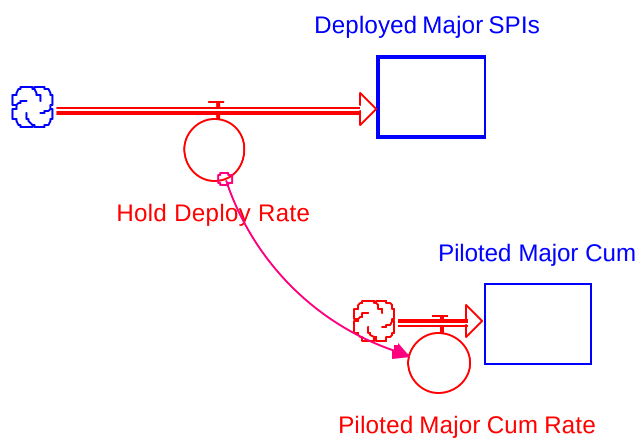
Model Verification and Validation

Sensitivity Analysis

Sensitivity analysis is used to determine how “sensitive” a model is to changes in the value of the parameters of the model and to changes in the structure of the model. Parameter sensitivity is usually performed as a series of tests in which the modeler sets different parameter values to see how a change in the parameter causes a change in the dynamic behavior of the stocks. If the tests reveal that the model is insensitive, then it maybe possible to use an estimate rather than a value with greater precision. Sensitive analysis can also indicate which parameter values are reasonable to use in the model.

Since the Burke’s model has been thoroughly verified using the NASA GSFC SEL data. With the modification made in this study, the purpose of a sensitive analysis is to find a proper pattern of time progress on achieving certain number of major SPIs (KPAs). By using the existing Xerox CCSD data of, it took 4 years to move from Level 1 to Level 2 (accomplishing 6 KPAs), as validation.

The parameter “Hold Deploy Rate” which determines “Piloted Major Cum Rate”, is the factor that controls the stock “Piloted Major Cum”. As depicted in the following picture.

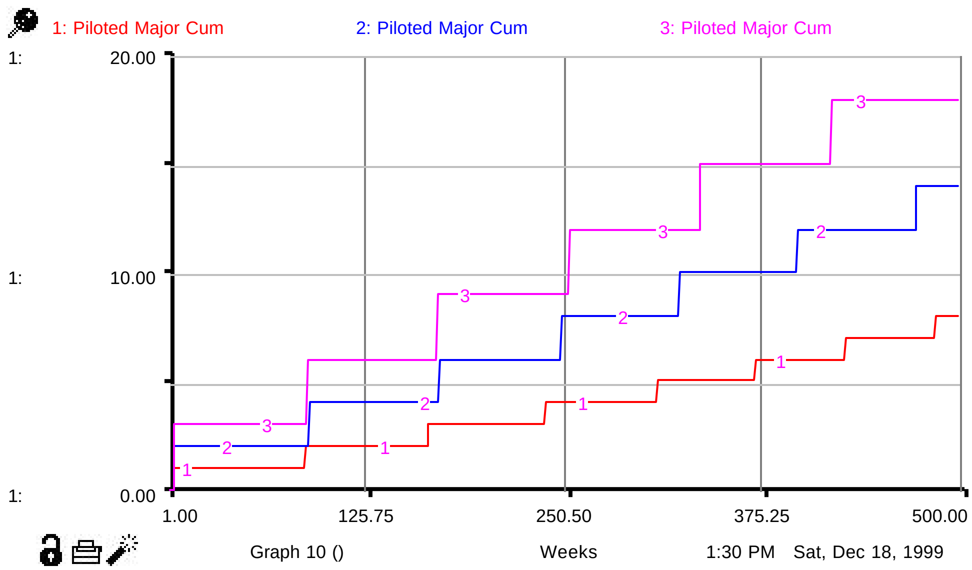


Our goal here is to find the right pilot major SPI deploy rate, so it will reflect the Xerox CCSD’s KPA implementation rate. Here the definition of the nominal case is with the following related parameters set at the values specified below:

- Size, the size of the project is set at 100 KSLOC.
- Cut Out or Non Cut Out, set at Cut Out case.

- Initial Pro/Con ratio set at 1.5 (represent 30% Pro SPI people, 20% Con SPI and 50% No Care)

We can try to use several values for “Hold Deploy Rate” to see how the stock of “Piloted Major Cum” changes over time. Three values that seem reasonable are 1, 2, and 3. The comparative run is shown in the figure below.



The three curves all show the same pattern of step increase. The value we are looking for is 6 KPA at about 208 weeks (4 years). Curve 1 shows 6 KPA happens at between 375 to 500 weeks, which is too slow. Curve 3 shows 6 KPA happens at between 1 to 125 weeks, which is too fast. While curve 2 shows it happen between 125 to 250 weeks, this is the reasonable range. So the pilot major SPI deploy rate of 2 is the reasonable value to use in the model.

Chapter

6

Model Application and Transition

Test runs with the original Burke's model

First started with the original Burke's model, to see how it would work given the Xerox CCSD group's condition.

Test Run 1

Based on Burke's baseline case for SEL, the values for the related variables were set at: pro/con ratio at 1.5, project size at 100 KSLOC, and Non Cut Out. Modified with Xerox CCSD's condition, changed initial task force to 40 people and changed Non Cut Out to Cut Out case. Test run showed the group would not reach next level of maturity in 10 years (500 weeks) time spec.

Test Run 2

Based on the result from the above test run 1, changed the initial task force to 100 people (if the management decided to grow the group because of extra process burden added to the group). Test run showed the group will reach the next level of maturity in 430 weeks (about 8.3 years, at least it's in the foreseeable future).

Test runs with the modified model

After modify the model, with initial task force set at 40 (group size of Xerox CCSD), test run with the following related input variables: Project size, Initial pro/con ratio, and Cut Out or Non Cut Out.

Test Run 1

With the nominal case for Xerox CCSD, values for the above parameters were set at, 100 KSLCO, 1.5, and Cut Out. Test results show 172 weeks (about 3 years) for accomplishing 6 major SPIs (KPAs), with the benefit of final size at 89 KSLOC. This represents the major input of this study, that the Xerox CCSD group should make their commitment to the management of accomplishing CMM Level 3 in at least the next 3 years.

Test Run 2

With all the nominal case variable values, only changed from Cut Out case to Non Cut Out. Test results show 121 weeks for accomplishing 6 major SPIs, and with significant benefit on the final size and schedule. This shows with different approach, with a support from a full time process group, would have saved the journey a full year to get to Level 3.

Test Run 3

With all the nominal case variable values, only changed variable value of Size from 100 KSLOC to 200 KSLOC. Test results show 292 weeks for accomplishing 6 major SPIs. This shows that with a bigger project size, it will take longer to finish the project with the same amount of resource available. In terms, it will take longer to pilot out all the KPAs.

Test Run 4

With all the nominal case variable values, only changed variable value of Size from 100 KSLOC to 50 KSLOC. Test results show 134 weeks for accomplishing 6 major SPIs. This shows that compare to Test Run 3 with a smaller project size, it will be faster to finish the project. In terms, it will shorten the time to pilot out all the KPAs. So it make sense to select some small projects for piloting the CMM KPAs, which was one of the good approach experienced by Xerox CCSD while working form Level 1 to Level 2.

Test Run 5

Rerun Test Run 3 with the Non Cut Out case.

Test Run 6

Rerun Test Run 4 with the Non Cut Out case.

The following table summarize all the Test Run results

	Input Variables			Output Values				
Test Run	**Size**	**Pro/Con**	**Cut/Non Cut**	**CMM SCHD**	**EFF End**	**ERR End**	**SCHD End**	**Size End**
1	100	1.5	Cut	172	14	1	82	89
2	100	1.5	Non Cut	121	6.5	1	46	56
3	200	1.5	Cut	292	42	3.7	138	178
4	50	1.5	Cut	134	9	1.3	63	45
5	200	1.5	Non Cut	180	12.8	1	78	112
6	50	1.5	Non Cut	88	4	1	29	28

Chapter 7

Conclusions and Recommendations

Conclusions

This study borrowing the concept and model from Burke's technical report, though the process improvement paradigm is the NASA SEL models of "Experience Factory" not the CMM paradigm. As Burke mentioned in his report:

> The original intent of the study was to determine the value of high CMM maturity. The SEL, however, did not pattern their Experience Factory directly after the CMM. Therefore, this study shows the value of high maturity in general, the value of the Experience Factory in particular, and how the Experience Factory can relate to the CMM in a very generalized sense.

The test run results with the modified model, however, does satisfy the Xerox CCSD need in helping the group make the proper schedule commitment to the management of moving from CMM Level 2 to Level 3, in its unique situation.

Given the increased interest in the CMM model in recent years, as a paradigm for process improvement. Software organizations are looking for answers in questions regarding the value of improving software process maturity. These include: What is the return on investment from CMM-based software process improvement? What have been the lessons learned by organizations that have improved? And how long does it take to improve? With the advance research in systems dynamics modeling techniques, maybe a comprehensive systems dynamics model of CMM can be developed some day to help answer those questions.

Future works

A comprehensive systems dynamics model of CMM is the ultimate goal, with an unachievable schedule and resource from the author. However, it is the author's intention to keep on working with the modified Burke's model in the following areas. Also to keep on recalibrate the model as Xerox CCSD started to collect more data on its Level 3 efforts, to make the model more adaptable to Xerox CCSD.

- Recalibrate the KPA benefit percentage effects when Xerox data available
- Modify "Effort Effect on Schedule" with Xerox data. Currently NASA data based on the equation: .153*Size+5.3*Errors-4.689
- Study how to apply minor SPIs in Burke's model towards contribution to reaching the next level of CMM

Chapter 8

Appendices and References

Appendix A: Burke's model description and design

The next three pages show the actual simulation coded in the iThink Systems Dynamics application. "Underneath" each box (stock) or circle (flow or converter) are equations and logic that control the execution of the simulation. A quick low-level design description of each diagram follows:

1. The life-cycle subsystem. Starting at the upper left hand corner and working clockwise, the personality (attitude) mix converter adjusts any of the schedule, size, etc., ROI rates based on the Pro to Con ratio. Moving over to the top center, cumulative size SPIs affect the rate of change of size. In the upper right corner, size changes affect error rate according to the graph in Figure 2. Also, a SPI may affect error rate directly. In the lower right corner, the impact that size or error rate might have on effort is modeled. In the lower center part of the diagram, changes in effort due to size or error rate changes are combined with changes in effort due to effort SPIs to model overall effort changes. Finally, in the lower left corner, effort changes affect schedule according to Figure 3. Also, schedule SPIs may also affect schedule directly.

2. The people (staff) subsystem. Starting at the right, experienced no-cares, cons and pros are separate flows that quit according to the quit rate. The quit rate itself changes with maturity. For each quit, a no-care, con or pro new hire is determined by a Montecarlo probability distribution that is also based on maturity (see the left part of the diagram). If an experienced con quits, a pro may replace him or her if the maturity is high. The time the no-cares, cons, or pros spend in training (the middle slotted boxes) is regulated by the current project schedule. As process improvements decrease cycle time, the learning curve time is also decreased. As process improvements increase maturity, the attitudinal mix based on the new pro and con totals also changes because new hires may be of a different attitude than the experienced person who quit.

3. The KPA processing subsystem. This is the most complex subsystem. The top half deals with major SPIs, the bottom half models the minor SPI suggestion process. For the major SPI section starting from the left, major SPIs are piloted and deployed as regulated by schedule (see top center for the schedule box). In the center, the "Pilot Major Cum" and "Deployed Major Cum" receive the impulse of the completed piloted or deployed SPI at the end of a cycle and store the cumulative total. In the center, the "Total Major SPI" sends out the appropriate ROI percent based on when the pilots and deploys are complete. This ROI percent is sent out to one of either the "Size Major ROI," "Effort Major ROI," "Error Major ROI," or "Schedule Major ROI" flows located on the top right corner. For the bottom half that models the minor SPI suggestions, the bottom center shows the "Sug Frequency" being a product of "Total Maturity," number of pros, "Suggestions per Pro per cycle," and "Percent of Suggestions from Outside." The structure to the right of this causes the suggestions to be implemented in a "low-hanging fruit" fashion. That is, within a project

cycle, a few suggestions are made at the beginning, many in the middle, then only a few at the end since all of the easy ones (the low-hanging fruit) were worked. The structure on the lower left corner is needed to compute the average suggestion frequency for a floating period of one half of a project cycle. As the suggestion frequency increases, the “Minor SPI Maturity” also increases because when an organization is mature, everyone is involved in making improvement suggestions.

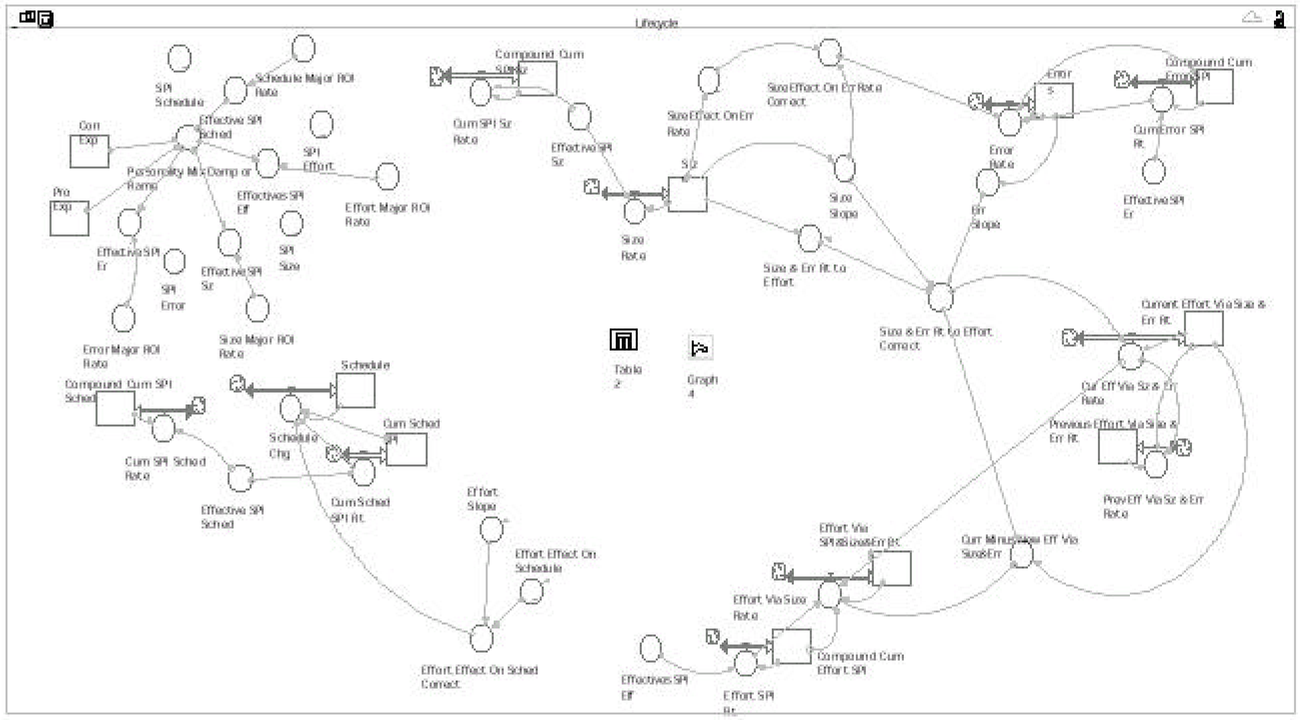


Figure 5. The Life-Cycle Subsystem

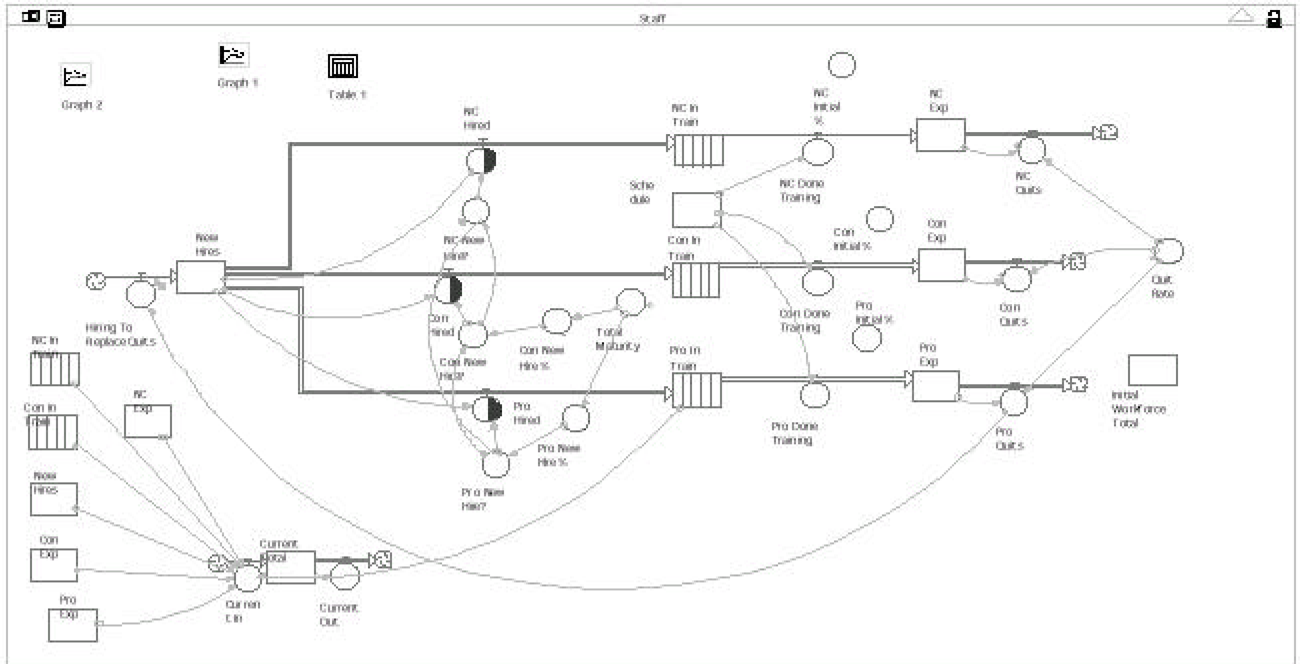


Figure 11: The People (Staff) Subsystem

Figure 6. The People (Staff) Subsystem

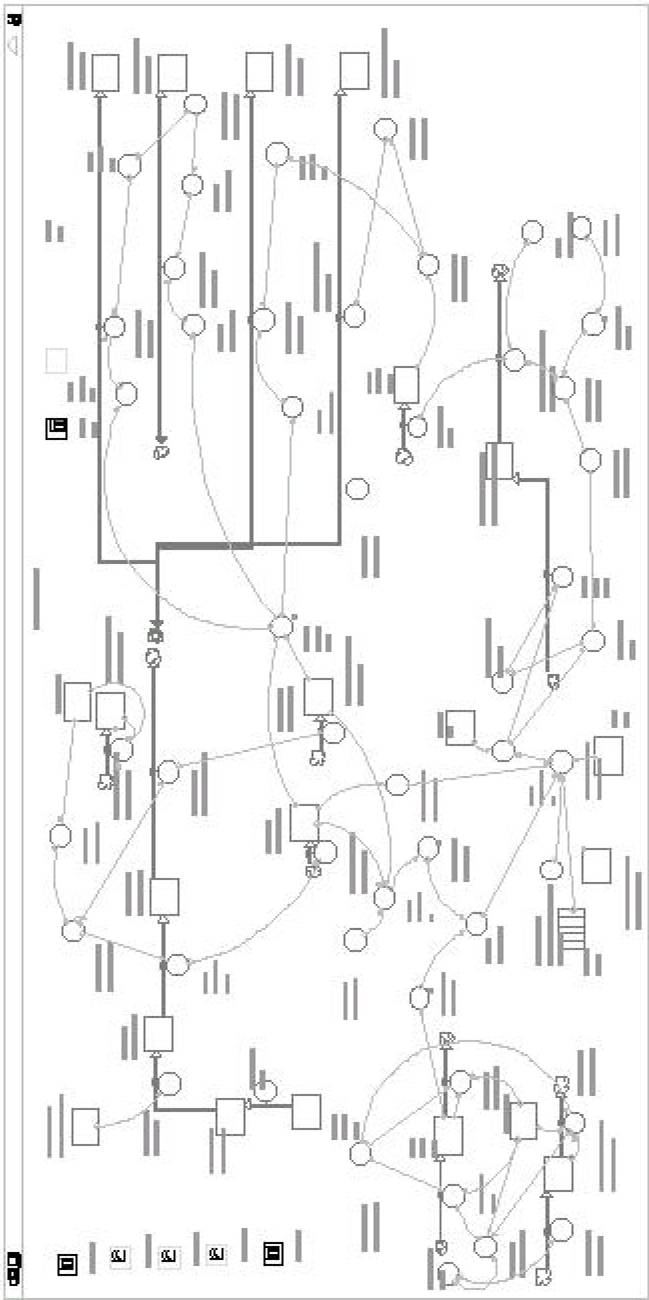


Figure 7. The KPA Processing Subsystem

[redacted]

References

Burke, S. (1997). Radical Improvements Require Radical Actions: Simulating a High-Maturity Software Organization. (CMU/SEI-96-TR-024, ESC-TR-96-024). Resident Affiliate, Computer Sciences Corporation.

Madachy, R. (1996). Process Modeling with Systems Dynamics. 1996 SEPG Conference, Atlantic City, NJ, Software Engineering Institute, Carnegie Mellon University.

HPS (1994). Introduction to Systems Thinking and iThink. iThink Technical Reference Manual. Hanover, NH, High Performace Systems, Inc.

Paulk, M., B. Curtis, et al. Capability Maturity Model for Software, V1.1 (CMU/SEI 93-TR-24, ADA 263403). Pittsburgh, PA, Software Engineering Institute, Carnegie Mellon University.

Herbsleb, J., Carleton, A., Rozum, J., Siegel, J., Zubrow, D. (1994). Benefits of CMM-Based Software Process Improvements: Initial Results. (CMU/SEI-94-TR-013, ESC-TR-94-013) Technical Report.

Goldenson, D. and J. Herbsleb. After the Appraisal: A Systematic Survey of Process Improvement, Its Benefits, and Factors that Influence Success (CMU/SEI-95-TR-09, ADA 302225). Pittsburgh, PA, Software Engineering Institute, Carnegie Mellon University.

Clark, B. (1997). The Effects of Software Process Maturity on Software Development Effort. A Dissertation presented to the Faculty of The Graduate School, University of Southern California.