THESIS

CAUSALITY TRACING USER INTERFACE DESIGN AND DEVELOPMENT FOR A SOFTWARE MANAGEMENT FLIGHT SIMULATOR

by

Luis Hernández

December 1995

Thesis Advisor: Tarek Abdel-Hamid

Approved for public release; distribution is unlimited.
**ABSTRACT (maximum 200 words)**

Interactive simulations are a highly suitable tool for training managers in their increasingly complex roles in software project management. This research effort designs and implements an interactive user-friendly interface for the system dynamics software development and project management model using a flight simulator as a metaphor. Methods and techniques for good user interface development are considered and implemented using the Ventana Simulation (Vensim) application development, modeling, and analysis environment. The resulting interface facilitates the user experimentation with management policy strategies and decision making, as well as the investigation of scenarios to determine what the circumstances were that caused the project's expected behavior to vary. The analysis capabilities of the interface enables the user to trace cause-and-effect relationships that are often invisible and not considered when making management decisions. The interface's causal tracing functionality significantly enhances the value of the underlying model as a learning tool by facilitating the development of an integrated, improved vision of the world that managers are responsible to control.

**SUBJECT TERMS** Simulations, Software Project Management, System Dynamics, User Interfaces,
CAUSALITY TRACING USER INTERFACE
DESIGN AND DEVELOPMENT FOR A
SOFTWARE MANAGEMENT FLIGHT SIMULATOR

Luis Hernández
Major, United States Army
B.S., University of Puerto Rico, 1983

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY
MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
December 1995

Author: Luis Hernández

Approved by: Tarek Abdel-Hamid, Thesis Advisor

Balasubramaniam Ramesh, Associate Advisor

Reuben T. Harris, Chairman, Systems Management
ABSTRACT

Interactive simulations are a highly suitable tool for training managers in their increasingly complex roles in software project management. This research effort designs and implements an interactive user-friendly interface for the system dynamics software development and project management model using a flight simulator as a metaphor. Methods and techniques for good user interface development are considered and implemented using the Ventana Simulation (Vensim) application development, modeling, and analysis environment. The resulting interface facilitates the user experimentation with management policy strategies and decision making, as well as the investigation of scenarios to determine what the circumstances were that caused the project’s expected behavior to vary. The analysis capabilities of the interface enables the user to trace cause-and-effect relationships that are often invisible and not considered when making management decisions. The interface’s causal tracing functionality significantly enhances the value of the underlying model as a learning tool by facilitating the development of an integrated, improved vision of the world that managers are responsible to control.
TABLE OF CONTENTS

I. INTRODUCTION ........................................................................................................... 1
   A. BACKGROUND ........................................................................................................ 1
   B. PURPOSE OF RESEARCH .................................................................................... 3
   C. SCOPE OF RESEARCH ......................................................................................... 3
   D. THESIS ORGANIZATION ..................................................................................... 3

II. RESEARCH BACKGROUND ....................................................................................... 5
   A. SOFTWARE PROJECT MANAGEMENT CURRENT STATUS .......................... 5
   B. MODELS AND SIMULATIONS .............................................................................. 7
   C. SMFS SYSTEM-DYNAMICS MODEL ................................................................. 9

III. VENTANA SIMULATION ENVIRONMENT ................................................................ 13
   A. OVERVIEW ......................................................................................................... 13
   B. CAPABILITIES ..................................................................................................... 13
   C. CAUSAL LOOP DIAGRAMMING AND TRACING ............................................ 15

IV. USER INTERFACE DESIGN CONSIDERATIONS ..................................................... 17
   A. ABOUT USER INTERFACES ............................................................................... 17
   B. DESIGN CONSIDERATIONS .............................................................................. 19
      1. General Design Guidelines ............................................................................ 19
      2. Organization Guidelines .............................................................................. 20

V. INTERFACE ARCHITECTURE ................................................................................ 23
   A. SYSTEM OVERVIEW ........................................................................................ 23
   B. MENU STRUCTURE ............................................................................................ 25
   C. USER’S HANDBOOK ........................................................................................ 27
      1. Introduction ..................................................................................................... 27
      2. System Requirements ................................................................................... 27
      3. Interface Starting Methods ......................................................................... 28
      4. Running the Simulation .............................................................................. 30
         a. System Dynamics Primer ......................................................................... 31
         b. Simulator Model Overview ...................................................................... 32
         c. Play a New Game ...................................................................................... 33
         d. Analyze Previous Run Game ................................................................... 35
         e. Exit .............................................................................................................. 41
      5. Simulation Results Storage ......................................................................... 41

VI. CAUSAL TRACING DEMONSTRATION .................................................................. 43
   A. INTRODUCTION ................................................................................................ 43
   B. INITIAL CONDITIONS ...................................................................................... 43
   C. SIMULATED PROJECT BEHAVIORS ............................................................... 44
      1. Project 1 ......................................................................................................... 45
      2. Project 2 ......................................................................................................... 46
   D. CAUSAL TRACING ............................................................................................ 47
VII. CONCLUSIONS ............................................................................................................. 57

VIII. ACCOMPLISHMENTS ............................................................................................... 59
   A. USER INTERFACE ..................................................................................................... 59
   B. USER'S HANDBOOK ................................................................................................. 60

IX. LESSONS LEARNED ................................................................................................... 61

X. FUTURE DIRECTIONS .................................................................................................. 63

APPENDIX A. SMFS VENSIM MODEL FILE (VMF) ......................................................... 65

APPENDIX B. SMFS VENSIM GRAPH DESCRIPTION (VGD) FILE ................................. 115

APPENDIX C. SMFS CUSTOM DESCRIPTION (VCD) FILE ............................................ 127

APPENDIX D. SMFS VENSIM MODEL EQUATIONS ....................................................... 147

LIST OF REFERENCES ..................................................................................................... 177

INITIAL DISTRIBUTION LIST ......................................................................................... 179
I. INTRODUCTION

A. BACKGROUND

Modern businesses depend on quick delivery and use of complex information in today’s highly competitive electronic economic environment. Technological advances in computer hardware are making computers more powerful and affordable. The tendency is for more companies to migrate their business activities to the electronic commerce environment. This migration has created a strong dependence on software to run business critical applications. Presently, businesses are automating key processes in order to gain a competitive edge, market shares, and merge into the national and global markets.

Key business factors, such as time to market, quality levels, manufacturing costs, and product innovations provide the impetus and motivation for companies to demand new and better software products for their critical applications. Companies require flexible and comprehensive software environments for information systems that can handle their needs as fast as their market evolves. How fast their critical business sectors are automated is critical for their economical survival. Speed in acquiring critical applications drive a tremendous demand for fast design of software, which in turn, exerts great pressure for capable, timely, and responsive software development projects. In addition, as time passes, and the number of business applications continue to grow, the need for upgrades further increases the demand for software maintenance, upgrade, and management. Today’s customers require development of high-quality software products, on time, and at a low cost.

As a consequence, the software development industry, has exhibited a tremendous growth path. In 1993, the number of businesses in the United States that were engaged in software development activities was estimated to be in the vicinity of 32,000 businesses [Jones, 1994, p. 19]. As a Nation, we spend close to 100 billion dollars a year on development, production, and maintenance of computer software [Pressman, 1993, p. 13]. Many companies engage in producing their own software or developing products for other companies. However, industries in the United States are following an alarming
path leading to what many have called the “software crisis” [Pressman, 1993, p. 12]. Many software products consistently fail to meet cost, schedule and performance goals. For these reasons, software development has been considered a risk-prone activity [Jones, 1994, p. 46].

Numerous analytical tools and software engineering techniques, such as Computer- Aided Software Engineering (CASE) technology, have been developed to assist in software development. However, management problems continue to exist [Sprague and McNurlin, 1993, p. 272-278, and U.S. General Accounting Office, 1993]. Managers responsible for software development must have a good understanding of the complex interrelationships that take place within a software project in order to minimize risks and be successful. Many problems associated with software have been caused by the manager’s lack of knowledge of important factors affecting development, identification, and project control.

Managers need to have an integrative view of the process they are trying to manage. They need to be able to study and understand the system as a whole, and not as individual interacting parts. An effective mechanism for learning a complex system and its behavior is through the use of computer modeling and simulation. From a system dynamics’ point of view, the use of computer-based tools should allow the analysis of interactive simulation results, the impact that management decisions have over time, and the causes for inexplicable or counterintuitive results.

The system dynamics model of software development, created by Tarek K. Abdel-Hamid, provides a comprehensive system-dynamics model of the software development and program management worlds. The model simulates the relationship between multiple functions of the software development process and its management functions through the use of feedback or cause-and-effect loops. As a training tool, this model allows for the simulation of a complex environment by providing numerical results. However, it lacked a friendly user interface capable of providing the means to modify and examine the cause-and-effect behavior inherent in the model’s information feedback loops.
B. PURPOSE OF RESEARCH

The objective of this thesis is to design and develop an effective graphical user interface implementing causal tracing. The interface design should result in a tool that facilitates the investigation of information feedback loops by allowing the user to trace back, and examine the cause-and-effect of selected variables during or after a given model simulation. A causal tracing user interface will substantially enhance the value of a system dynamic model as a learning tool, as well as provide the means for further modeling research and the study of the complexities affecting the software development and management worlds.

C. SCOPE OF RESEARCH

The research focuses on the design, development, and documentation of a graphical user interface using the Ventana Simulation (Vensim) Environment and its demonstration as a potential learning aid for software managers. Vensim is a Windows based modeling and simulation environment that provides a comprehensive set of analytical tools. This research focuses on the design considerations and prototype development of a graphical user interface that incorporates Vensim’s Causal Tracing features as an investigative tool. This research expands the work on model conversion and experimental interface design completed by Captain Richard D. Davis [Davis, 1995].

D. THESIS ORGANIZATION

Chapter II discusses the current software development and project management state of affairs, the problem areas in current methods of training software managers, and the role of system-dynamic software modeling and simulation play to solve them. Chapter III will provide an overview of VENSIM salient features and capabilities.
The user interface design and architectural considerations are discussed in Chapter IV and V. In Chapter VI, the user interface is fully explained. Chapter VII provides a demonstration case of the analysis capabilities of the user interface. Chapters VIII through X summarize this research effort.
II. RESEARCH BACKGROUND

A. SOFTWARE PROJECT MANAGEMENT CURRENT STATUS

Computers, associated software, and information are an integral part of every aspect of today's society. Because corporations and the Government, especially the Defense sector, depend more and more on information systems to support their critical business functions, the role of computers and software is increasingly critical. In a computer-bound environment, the more quickly and efficiently that high-quality software can be developed, the more flexible Industry and the Government can be in responding to changes in today's dynamic and unstable world. Software is a central ingredient for success. For these reasons, the demand for software systems has shown a fast growth. However, the record of software development and project management is not good.

Examination of the current state of affairs with software development and project management reveal very disturbing facts [Jones, 1994, p. 27-61]:

- Most new systems are extremely complex due to demanding requirements during the formal requirements phase or creeping user requirements (Changes to the project requirements which occur after the formal requirements phase).
- Most projects exhibit excessive schedule pressure, as the users or management try to force the completion of the project ahead of the capabilities of the development team.
- Delivered software for projects either do not work or fail repeatedly in operation with the subsequent increase in user dissatisfaction.
- Inaccurate cost estimating of the level of effort needed result in cost overrun and time delays.
- There is a lack of fully skilled managers in software management techniques. Many managers lack the relevant experience required for proper oversight.

As a consequence, many software-driven systems are delivered late, have cost overruns, rarely meet user performance requirements upon initial delivery, and are difficult to maintain in the outyears.
An investigative report completed by Senator William S. Cohen, affirms that:

The development and testing of large government computer programs can be as difficult to manage as any weapon systems acquisition. Like weapon systems, there is an incentive for agencies to initially focus on buying the platform—the computer hardware—rather than on software development and future personnel and operating costs. Decision makers may feel more comfortable reviewing something tangible, and agencies may push to buy into hardware before adequately thinking through what to do with it. Once the large hardware costs are incurred and the program bought into, inevitable problems with software development to tie together diverse hardware occur. At this point, it becomes extremely difficult to cancel or modify the program....Large computer acquisitions demand greater attention, because history shows that they are not being managed in the most efficient manner. Within the last four years, GAO published 74 reports on information technology programs. The reports identified problems with requirements analysis, management, cost/benefit analysis, and limited competition. [Cohen, 1994, p. 7]

Lack of relevant experience and education have been identified as a contributing factor.

Software managers are seldom adequately trained for their jobs, and many lack even rudimentary skills in normal managerial tasks. The root cause can be traced back to inadequate curricula at the undergraduate, graduate and professional level. Many software managers do not receive adequate training in the six basic tasks of software project management: sizing, estimating, planning, tracking, measurement, and assessment. [Jones, 1994, p. 52].

All too often, managers tend to treat the symptoms to these problems and not their causes. Also, many use “gut feelings” instead of hard data to develop quick fixes to management problems [Reifer, 1993, p. 1]. The use of intuition alone is not sufficient as “sheer intuitive judgment is unreliable about how the total process will change with time, even when there is good knowledge of the individual parts of the system” [Neimeir, 1990, p. 29.2.1].
Software managers need a robust tool that allows them to learn to cope with change and uncertainty in this dynamic complex environment. The use of models and simulations provide the means for experimenting complex situations involving change in order to gain insight without the expense of studying the real systems.

B. MODELS AND SIMULATIONS

Models and simulations are necessary to understand the behavior of complex system dynamics. Such a model makes use of time-based mathematical formulations to describe the behavior and structure of systems composed of interacting feedback loops. In system dynamics, the present state of the system will depend on all its past states (i.e. organization behavior or time history).

Behavior in complex systems is molded by the way people think about the world surrounding them. Models help us to better understand world problems and to evaluate courses of action in making decisions about a system. People create mental images or models to help them reflect on problems and cope with the uncertainty of making decisions. Senge characterized these mental models as “deeply engrained assumptions, generalizations, or even pictures or images that influence how we understand the world and how we take action” [Senge, 1994, p. 8].

However, a mental image or representation of the real system is not sufficient to explain behavior. At times, our mental model is “biased, fuzzy, incomplete, inaccurate, or just plain wrong” [Austin and Ghandforoush, 1993, p. 1]. For example, the average owner of a car has a very simple mental model of how the car engine works. He can model the engine of his car as a system of related components such as gasoline, a battery, spark plugs, ignition keys, oil, gears, and pistons. The owner’s mental image of how these parts of the system interact together under certain conditions, and how these interactions effect the overall behavior of the car help him deal with problems under normal operating conditions. His representation of an engine is sufficient for him to determine malfunctions in his auto. He knows that if the engine will not start, he can
attribute the possible causes of the malfunction to a dead battery or an out-of-fuel condition. [Mayhew, 1992, p. 81]

The car owner becomes helpless when his mental model can no longer explain unusual engine failures which affect the behavior of his car - the car’s ability to travel. The only available recourse to him is to enhance his mental model, or get an experienced mechanic, whose mental model of an engine is more accurate and complete due to his training and experience. Owners must increase their knowledge about the system by adding the missing parts and processes needed to further understand the inner workings of the engine. His mental model needs to become more complex and complete to properly diagnose the problem and fix it.

Managers, just like car owners, must constantly modify, alter, and transform their mental model, their image of the business, to better understand how their management system’s represent reality, how feedback relationships affect the dynamics and behavior of the system, and how to cope better with uncertainty in decision making. However, in the face of complex dynamic systems, the use of mental models, intuition, or perception alone is not sufficient to deal with the complexity of the management system, as “a vast body of experimental work demonstrates that individuals make significant, systematic errors in diverse problems of judgment and choice” [Senge and Sterman, 1991, p. 2]. In order to handle the complexity of dynamic managerial systems, a meticulous mathematical modeling approach and a tool to deal with its framework is needed since,

Managerial systems contain as many as 100 or more variables that are known to be relevant and believed to be related to one another in various nonlinear fashions. The behavior of such system is complex far beyond the capacity of intuition. Computer simulation is one of the most effective means available for supplementing and correcting human intuition. [Roberts, 1981, p. 6].

Computer simulation is of great value in the study and modeling of complex systems. It can be described as tool that executes a model that imitates the real life. Simulations allow us to model large masses of data and relationships that otherwise could not be easily understood. Simulations permit the analysis of thousands of variables, many interconnected by feedback loops, as they develop over time from a given starting point.
In this way, the effects of change in the initial condition and the impact of external variables and feedback on processes can be traced. Simulations are useful in the study of complex dynamic systems. They provide the means for controlled experimentation to gain insight at a fraction of the cost of studying the real physical system through other methods.

The use of simulation technology to facilitate learning has been evident for many years. Simulations help people to change their mental models. It helps to make intuitive conclusions about complex situations. As Jensen suggests,

Learners can be “placed” in worlds thousands of years forward or backward in time ... in “dangers” that are out of the question for real-world training and education. Science students can handle toxic and explosive materials in virtual labs ... Naval trainees can sail any day of the week through any type and level of virtual typhoon of their choosing. American pilots and tank commanders trained repeatedly in virtual worlds before embarking on real-world missions in the Gulf War. [Jensen, 1993, p.11]

Just as pilots train on how to fly an airplane under various conditions to enhance their skills and broaden their “mental model” of aircraft flying, software managers should be allowed to practice, in a software management “flight simulator,” their skills in managing complex software development projects. Simulations provide the necessary tools.

C. SMFS SYSTEM-DYNAMICS MODEL

Adequate representation of the complexity of the software development and management worlds require the use of a model capable of handling a myriad of variables, relationships, and processes to properly represent them. Such a model was created by Tarek K. Abdel-Hamid using Dynamo simulation language. An integrative system-dynamics based model of the entire software development process provides the foundation for the development of the Software Management Flight Simulator (SMFS). The SMFS is a metaphor for a computer-based simulation and learning environment. It intends to provide a gaming system in which managers can acquire experience and skills
to manage software projects safely, much like the flight simulators, where pilots are trained in handling unexpected flight conditions found under bad weather without risk to the passengers or the aircraft.

The mathematical formulation provided by Abdel-Hamid captures the micro components of project management, programming, testing and productivity into a continuous view of the software development process. His approach provides a more realistic view of the interactions and dependencies between the variables of software project management. [Abdel-Hamid and Madnick, 1991, p. 12]

The model consists of four major subsystems: human resource management, planning, controlling, and software production. Figure 1 depicts a conceptual view of the model with its flows and interconnections between the subsystems.

![Diagram](image_url)

**Figure 1.** Software Development Subsystems [Abdel-Hamid and Madnick, 1991, p. 22]

The Human Resource Management Subsystem captures the hiring, training, assimilation, and transfer of people in the project [Abdel-Hamid and Madnick, 1991, p. 63]. The Software Production Subsystem includes the design, coding, and testing phases

The model simulates the relationship between multiple functions of the software development process (e.g., design, coding, testing) and its management functions (e.g., planning, staffing, controlling) through the use of feedback or cause-and-effect loops. Figure 2 depicts a high level view of the model’s subsystems cause-and-effect relationship.

**Figure 2.** Causal Loop Diagram of Key Influences [Abdel-Hamid and Madnick, 1989, p. 1430]
The causal loop structures depicted in Figure 2 capture the circular relationships among cost and schedule estimates, the workforce size, and productivity through a series of feedback loops. These loops provide visibility of the many management-system's dynamic relationships that exist in a project. They also provide a continuous simulation capability for decision making. The flow of information, resources, and products used in a software project are modeled as a series of time based equations where calculations take place at discrete and small time intervals to provide numerical values as opposed to analytical results [Abdel-Hamid, 1993, p. 24].

This model provides the base needed for the development of a SMFS that, when implemented as an interactive application, allows the user to experiment with policies and strategies via games. This facilitates insight into the issues and difficulties that effect software projects without risking real projects and financial losses.
III. VENTANA SIMULATION ENVIRONMENT

A. OVERVIEW

The Ventana Simulation (Vensim) environment is a Windows-based set of tools designed to provide the means for developing and analyzing models. The environment provides all the functionality needed to define, modify, simulate and optimize models efficiently. Through the use of a suite of graphics, sketch and text editing, macro definition, analysis tools, data manipulation and optimization functions, Vensim allows greater flexibility in creating, documenting, and simulating models. Vensim allows the user to conceptualize a model in a sketch, complete the sketch by adding applicable mathematical formulations, simulate it, and present the results graphically or in table format quickly and easily.

The program is functionally divided into three components: the Workbench-Toolbox, the Sketch Tool, and the Equation Editor. The workbench is the main window from which Vensim is controlled, and which displays Vensim's main menu functions. It provides an input window for model development and output windows to display the results from analytical tools available in the toolbox. The Sketch tool is the main building tool. It allows the user to create and modify models. The equation editor allows the user to enter equations for the variables previously defined as concepts or words in the sketch diagrams. [Ventana, 1994, p. 11-12]

B. CAPABILITIES

Vensim provides an integrated environment for the development, analysis, simulation and optimization of system dynamics models. It is capable of incorporating data from external sources, providing subscripting capabilities to represent variables, and supporting the definition or specialization of functions by means of macros.

Vensim provides the mechanisms necessary to develop interfaces that allow user access and interaction with developed models. The application combines a model, or a series of models, with a user-designed and customized menu-screen-driven interface capable of
interacting and running a simulation and displaying highly formatted output data and results. A Vensim Application, or Venapps, defines the appearance and behavior of the interface by using a scripting language. Also, Vensim is capable of translating models developed in other modeling languages.

Vensim is a powerful modeling and simulation tool. It supports model building by allowing the user to sketch information or concepts in free text form. The text can later be arranged through the use of arrows to establish cause and effect among the concepts. An added feature of Vensim is that it allows the user to work with multiple views of the same problem, in which each view can show a different perspective of the problem in question.

Word and arrow diagrams establish the structure and relationship of the model in a graphical and straightforward manner. These sketches are known, within the system dynamics methodology, as causal loop diagrams. Vensim interprets these created diagrams as having a meaning. The attachment of arrows to words are interpreted as an indication of the direction of causality. They are beneficial in communicating the user’s mental image of reality as a set of objects joined by unidirectional arrows, as opposed to a set of interacting equations.

The equation editor allows the user to move effortlessly from a sketch to a working simulation. The editor facilitates the documentation and mathematical definition of the concepts by using the cause-and-effect information entered in the diagrams. Vensim automatically maintains the relationships between words and arrows depicted in the diagrams and the mathematical formulations to ensure consistency. It facilitates re-arrangement and customization of models. Since Vensim keeps track of the structure of the model, it makes model building much easier and more efficient.
C. CAUSAL LOOP DIAGRAMMING AND TRACING

Causal Loop Diagramming is the act of representing graphically important system structures and processes that have loops or feedback. As explained earlier, Vensim makes the task of building causal loop diagrams easy and fast. It also allows the thorough examination and analysis of the diagrams to determine what activities, if any, are needed for a more realistic representation of the problem being modeled.

Causal loop diagrams capture dynamic situations in a visual model. The user, when brainstorming which concepts to include in his modeling definition effort, needs to think about the progression of activities, the underlying structure, and its associated dynamics. Vensim provides an excellent way of structuring this information by enabling the user to do this graphically by means of drawing tools. Once the model is represented graphically, its applicable equations can be easily established and documented through the use of the on-line equation editor.

Vensim capabilities to develop causal diagrams is not limited to new models. Existing mathematical model formulations can be imported into Vensim. Graphical tools facilitate the development of its applicable causal diagrams. Given the model is visualized, further modifications or studies of the model's cause-and effect relationships can be performed quickly.

Vensim, through the use of tools, can quickly turn causal loop diagrams into simulations. A number of structural and dataset analysis tools are included in Vensim to provide the user with information about the structure of a model and help him in the analysis of simulations. The toolbar found in the Vensim Workbench provides access to the following important tools: the tree diagram, the document tool, the loops tool, the strip graph, and the table tool.
These tools form the foundation for Vensim’s Causal Tracing capability. With the tree diagram tool, Vensim can create a diagram of the cause or use of any variable. The strip diagram and table tools allow inspection of the graphs or numerical values of a variable displayed with its cause or use. The document tool displays the equation formulation of a variable, its unit of measure, and its selected values. These tools, when configured in a user interface, enable the user to quickly find his way from interesting simulation behavior to its root cause. [Ventana, 1994, p. 5]
IV. USER INTERFACE DESIGN CONSIDERATIONS

A. ABOUT USER INTERFACES

The objective of a user interface is to establish a communication exchange between the user, the computer system, and a software application capable of proving a desired functionality - what tasks must be carried out to accomplish a number of desired functions. This exchange is made through the use of dialogs in which symbols, commands, or actions give way to communications. During the course of these exchanges, each party can interrupt, ask, and correct the communication dialog. Involved in this exchange are physical devices, such as the keyboard or other input devices and display hardware, as well as the software that controls the dialog and exchange output.

The user interface plays an important role in understanding the application's underlying processes. It provides the means for the user to develop his attitude toward the application. A software application is considered to be user-friendly when a person with limited computer knowledge or experience has little difficulty using it. An application that is perceived as difficult to learn, due to inadequate or excessive functionality, or hard to interact, due to deficient interface design, will most likely frustrate the user. The result will be for users to label the application as "unfriendly" and to reject it, regardless of its computational efficiency [Ehrhart, 1990, p. 848]. Successful acceptance of the application depends on how well the provided user interface is designed to interact with the system to perform tasks.

Interaction can be achieved through command driven interfaces, menu driven interfaces, or graphical user interfaces. Command-based interfaces allow direct and to the point execution of the elements of the application by use of commands or abbreviations entered by the user. However, interfaces designed this way require the user to learn and remember a set of commands in order to execute the various elements within the application.

A menu driven interface is analogous to a restaurant menu for selecting food from a list. The available choices are listed and displayed on the screen in some logical order to facilitate the user selection of a desired item using an input device such as a keyboard or
mouse. This method is good for the inexperienced person because they do not have to memorize command instructions.

The graphical user interface is considered a more user friendly environment since it provides communication and ease of interaction between the application and the user through a graphics-oriented display and a “direct manipulation” environment. The concept of direct manipulation was coined by Shneiderman to describe an interaction style characterized by:

- Continuous representation of the object of interest;
- Physical actions or presses of labeled buttons instead of complex syntax;
- Rapid incremental reversible operations whose impact on the object of interest is immediately visible. [Shneiderman, 1992, p. 205]

A graphical user interface can accommodate the previous interaction methods easily. It combines objects such as menus, icons, dialog boxes, commands, and mouse control with rectangular work areas called windows. The use of graphics or pictures enhance the application ease of use since the user does not need to know cumbersome commands. In this type of interface, the user is able to display several windows on a single display screen. The user can manipulate text, graphics, or any other available element in any window displayed by switching among them with relative ease.

To most computer users, the software interface is the system itself. Therefore, friendly user interfaces must be able to facilitate the interaction of a set of input/output formats that the user can easily recognize have a functionality that the user can associate with his needs, and be able to engage in an easy man-machine dialogue.
B. DESIGN CONSIDERATIONS

A goal in designing interactive user interfaces is to develop a mode of communication that is both error tolerant, functional, and easily learned. With regards to user interfaces, their design should address two key questions:

What makes a user interface good?

How can, or should, a user interface be organized to be effective?

1. General Design Guidelines

Previous work in software user interface design by Mayhew [1992, p. 8-27] and Shneiderman [1992, p. 1-36] have identified and described several guidelines that affect the characteristics and aspects of any interface, and that designers should always have in mind in order to obtain well-designed user interfaces. These guidelines provide insight into several areas that must be considered for successful interactive interface design:

- what type of data is required and is relevant to the user at each step in the interface;

- how data should be presented (i.e., via text, graphics, tables, diagrams, etc.) to effectively capture the functionality of the application and ensure ease of use at the same time;

- what input devices facilities and methods, such as windowing and menus, should be provided for the users to communicate with the application;

- what output capabilities are needed (i.e., display types, hardcopy documentation, etc.);

- how flexible the interface should be designed to accommodate variations in user skills and preferences.
2. **Organization Guidelines**

The wide variety of human abilities, backgrounds, learning styles, and personalities pose a tremendous challenge for designers trying to organize the various interaction constructs and methods available in an application in a coherent interface. Shneiderman listed eight principles to adhere to during the design and organization of interactive interfaces [Shneiderman, 1992, p. 72] The principles are:

- Strive for consistency
- Enable users to use shortcuts
- Offer informative feedback
- Design dialogs to yield closure
- Offer simple error handling
- Allow easy reversal of actions
- Make the user sense they are in control
- Minimize short term memory requirements

Consistent organization is obtained when similar situations require similar sequences of actions from the user. The terminology used in prompts, menus, and commands should be similar to ensure the user is always aware of what to do next. The screen display should be formatted so that the various types of information, instructions, and messages always appear in the same general area. As users gain experience in interacting with the application, the interface should be able to provide the option of using key-based commands as shortcut to facilitate faster response and results display.

The interface feedback should be organized around simple messages that provide an indication of the system status from a user requested action, or as result of his inaction. The sequence of actions in an application should be organized into groupings with a beginning, a middle, and an end to prevent user disorientation. The user should not wonder where he is in the process at any time. The interface error handling or action reversal methods should have
explicit corrective actions for the user to recover from mistakes, or to return to previous states without having to restart the whole process again. The user should be able to feel in control of the interface. Unexpected outcomes are not welcomed by a user. The interface should tell the user whether the desired task was completed or not, and the reason for any processing delays.

Simplifying complex functions execution and input sequences make the interface easier by not forcing the user to remember a complicated set of code abbreviations, or command-syntax forms to efficiently control the interface dialog.

Interaction with computers is mostly carried out through a visual medium, namely the display terminal. Getting and directing the user attention are very important design and organization considerations. Chabay and Sherwood provided the following guidelines and suggestions for good, useful displays:

- Let the display determine structure - make the display central to program structure by tying data structures and programs to displays.
- Use simple displays.
- Use a judicious mix of text, pictures, and graphics.
- Highlight - emphasize most important parts of the display by means of underlining, reverse video, blinking, size, and color techniques and strategies.
- Use multiple “pages” - keep a strong content framework visible at all times by displaying several “pages” of related information in one screen display.
- Make display understandable by use of adjacency - new information should be presented near the most recently displayed information.
- Provide help and task instructions as part of its display. [Chabay and Sherwood, 1992, p. 155-186]
The manner in which information and results are displayed must be carefully organized in an interface because of the available screen's display size area. Improperly organized windows and menus can result in the user getting lost and frustrated in a multiple display system. Windows should be kept uncluttered and menu items should be organized and structured to minimize the amount of time the user needs to make a selection.
V. INTERFACE ARCHITECTURE

A. SYSTEM OVERVIEW

The SMFS was developed using the Vensim application development facilities. A Venapps or Vensim application combines a model with a set of interfacing instructions and rules for model interacting, control, display, and storage of results generated by the model simulation. The SMFS requires the use of three files in order to operate as an interactive simulation, and one file to store the results. These files are easily identified by its extensions as follows: Vensim Model File (.vmf), the Vensim Custom Definition (.vcd), the Vensim Graphics Definition (.vgd), and the Vensim Data File (.vdf).

The .vmf file contains all the equations and the relations that define the SMFS model. The .vcd files contains the necessary control information needed to run the SMFS. Specific display and format information is stored in the .vgd file. Data and simulation specific results can be found in the .vdf file. The Appendices A through C provide detailed coding and scripting information concerning the .vmf, .vcd, and .vgd files. Figure 3 depicts a high level view of the SMFS architecture.

![Diagram](image_url)

**Figure 3.** SMFS Architecture Overview
The user encounters a user interface upon starting the system. The .vcd file exercises the SMFS’ .vmf file as a simulation, and controls the user interaction via a set of instructions in script format. The scripted instructions, separated into screen views, carry out a number of methods and functions for simulation analysis, and a set of format instructions that generate, on demand, a number of predetermined graphics, reports, and help texts from the .vdg file. Screen display sequencing and user interaction are controlled through menus and input dialog boxes, respectively. The .vdf file, created at the beginning of the simulation, stores the value of each variable, at every time interval, during the running course of the SMFS.

The SMFSF interface allows the user to manipulate the values of the following selected variables at various time intervals during the course of a simulation:

- PROJDR - Project Duration
- TOTMD1 - Total Project Cost in Man-Days
- WFS1 - Staffing Level (Work Force)
- FRMPQA - Percent of Manpower Allocated to Quality Assurance

For the purpose of simulation analysis, the interface provides access to thirteen selected variables during the course of a simulation. However, this is not intended to limit the user access to variables in the model. The user is capable of accessing any variable within the SMFS’ model during the execution of the provided analysis routine at any time during a simulation run. The interface will always revert the selection of variables for analysis to the following list of variables as a default starting point:

- INSPRD - Instantaneous Productivity
- PRDPRD - Perceived Development Productivity
- PRDPER - Productivity in Last 40 Days Period
- PRTKDV - Tasks Developed in Last 40 Days Period
- PRMD - Person-Days Spent in Last 40 Days Period
- FRWFEX - Fraction of Work Force Experienced
- AFMDPJ - Actual Fraction of Man-Days on Project
- COMMOH - Communications Overhead
- SDVPRD - Software Development Productivity
- DMPTRN - Daily Manpower for Training
- CMRWMD - Cummulative Rework Man-Days
- PRDFDS - Last 40 Days Period’s Defect Density
- ERRGRT - Error Generation Rate

B. MENU STRUCTURE

The SMFS graphical interface is based on a menu driven dialog style network. This interaction technique was chosen because it allows inexperienced users to easily interact with the SMFS simulation. A cyclic, tree-structured menu hierarchy facilitates the execution of tasks, the display of information, and the navigation of screens within the user interface. Figure 4 illustrates the SMFS interface high level menu structure using a block diagram.

The SMFS Main Menu allows the selection of the following options: 1) Access the System Dynamics Primer, 2) View the SMFS Simulator Model Overview, 3) Play a New Game, 4) Analyze Previous Run Games, and 5) Exit the SMFS. Selection of option 1 or 2 provides to the user relevant information about the System-Dynamics theory and the SMFS model main modules. Option 4 begins a new simulation. The user is allowed to access a series of customized reports, graphs, and data tables with return to the Project control Center along the original path. The option Analyze Scenario opens a series of tools designed to facilitate analysis of the scenario at the current time interval. Upon completion of analysis, the user is returned to the Control Center screen to continue with the simulation. Menu item 4 also enables the user to load a previously run simulation file to perform post-mortem analysis. Option 5 terminates the simulation execution, closes the data .vdf file and returns the user to the initial location were the interface was first activated.
Figure 4. SMFS Interface Menu Structure
C. USER’S HANDBOOK

1. Introduction

The purpose of this manual is to allow the user of the SMFS to load, execute a simulation, and conduct analysis using the built-in features of the interface. The only assumptions that are made is that the user is working on an IBM compatible personal computer or similar system, that the user is familiar in using Windows-based software products, and that the user has the Vensim main program installed on his microcomputer.

The SMFS graphical user interface follows a menu driven format. The user is presented with a variety of choices within each menu. Selection of any menu item is accomplished by clicking the left button on the mouse, or by depressing the ALT key together with the first letter of the label identifying the menu item selection. The user has a series of exit options. Their access depends on where he is located with respect to the simulation execution. The user has the ability to quit the ongoing simulation and return to the Vensim workbench or to the Windows environment, depending on where the SMFS simulation was first initiated, return to a previous menu within the simulation, or exit in order to initiate a new simulation game via the interface.

2. System Requirements

The SMFS requires the following hardware and software for execution:

- An IBM 286 compatible computer or better running MS-DOS and Microsoft Windows 3.1 or higher, with a mouse system. For large model simulation, a 386 or better computer, with math coprocessor is recommended.
- Two Megabytes (MB) minimum RAM (four MB minimum required for large model simulation).
- A hard disk with 1.6 MB of available space for the Vensim program files minimum installation, 6 MB of disk space required for full installation.
- Vensim Professional DSS Simulation Environment, Version 1.61.
- Floppy Disk containing interface files Causal.vmf, Causal.vcd, Causal.vgd; and auxiliary files Base.vdf, Causal.vts, Vskt0000.bmp and Vskt0001.bmp.
3. Interface Starting Methods

Before you can start the SMFS interface, the files residing in the floppy disk must be copied to a working directory on the hard disk. The current version of Vensim will not load and execute simulation files stored on floppy disks. Vensim requires that these files be stored in the hard disk because it uses and creates a number of different files at once during simulation execution. Use Windows to create a directory beforehand and to copy the files to it. Once the working directory has been created with the applicable files, the user can launch the Vensim environment.

Once the Vensim environment is active, the user can activate the interface by way of one of two methods available. The first method assumes the user is performing editing work in the Vensim workbench area. To run the simulation interface from the workbench, the user must load first the Causal.vmf and the Causal.vcd files. Vensim Applications (Causal.vcd) require that the simulation model file (Causal.vmf) be active in the background for proper execution.

To load the Causal.vmf file, select **Open** from the **File** pull-down menu and select the file by double clicking on it. The SMFS Interface requires that the Causal.vmf file be displayed in the sketch mode. If the file is loaded in the text mode, select **View** from the pull-down menu and choose **Sketch** to change its view. To load the Causal.vcd file, press on the VCD button located at the bottom of the vertical toolset. This action displays a **Name of File to Edit** dialog box with a list of applicable filenames in a selection box. Select Causal.vcd by double clicking on it with the left mouse button. At this point, the two loaded files will be displayed (Figure 5), and the user is ready to start the SMFS Interface. From the File pull-down menu, select **Run App Int** option to activate the interface.

A variation of this method is obtained when the user selects **Open** from the **File** pull-down menu and select the .vcd radio button to change the filename type displayed in the selection box. By double clicking with the left mouse button on the Causal.vcd file, the Causal.vcd file will be automatically loaded along with the model file, and activated.
immediately. With these methods, the user can return to the Vensim workbench to perform editing on the simulation files.

![Vensim Workbench Environment](image)

**Figure 5.** The Vensim Workbench Environment

The second method to activate the SMFS interface is to have an icon configured to by-pass the workbench and launch the SMFS Interface automatically by double clicking on it. The easiest way to create the icon is to copy the Vensim icon to its program group by use of the Windows Program Manager's File>Copy command. Once copied, and using the Windows Program Manager's File>Property command, change the icon's properties as follows:

- **Description:** SMFS Interface
- **Command Line:** `C:\Vensim\Vensim.exe /Vensim\model\Causal.vcd`
- **Working Directory:** `C:\Vensim\Model`
- **Short Cut Key:** None
Once the changes have been made and the modified icon is saved, the interface is ready to be activated directly from the Vensim program group. This method is the recommended way to install and initiate the simulation interface because it will effectively prevent accidental modification by users by denying access to critical interface files.

4. Running the Simulation

Once the user interface is activated, the user is ready at this point to start a simulation or game. The SMFS Interface displays a welcome screen and waits for the user to initiate the game. The user is then presented with the menu screen shown in Figure 6. From this menu, the user can select an option from five choices available. The first four selections produce secondary menus with multiple options, as depicted in Figure 4. The fifth option terminates the simulation, and, depending on the method used to launch the interface, sends the user back to that starting location.

Software Management Flight Simulator

<table>
<thead>
<tr>
<th>System Dynamics Primer</th>
<th>Simulator Model Overview</th>
<th>Play a New Game</th>
<th>Analyze Previous Run Games</th>
<th>Exit Flight Simulator</th>
</tr>
</thead>
</table>

Figure 6. SMFS Main Menu
a. **System Dynamics Primer**

This menu item provides the user with concise information about System Dynamics' theory, principles, and notation formats. The option intends to provide background about the underlying theory used in developing the SMFS model. When the user selects this option, the interface presents the menu displayed in Figure 7.

![System Dynamics Primer](image)

**Figure 7.** System Dynamics Primer Menu Display

Clicking the **Primer** button displays a Vensim window allowing the user to print the information presented or to export the contents of the window. Clicking the **Model Overview** button causes the Simulator Model Structures menu to be presented. The **Return to Options** will take the user back to the main menu display, while the **Exit** Button will cause the interface to stop the current game and return the user to the location where the interface was first activated.
b. **Simulator Model Overview**

This option displays the Simulator Model Structure screen, which contains a menu of choices that allow the user to individually select and familiarize with any of the subsystems, sectors, or subsectors that, when exercised together, form the heart of the SMFS model simulation. The user can access any of the model structure’s causal diagram, as well as display a Vensim window with details about the structure’s particular behavior. For example, if the user decides to choose the **Human Resources Subsystem** option by clicking its button, the display shown in Figure 8 will be presented.

![Diagram](image)

**Figure 8. Human Resources Management Subsystem Structure**

Every option available in the Simulator Model Structure menu presents similar screens that contain the applicable causal diagram for the component in question, and the button bar shown at the bottom of Figure 8 above. The **Model Component** button allows navigation through the model structures without having to return back to previous screens. The **Print** buttons will display a print dialog box to print the causal diagram. The **Info** button displays a Vensim window with information about the structure being reviewed. The **Return to Options** and **Exit** buttons are mechanisms to return back to the main menu of the interface, or to terminate the simulation and leave the interface.
c. **Play a New Game**

This option causes the SMFS interface to initiate a new simulation by displaying a filename dialog box and requesting the user to enter a filename for the simulation data file (.vdf). The user has the option of typing a new name, or reusing a previous filename. A series of sequential screens are presented to the user to provide instructions and game background about the particular situation being simulated. The user is then requested to make his first decision by entering values for the staffing level desired and what percentage of it should be allocated for QA purposes. The interface then proceeds to present the Control Center display (Figure 9), the location where the user will control the ongoing simulation until completion.

![Control Center](image)

**Figure 9. SMFS Interface Control Center**

The Control Center screen is divided into four information and control areas. The top two panels display information concerning the simulated software project and the current statistics of the project. The third panel, on the left, allows the user to enter his decisions concerning key project metrics under his control. The fourth area
presents a series of buttons that give access to a series of pre-defined custom reports, data tables, graphs, and access to the SMFS analysis capabilities. The user advances the simulation in increments of 40 days by clicking the **Advance Time** button.

The simulated software project prevalent situation at a given period of time must be analyzed to determine what decisions must be made to maintain the project under cost and schedule control. The user determines what his next decisions will be by reviewing the available reports (Figure 10), graphs, or by use of the provided analysis functions. The details of the **Analyze Scenario** button will be presented in the next section. The user enters his decisions in the Input Variable panel and advances the time to the next reporting time period. The user repeats this process until the Project Status Report indicates the Percent Delivered Source Instructions Reported Complete equals 100% or the Game Over dialog box appears. At the completion of the simulation, the user selects **OK** to close the dialog box, then clicks the End Simulation button to return to the beginning of the SMFS.

![Software Management Flight Simulator](image)

**Figure 10. SMFS Report Examples**

34
d. **Analyze Previous Run Game**

Selection of this option causes the SMFS interface to display the Simulation Post-Analysis menu screen (Figure 11), which contains a series of commands and choices set to assist the user in performing analysis.

**Figure 11.** SMFS Post-Analysis Capabilities

The purpose of this option is to facilitate the loading of up to eight stored simulations or games, and to enable the user to study and make performance comparisons between these simulation runs to determine particular behavior tendencies. Even though the user can load up to eight scenarios at a time, it is recommended that no more than four be loaded at any given time since it becomes difficult to read graphs with more than four scenarios loaded if you have a small monitor display.

In order to conduct post-analysis of games, the user must load a previously run game. Once it is loaded, the user can easily perform analysis of the output. The user will load in any previous games by clicking on the Load button. This action will display a Load and Reorder Previous Scenarios dialog box, which displays the names of
previously saved games. This command will display two filename lists. On the left box is a list of loaded scenarios and on the right is a list of scenarios that are stored. The user can select/deselect the game of interest by following the steps below:

- The << button loads scenarios. Click on the scenario you want to load in the right hand list, and then click the << button. If you do not have a mouse available, use the TAB and Return (or Enter) key combination to move among the boxed selection items and to make your selection.

- The >> button unloads scenarios. Click on the scenario you want to unload in the left hand list, and then click on the >> button. If the list on the right is empty, you will need to run some simulation games. You can do that by exiting to the main menu and selecting to run a game.

The SELECT option allows the user to select a variable of interest for analysis (other than the default variable, which is the active variable at the time the simulation was last terminated.). A Variable Selection dialog box that contains a list of variable names in the current model will be displayed. The user can make his selection by scrolling down the list and double-clicking on any variable on the list, by clicking on the variable of interest and then clicking on the SELECT button, or by typing in the space provided the name of the variable and then clicking the SELECT button.

As previously discussed in Chapter III, Causal tracing is a process that assists the user in determining the underlying causes of model behavior, and the differences in behavior between different scenarios. There are four options available for doing causal tracing. They are:

- **Tree** - This option enables the display of the causes of the variable of interest, as a tree network that branches from the right. Variables that are shown between parenthesis to indicate that they appear somewhere else in the tree. Also, this option displays a graphical representation of the variable of interest. An option menu is provided for the user to continue further analysis.

- **Graph** - this option is a refinement of the Tree option. It allows the display of the equation and other pertinent information of the variable of interest, and the graphical display of a predetermined set of variable levels affecting the variable of interest. An option menu is provided for the user to continue further analysis.
• **Uses** - This option enables the display of the uses of a variable as a tree network branching from the left. Variables are shown between parenthesis to indicate that they appear somewhere else in the tree. Also, this option displays a graphical representation of the variable of interest. An option menu is provided for the user to continue further analysis.

• **Loops** - This option displays a list of all feedback loops passing through the variable of interest. The list is ordered from the shortest loop (the one involving the least number of variables) to the longest loop. Loops provides useful information about model interactions. Also, this option provides useful information about the variable of interest. An option menu is provided for the user to continue further analysis.

Each of these options present the user with a menu selection of their own to allow the user to select a different variable of interest and switch "on the fly" between the various causal tracing available functions to study the variable characteristics and behavior. Figures 12 through 15 provides display examples of these options.

---

**Figure 12.** SMFS Post-Analysis Tree Causal Tracing Example
Figure 13. SMFS Post-Analysis Graph Causal Tracing Example

Figure 14. SMFS Post-Analysis Uses Causal Tracing Example
Figure 15. SMFS Post-Analysis Loop Causal Tracing Example

The cumulative functionality designed in these options is described below:

- **Tree, Graph, Uses, and Loops** causal tracing utilities, as previously discussed.

- **Data** - This option enables the display of a Vensim Window of the numerical values of the variable of interest and its causes in a table format by time periods. This window can then be printed, its content copied to the Windows clipboard, or deleted when no longer needed.

- **Print** - This command let the user print applicable **Tree, Graph, and Loops** outputs and the equation and other related information about the variable under causal tracing.

- Trace on Highlight - This option enables the user to change the variable of interest by clicking on any variable displayed in the tree or uses network, graphics, or loop listing, to highlight it. Clicking on the **Yes** button causes the interface to present the new variable of interest and the respective **Tree, Graph, Uses, or Loops** display, depending on where the user selected to make the change.

- **Select** - this option allows the user to change the variable of interest by choosing from a list, as previously discussed earlier in this chapter.
• **Help** - This option will cause a Vensim window to be displayed with instructions and concise information about the functionality of the options available at that particular screen display. The window contents can be sent to a printer or copied to the Windows' clipboard.

• **Exit to Analysis** - This item allows the user to return back to the Post-Analysis screen menu.

The **List Differences** option allows the user to make comparisons of the initial constants used by the various loaded games or simulations by means of a table of numerical values. The user, at his discretion, can print this table or opt to return back. The **Help** option will present Vensim information window about the menu items and provide instructions on how to use them. The information contained in this window can be printed, copied to the Windows clipboard, or deleted when no longer needed. The **Return to Option**, when activated, causes the interface to return the user to the main SMFS menu screen, or to terminate the application, respectively.

It should be noted that the analysis capabilities described in this section are, with a few exceptions, identical to the analysis capabilities contained in the **Analyze Scenario** module described in paragraph 4.c above. The differences are as follows:

• The Simulation Analysis menu that is controlled by the **Analyze Scenario** button is designed to force the user to terminate the game at the control Center screen always. The Simulation Post-Analysis menu provides facilities to exit the simulation or to initiate a new game directly.

• The option for **Selecting a New Variable** in the Simulation Analysis Menu has been fixed to the variables listed in Figure 16 in order to focus the user's initial analysis of the situation. The user still has access to any variable in the model by using the Trace on Highlight command available in the **Tree, Graph, Uses, and Loops** causal tracing function facilities. The Simulation Post-Analysis menu makes use of a dialog box to provide global access to the variables in the model.

• The Simulation Analysis menu does not have the following items available: Load, Unload, and Reorder Previously Run Scenarios and the List Differences of First Two Loaded Scenarios.
VARIABLE SELECTION
OPTIONS
- Instantaneous Productivity
- Perceived Development Productivity
- Productivity in 40 Days Period
- Tasks Developed in 40 Days Period
- Person Days Spent in Period
- Fraction of WF Experienced
- Actual Fraction of Man-Days on Project
- Communications Overhead
- SW Development Productivity
- Daily Manpower for Training
- Cumulative Rework Man-Days
- Period's Defect Density
- Error Generation Rate

Figure 16. Variable Selection Menu

e. Exit

The Exit button will cause the SMFS application to terminate its execution and return the user to the initial activation environment. Depending on the method in which the SMFS Interface application was first launched, the user will be returned to either the Vensim workbench or the Windows program manager.

5. Simulation Results Storage

The simulation’s initial data and game results are stored by the Vensim environment as datasets, in Vensim Data format (.vdf). When the user selects to play a new game from the menu, the SMFS will ask the user to input a filename. This filename is important because it will contain all the initial data used to activate the simulation, as well as the results of the user interaction with SMFS interface. This files are saved using a Vensim specific format. The files can not be examined or modified via a text editor. The Vensim workbench has facilities to manually load, manipulate, and export the results contained in a .vdf file. The Post-Analysis facilities of the SMFS interface allow the user to only load these files for display and later study.
VI. CAUSAL TRACING DEMONSTRATION

A. INTRODUCTION

This chapter will provide a demonstration of the utility of the SMFS interface by simulating two scenarios of equal size to investigate the impact of different staffing profiles on project performance. The scenarios, Project 1 and 2, focus on executing different hiring policies while keeping other variables under the manager’s control constant. The simulation results are compared and the applicability of Brooks’ Law is investigated using the SMFS post-analysis causal tracing facilities.

B. INITIAL CONDITIONS

Once the SMFS is initiated, the manager receives the project’s initial estimates and available staffing level. In both scenarios, at the start of the project, the manager is informed that a project of 42,879 Delivered Source Instructions (DSI) is to be completed in 296 days. The manager must determine the number of people he desires to hire initially and what percentage of the total staff hired will be set aside to perform quality assurance tasks. The initial estimate of staff size is determined by dividing the estimated value of effort needed in person-days (pd) by the value of the development time, in days. The initial estimate for the effort needed to complete the project is provided by the following Constructive Cost Model (COCOMO) equation for medium sized project:

\[ \text{Effort} = 2.4 \times (\text{KDSI})^{1.05} \quad \text{(pd)} \quad (1) \]

Assuming 19 working days in a calendar month, the effort needed for developing and testing a project of 42,879 DSI (42.879 KDSI) is:

\[ \text{Effort} = 2.4 \times 19 \times (42.879)^{1.05} = 2360 \text{ pd} \quad (2) \]

The initial number of people needed is calculated as:

\[ \text{Number of people} = \frac{\text{Effort}}{\text{Time to develop}} \quad (3) \]

\[ = \frac{2360}{296} = 8 \text{ people} \]

Once the initial effort needed is calculated, the manager is ready to make his first staffing decision and continue his simulation. The SMFS gaming interface allows the
manager to vary the estimate of the effort skill needed, the schedule, the staff level, and the percentage of staff to allocate to quality assurance in future reporting periods. Only the staff level will be allowed to change in each reporting period. In both scenarios, the management will maintain ten percent of the staff performing quality assurance tasks for the duration of the project, the effort needed at 2,360 person-days, and the time to develop the project to 296 days. The manager’s challenge is to accomplish the project under these conditions by only varying the level of staff in the project.

C. SIMULATED PROJECT BEHAVIORS

Both scenarios are started by hiring eight people. As time progresses, the manager monitors the project’s behavior by making use of the graphs, data tables, and pre-determined custom reports in the Control Center screen of the SMFS interface. By monitoring the project’s reported size in DSI, the number of DSI completed, the reported productivity, and the effort expended, the manager can determine how much progress the project has achieved and what additional effort is needed to complete the project within the given constraints.

New project tasks are reported as better understanding of the requirements are achieved by the people working on the project or because of user-directed specification changes. These new discovered tasks may or may not generate an adjustment to the current project’s level of effort and schedule estimates. As a consequence, the staffing level required to accomplish the added work may or may not be increased as time progresses. Before hiring new people, managers take in consideration the ability of the experienced staff members to absorb (for example, in terms of orienting and training, both technically and socially) new employees. As new people are added, the project productivity drops because newcomers’ training is usually carried out by the project staff, reducing the staff own productivity as result. These considerations influence the manager decision of when to hire people and at what level. [Abdel-Hamid, 1989, p. 114]
1. Project 1

In this scenario, management is willing to increase the staff level to meet the project’s schedule. As the project approaches its completion date, the manager realizes that the project can not be completed in the time left and with the current staff. As result, at day 240, the manager hires more people. Considering the current project size, the reported productivity, the number of DSI completed, and the time left, he determines he must double the current staff (from eight to 16 persons) in order to finish the project on schedule. On completion, this scenario delivered 64,000 DSI (a 49% increase) in 2,597 person-days (a 10% increase) and 320 days (a 8% increase). Details of the simulation results reported by the SMFS’s pre-defined reports are summarized in Table 1.

<table>
<thead>
<tr>
<th>Report Period, days</th>
<th>0</th>
<th>40</th>
<th>80</th>
<th>120</th>
<th>160</th>
<th>200</th>
<th>240</th>
<th>280</th>
<th>320</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Project Duration, days</td>
<td>296</td>
<td>296</td>
<td>296</td>
<td>296</td>
<td>296</td>
<td>296</td>
<td>296</td>
<td>296</td>
<td>296</td>
</tr>
<tr>
<td>Project Effort Budget, pd</td>
<td>2360</td>
<td>2360</td>
<td>2360</td>
<td>2360</td>
<td>2360</td>
<td>2360</td>
<td>2360</td>
<td>2360</td>
<td>2360</td>
</tr>
<tr>
<td>Project Staff Levels, p</td>
<td>2</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Estimates Reported</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reported Size, DSI</td>
<td>42879</td>
<td>43210</td>
<td>44832</td>
<td>48732</td>
<td>55667</td>
<td>60859</td>
<td>63306</td>
<td>63962</td>
<td>64000</td>
</tr>
<tr>
<td>Reported Progress</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% Complete DSI</td>
<td>0</td>
<td>7.71</td>
<td>18.02</td>
<td>28.57</td>
<td>37.04</td>
<td>46.90</td>
<td>58.52</td>
<td>85.26</td>
<td>97.60</td>
</tr>
<tr>
<td>Completed DSI</td>
<td>0</td>
<td>3331</td>
<td>8078</td>
<td>13923</td>
<td>20621</td>
<td>28542</td>
<td>39580</td>
<td>54532</td>
<td>62467</td>
</tr>
<tr>
<td>Effort Expanded to Date, pd</td>
<td>0</td>
<td>165.84</td>
<td>425.08</td>
<td>716.59</td>
<td>1018</td>
<td>1322</td>
<td>1627</td>
<td>2049</td>
<td>2597</td>
</tr>
<tr>
<td>Reported Productivity</td>
<td>-</td>
<td>20.09</td>
<td>19.00</td>
<td>19.43</td>
<td>20.26</td>
<td>21.57</td>
<td>24.31</td>
<td>26.65</td>
<td>25.31</td>
</tr>
<tr>
<td>Staff Levels</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Current Staff Size</td>
<td>2</td>
<td>5.71</td>
<td>7.03</td>
<td>7.46</td>
<td>7.59</td>
<td>7.82</td>
<td>7.62</td>
<td>12.66</td>
<td>14.44</td>
</tr>
<tr>
<td>Programming Staff</td>
<td>1.8</td>
<td>5.14</td>
<td>6.33</td>
<td>6.71</td>
<td>6.83</td>
<td>6.86</td>
<td>6.66</td>
<td>11.39</td>
<td>13</td>
</tr>
<tr>
<td>QA Staff</td>
<td>0.2</td>
<td>0.57</td>
<td>0.70</td>
<td>0.75</td>
<td>0.75</td>
<td>0.76</td>
<td>0.76</td>
<td>1.27</td>
<td>1.44</td>
</tr>
<tr>
<td>% Experienced, Reported</td>
<td>100</td>
<td>49</td>
<td>58</td>
<td>69</td>
<td>77</td>
<td>82</td>
<td>85</td>
<td>62</td>
<td>68</td>
</tr>
</tbody>
</table>

Table 1. Simulation Statistics for Project 1
2. **Project 2**

In this second scenario, the manager perceives the project to fall behind schedule because of the gradual increase in the project size. However, the manager, as a firm believer of Brooks’ Law, opted not to increase the staff level. In this case, the manager decides to maintain the staff level constant at eight persons during the late stages of development to avoid the negative impact that communications and training overheads have on software development productivity. As a trade-off, he rather relaxes the development schedule. On completion, this project delivered 64,000 DSI (a 49% increase) in 2,541 person-days (a 8% increase) and 360 days (a 22% increase). Details of this simulation’s results, as reported by the SMFS’s pre-defined reports, are summarized in Table 2.

<table>
<thead>
<tr>
<th>Report Period, days</th>
<th>0</th>
<th>40</th>
<th>80</th>
<th>120</th>
<th>160</th>
<th>200</th>
<th>240</th>
<th>280</th>
<th>320</th>
<th>360</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Project Duration, days</td>
<td>296</td>
<td>296</td>
<td>296</td>
<td>296</td>
<td>296</td>
<td>296</td>
<td>296</td>
<td>296</td>
<td>296</td>
<td>296</td>
</tr>
<tr>
<td>Project Effort Budget, pd</td>
<td>2360</td>
<td>2360</td>
<td>2360</td>
<td>2360</td>
<td>2360</td>
<td>2360</td>
<td>2360</td>
<td>2360</td>
<td>2360</td>
<td>2360</td>
</tr>
<tr>
<td>Project Staff Levels, p</td>
<td>2</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Estimates Reported</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reported Size, DSI</td>
<td>42879</td>
<td>43210</td>
<td>44832</td>
<td>48732</td>
<td>55667</td>
<td>60859</td>
<td>63306</td>
<td>63957</td>
<td>64000</td>
<td>64000</td>
</tr>
<tr>
<td>Reported Progress</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% Complete DSI</td>
<td>0</td>
<td>7.71</td>
<td>18.02</td>
<td>28.57</td>
<td>37.04</td>
<td>46.90</td>
<td>62.52</td>
<td>82.48</td>
<td>93.87</td>
<td>100</td>
</tr>
<tr>
<td>Completed DSI</td>
<td>0</td>
<td>3331</td>
<td>8078</td>
<td>13923</td>
<td>20621</td>
<td>26542</td>
<td>39580</td>
<td>52754</td>
<td>59950</td>
<td>63320</td>
</tr>
<tr>
<td>Effort Expended to Date, pd</td>
<td>0</td>
<td>165.84</td>
<td>425.08</td>
<td>716.59</td>
<td>1018</td>
<td>1322</td>
<td>1627</td>
<td>1932</td>
<td>2236</td>
<td>2541</td>
</tr>
<tr>
<td>Reported Productivity</td>
<td></td>
<td>20.09</td>
<td>19.00</td>
<td>19.43</td>
<td>20.26</td>
<td>21.57</td>
<td>24.31</td>
<td>27.3</td>
<td>27.26</td>
<td>26.87</td>
</tr>
<tr>
<td>Staff Levels</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Current Staff Size</td>
<td>2</td>
<td>5.71</td>
<td>7.03</td>
<td>7.46</td>
<td>7.59</td>
<td>7.62</td>
<td>7.61</td>
<td>7.60</td>
<td>7.60</td>
<td>7.60</td>
</tr>
<tr>
<td>Programming Staff</td>
<td>1.8</td>
<td>5.14</td>
<td>6.33</td>
<td>6.71</td>
<td>6.83</td>
<td>6.86</td>
<td>6.85</td>
<td>6.84</td>
<td>6.84</td>
<td>6.84</td>
</tr>
<tr>
<td>QA Staff</td>
<td>0.2</td>
<td>0.57</td>
<td>0.70</td>
<td>0.75</td>
<td>0.75</td>
<td>0.76</td>
<td>0.76</td>
<td>0.76</td>
<td>0.76</td>
<td>0.76</td>
</tr>
<tr>
<td>% Experienced, Reported</td>
<td>100</td>
<td>49</td>
<td>58</td>
<td>69</td>
<td>77</td>
<td>82</td>
<td>85</td>
<td>87</td>
<td>88</td>
<td>89</td>
</tr>
</tbody>
</table>

**Table 2. Simulation Statistics for Project 2**
In summary, both hiring policies resulted in projects that were underestimated, late and overbudget. Project 2, the project having a constant, stable staff over the entire period of development was finished later than Project 1, the scenario that had more people added later. These results describe a behavior that is not consistent with what Brooks’ Law, which states that adding people to a late project delays the project further. To investigate the reasons why this happens, the causal tracing tools available in the SMFS were used to explore the impact that these hiring practices had on productivity.

D. CAUSAL TRACING

By using the SMFS’ Simulation Post-Analysis tools (Figure 11), the two cases can be easily loaded for comparison. Once the two scenarios are loaded, we must select what variable of interest we want to investigate. Since managers assess progress by evaluating productivity, the variable INSPRD (Instantaneous Productivity) is selected as a starting point. Clicking on the Tree button in the Simulation Post-Analysis screen displays the causal tree and the graph for INSPRD (Figure 17).

The INSPRD graph indicates that productivity fell at the beginning of the projects and then rose steadily, as the projects progressed in time. After peaking toward the end of the coding phase, the productivity of both projects decline at different rates. From the causes tree, the manager can observe that the results of this variable depends directly on two other variables: SDVRT (Software Development Rate) and TOTDMP (Total Daily Manpower). Also, from the causal tree, notice that INSPRD is affected indirectly by the behavior of variables found within three additional levels. A printout of the causal tree can be obtained easily from this screen. The printout can be used as a quick “road map” guide for further causal tracing. To examine behavior, click on the Graph button to display the direct causes of INSPRD in graphical form (Figure 18).
**Figure 17.** Instantaneous Productivity Tree Causal Tracing

**Figure 18.** Instantaneous Productivity Graph Causal Tracing
In this screen, it is observed that the Software Development Rate increases as the Total Daily Manpower increases with time. **INSPRD** decreased initially because Total Daily Manpower (the denominator) increases at a faster rate. Toward the end, the Software Development Rate slows down as we add more people to Project 1, while Project 2 decreases at a faster rate. To further investigate this behavior, we select the **SDVRT** variable, and click the **Tree** button to display its associated causes (Figure 19). Using this tree as a road map, and by using the **Graph** Button to examine **SDVRT** behavior, we obtain the set of graphs shown in Figure 20.

![Graph for SDVRT](image)

**Figure 19.** Software Development Rate Tree Causal Tracing
Figure 20. Software Development Rate Graph Causal Tracing

Since we are interested in productivity and its causes, select the variable **SDVPRD** (Software Development Productivity), one of the causes of **SDVRT**. From the graphs in Figure 21, **SDVPRD** is dependent on the behaviors of **MPDMCL** (Multiplier to Productivity due to Motivation and Communication Losses) and **POTPRD** (Potential Productivity).
The MPDMCL indicates an equal increase in motivation and communications overhead for both projects. By tracing back MPDMCL (Figure 22), AFMDPJ (Actual Fraction of a Man-Day on Project), and COMMOH (Communications Overhead), its behavior can be easily inspected. Causal tracing of AFMDPJ results in the set of graphs shown in Figure 23. Its behavior is mainly dependent on WRADJR (Work Rate Adjustment Rate). The WRADJR graph suggests that, as the projects approach the end of the coding phase, people tend to work more hours in an attempt to bring the project back on schedule to meet its deadline. This additional effort causes the MPDMCL to increase and positively affect the projects’ SDVPRD.

Figure 21. Software Development Productivity Graph Causal Tracing
Figure 22. Multiplier due to COMMOH and Motivation Graph Causal Tracing

Figure 23. Actual Fraction of a Man-Day on Project Graph Causal Tracing
Further, tracing the variable COMMOH, shown in Figure 24, indicates that it is proportional to the total workforce level, TOTWF. As expected, an increase in the staff level creates a rise in COMMOH, as illustrated by the behavior of Project 1 in Figure 24. In turn, this increase in COMMOH causes MPDMCL to be smaller, which impacts negatively on the project’s SDVPRD.

Continuing tracing the causes of SDVPRD, the POTPRD drops when people are added to Project 1, but starts to increase as the project approaches completion. The POTPRD in Project 1 is lower because hiring new inexperienced people at day 240 lowers its overall value.

Figure 24. Communications Overhead Graph Causal Tracing
Figure 25. Potential Productivity Graph Causal Tracing

Figure 25 indicates that POTPRD depends on the behavior of the ANPPRD (Average Nominal Potential productivity) and a factor, MPPTPD, that accounts for learning. From the graph, Project 1 ANPPRD drops drastically as new people are added to the project, but starts to recover as the new staff gain in experience.

Double clicking on the variable ANPPRD displays its causes, as shown in Figure 26. The FRWFEX (Fraction of Workforce that is Experienced) is the major contributor cause for the value of the ANPPRD.
Figure 26. Average Nominal Potential Productivity Graph Causal Tracing

The SMFS’ causal tracing tools assisted in revealing the following: As the total number of people increases in Project 1, the Communications Overhead associated with new people will increase. This in turn, affects the Software Development Productivity of the whole team. Similarly, as the number of people increases, the amount of experienced personnel in the project available to perform coding decreases, which in turn decreases the Potential Productivity of the group. In Project 2, the Potential Productivity is higher towards the end. The ultimate effect in Software Productivity depends on how fast the number of people is integrated and assimilated in the project.

In our scenarios, adding more people lowers the Average Nominal Productivity, which in turn, reduces the Potential Productivity of the team. This decline, when coupled with the associated communications overhead, further reduces the Software Development Productivity, which influences the Software Development Rate and Instantaneous
Productivity. However, even though the nominal software development productivity in Project 1 dropped and \textit{COMMOH} increased as new staff was hired, the amount of productivity drop in Project 1 was not significant enough to offset the increase in staff. Since Project 1 ended before Project 2, it can be concluded that Brooks' Law does not apply in the project environment modelled here. Similar results were observed in [Abdel-Hamid, 1991].

In summary, this demonstration illustrates the utility of the SMFS Simulation Post-Analysis Screen. Similarly, a user could perform causal tracing on any variable found in the underlying model by using the analysis option available in the SMFS Control Center screen while running a simulation.
VII. CONCLUSIONS

With management's increasing reliance on mission-critical systems and the operational data they supply, information systems management faces a gigantic training and educational task. Managers need to learn how to manage in a computer-abundant world because they have the responsibility for ensuring successful delivery of the software.

Modeling and simulations play an important role in educating managers on the software project management arena. They help training students of management, as well as experienced managers on a very complex and difficult process.

Managers always have mental models, and will always modify them in the face of uncertainty. The user interface provided facilitates the process of developing, modifying and enhancing effectively the manager's mental model of the software development and project management processes that they are responsible to manage. It helps in increasing the managers current state of knowledge about how to grasp particular courses of events in software projects affected by complex, and often invisible feedback processes.

Decisions affecting complex software projects, involving highly sophisticated and technological systems, can have far-reaching consequences socially and economically. These consequences are often difficult for decision makers to predetermine. For this reason, many managers rely on their best judgment when developing plans and forecasts. Lack of experience in dealing with the complexities of software projects reveal a fundamental shortcoming. The interface provided in this thesis has been developed to help managers and students of management explore new strategies, develop new attitudes and views, and examine possible future consequences of their actions by means of causal tracing.

In a conventional classroom training, the managers are taught a wide range of material that they must remember when selecting the pertinent information to perform a particular task. This method emphasizes learning by facts. Computer-based simulation provides a training environment that differs from the traditional, lecture-based classroom environment. It forces the student to confront their weaknesses in the comprehension of
the material immediately, as opposed to merely passively receiving a lecture. The use of a flight simulator metaphor allows software project management training to be on-demand, interactive, and specific to the individual. The graphic user interface allows the manager to practice skills, and assist in developing a more complete view of their work and of reality.
VIII. ACCOMPLISHMENTS

A. USER INTERFACE

The principal focus of this thesis is to develop a graphical user interface capable of effectively implementing feedback or causal tracing, and to demonstrate its utility as a learning tool. Given the proven model for software development, the interface developed and documented in this thesis accomplishes these goals.

The interface provides a capable tool for training students and managers in making program decisions and understanding the possible unintended consequences that such decisions may have in a fictitious project’s future by use of the various causal tracing mechanisms available within the SMFS Vensim based application. These mechanisms were implemented using Vensim Application (Venapps) scripting language. The equation modifications needed in the model to implement causal tracing were performed. A menu structure that provides an easy road map to information reporting and analysis was developed. A windows-based interface capable of displaying simulation results in several graphical and text formats, as well as displaying help information, was successfully created following the best guidance available for design of user interfaces.

The resulting easy-to-use interface makes the model more interactive, rather than just providing an answer requiring an interpretation by an expert. Users can either analyze their decisions while running the simulation or after the simulation has ended. Further, they can compare a number of previous simulation results to understand their different approaches, interests, and consequences. This feature provides a very powerful source for learning common, important, and often difficult real problems in software project management.
B. USER’S HANDBOOK

A user handbook was written to provide guidance on how to use the graphic user interface. Emphasis was placed on describing what the interface is capable of doing instead of the fine details of how the system operates.
IX. LESSONS LEARNED

The Vensim sketch tool allows the designer to easily create and modify the feedback structure, and the equations of a model visually. Any model imported as text can be easily depicted into a Causal Diagram, with shapes and color easily assigned to enhance its meaning. However, in the process of calling the desired variables into the sketch pad, care must be taken not to delete a variable name, if by mistake, it is selected to be shown in the sketch pad. Using the deletion tool will cause the variable to be completely eliminated from the model. Even though the environment warns you of the consequences of doing the requested action, the chance of deleting a variable by mistake is present. If you do, you will be forced to recreate the variable again - a difficult task if your model is very complex. Use the hide tool to eliminate the undesired variable instead of deleting it.

The development of the SMFS interface requires you to become very familiar with the Vensim scripting language and command syntax in order to successfully design an application. This proves to be a very challenging task. First, the Vensim documentation is very general in certain areas, and at times, very vague about most of the command syntax, and instructions provided. This makes the process of designing graphical applications a lengthy, tedious, iterative trial and error effort. The location and size of the objects, and of the screens themselves, are required to be specified in percentage of screen size, which is not conducive to rapid development of applications.

Second, to view the results of the commands used, the current version of Vensim requires the designer to launch the application to see its effects, then to exit it, if further modification to the command is needed. This makes the design of applications a time-consuming process.

Third, when generating text-based reports, their presentation to the screen is controlled by specifying its location in screen pixels, its width in number of characters, and its height in number of lines. This makes the generation of highly structured reports and text files a very difficult task. Also, the report appearance will vary according to computer monitor in use and the type font specified. Lastly, changes made to the .vgd

61
files using the Vensim workbench editing tool require that the designer exit the Vensim program and launch the Vensim program again for changes to take effect - a very cumbersome method to update text based reports.

In sum, Vensim will produce a successful interactive application, given sufficient time to build experience in using the available commands, and patience to go through the testing of the designed screens.
X. FUTURE DIRECTIONS

The interface provided in this thesis can help students enhance their understanding of the phenomena involved in a project management situation. However, future research is necessary exist in the area of interface refinement and expansion. Future interface design should implement a game “rewind” function. The new interface capability should allow the user that is running a simulation to be able to playback the current simulation to a user specified point in time, and continue the game in the desired new direction the user elects to take, without having to finish the current game, and replay all the game decisions until that point when he chooses to depart to a new direction. This additional capability will enhance the use of the SMFS by allowing the user to analyze his previous decisions, by being able to locate, through causal tracing and analysis of the reported results, a point in time where he feels a faulty policy or decision was introduced that effected the overall outcome of the simulated project, change it, and continue to replay the simulation in a completely new direction.

Refinement work is needed in terms of naming conventions and accessibility to key initial parameter settings. Current naming convention makes difficult to read and understand the names of the variables displayed in the various graphs and reports generated by the interface. The interface should have provisions to allow the instructor, or authorized user, to modify the initial default parameters setting that controls the simulation behavior. This addition would greatly increase the utility of the interface by enabling the interface to be tailored to different circumstances that could effect the project.

The creation and use of a series of simulation scenarios controlled by Vensim interfaces, in areas such as motivation, goals achievement, and risk, offer the potential opportunity for a controllable “microworld” learning environment to be available to the students. This “microworld” will enable the user to probe and experience situations that have proven to be detrimental to the profession, and impossible to experiment within the real world. Respective interface development in the areas discussed above is an area open for future work.
Use of Vensim capabilities enable the creation of a laboratory environment capable of controlled experimentation and data collection for further research in topics associated with manager’s mental models and behavior interactions. The design of experiments to achieve research specific goals in these fields are areas open for future work using Vensim.
APPENDIX A.

SMFS VENSI MODEL FILE (VMF)

This appendix provides the equation listing for the model used in the SMFS Vensim application. For the purpose of the computer simulation, this listing is identified as the file Causal.vmf. This listing follows the Vensim modeling language. This language defines a model by use of a set of equations and expressions. The language modeling structure and syntax [Ventana, 1994, p. 20] is as follows:

```
Causal
**************...
| {Comments, notes, etc.} | Represent a Group
Profit = Revenue - Cost
~$/yr [0,1000]
~The profit of the company
:sup
|Comments Syntax
| Represent the basic equation
| Represent the Equation Units, Range [min, max]
| Equation comments, can have more than one line
| Optional Supplemental flag,
| Indicates the end of the equation.

A shorter form of an equation, were the dimensions, comments and optional flags are omitted, is as follows:

Profit = Revenues - Cost ~~|
```

The tilde (~) acts as separators and the bar (l) signals the end of an equation group. These delimiters are required in order to have a valid equation. Any spacing or line breaks are ignored within equations. However, line breaks are not allowed within variable names. Comments can be added anywhere in the model or equations by enclosing them in braces { }.
Causal

VERSION 1
October 1995

SUMMARY OF CHANGES

Dynamo Changes prior to conversion to VENSIM:
1. MDSWCH=0, TOTMD1=2359.60
2. SCSWCH=0, TDEV1=296.79
3. ADDED WCWFTP, WCWFTP.K=SWITCH(HIREDY+ASIMDY,TMPRMR,TMPRMR)
4. BASE5 / BASE MODEL: VERSION 5

VENSIM Conversion:
1. WORK FORCE SOUGHT (WFS) EQUATION WAS CHANGED TO INSERT THE CONSTANT (WFS1/ADMPPS) IN PLACE OF (WFNEED).
2. WORK FORCE AT BEGINNING OF DESIGN (WFSRT) EQUATION WAS CHANGED TO INSERT THE CONSTANT VALUE (WFS1) IN PLACE OF INITIAL(TEAMSZ*INUDST).
3. DAILY MANPOWER ALLOCATED FOR QA (DMPQA) EQUATION WAS CHANGED TO INSERT THE CONSTANT (FRMPQA) IN PLACE OF (AFMPQA).
4. SCHEDULED COMPLETION DATE (SCHCDT) EQUATION WAS CHANGED TO INSERT THE CONSTANT (PROJDR) IN PLACE OF (INTEG((TIME STEP*(INDCDT-SCHCDT)/SCHADT/TIME STEP,TDEV))).
5. TOTAL JOB SIZE IN MAN DATA (JSZMD) WAS CHANGED TO INSERT THE CONSTANT (TOTMD1) IN PLACE OF (Integ((IRDVT+IRTS+ARTJB), DEVMD+ TSTMD)).
6. TOTMD1 EQUATION IS LISTED TWICE. REMOVED EQUATION TOTMD1= 944.
7. MISSING DYNAMO EQUATIONS WERE INSERTED INTO THE VENSIM MODEL.
8. VALUES OF (SAVEPER) AND (TIME_STEP) WERE CHANGED TO READ 40 AND 0.5 RESPECTIVELY.
9. EQUATION (IPRSZ=(RJBDI)*(1-UNDESI)) WAS INSERTED IN THE VENSIM MODEL.
10. VARIABLES (TOTMD1), (PROJDR), (WFS1), (FRMPQA) WERE CHANGED TO "GAME" VARIABLES.
11. ALL NON-MACRO TIME_STEP CHANGED TO TIME STEP.
12. THE FOLLOWING EQUATIONS WERE ADDED TO SUPPORT SPECIFIC GRAPHS

- CUMMD TD: PROGRAMMING PERSON DAYS EXPENDED TO DATE
- CMDSDI KDSI: TOTAL KDSI COMPLETED
- CMERD KDSI: DEFECT DENSITY
- PJBSZT KDSI: ESTIMATED SYSTEM SIZE
- PROQAMD: PERIOD QA PERSON DAYS PER KDSI DEVELOPED IN PERIOD
- FRWFFEX PCT: PERCENT OF WORKFORCE THAT IS EXPERIENCED

13. EQUATIONS ADDED TO TEST CAUSALITY

\[ PRMD = \text{PERSON DAYS SPENT IN PERIOD} = \text{INTEG}((\text{TOTDMP} - (\text{PRMD} / \text{TIME STEP}))*\text{M PULSE}((\text{Time}, 1, \text{TIME STEP}, \text{TIME STEP}, 40)), 0.1) \]

\[ PRDPER = \text{PRODUCTIVITY IN 40 DAY PERIOD} = \text{PRTKDV/PRMD} \]

\[ \text{INSPRD} = \text{INSTANTANEOUS PRODUCTIVITY} = (\text{SDVRT*TIME STEP}) / \text{TOTDMP} \]

14. ADDED WFSINI AS INITIAL GAME VALUE. THIS PREVENTS THE CONSTANT UPDATE OF INITIAL DISPLAYED VALUES IN CONTROL SCREEN.

15. CHANGED WFS1 = GAME(2) TO WFS1 = GAME(WFSINI) TO AVOID CONSTANT UPDATE IN CONTROL SCREEN DISPLAY.

16. ADDED TDEVVINI = INTEGER(TDEV1) TO AVOID DISPLAY OF DECIMAL FRACTION IN CONTROL SCREEN DISPLAY.

17. ADDED TOTMD1INI AS INITIAL TOTAL ESTIMATED MAN DAYS. CHANGED TOTMD1INI = 2359.6 TO TOTMD1INI = 2360. THIS PREVENTS THE CONSTANT UPDATE OF INITIAL DISPLAYED VALUES IN CONTROL SCREEN.

18. CHANGED TOTMD1 = GAME(2359.6) TO TOTMD1 = GAME(TOTMD1INI) TO AVOID CONSTANT UPDATE IN CONTROL SCREEN DISPLAY.

19. CHANGED PROJDR = GAME(272) TO PROJDR = GAME(TDEVVINI) TO MATCH THE INITIAL DURATION REPORTED INITIALLY.
INITIAL VALUES FOR SIMULATION

THE REAL JOB SIZE = 64,000 DSI

FROM BOEHM PAGE 90:
DISTRIBUTION OF EFFORT BY PHASE IS:
    DESIGN (39%), PROGRAMMING (36%), INT TESTING (25%)

FROM BOEHM PAGE 64-65:
    EFFORT = 2.4*(KDSI)**1.05 = 190 MM = 190 * 19 = 3592 MAN-DAYS

DEVELOPMENT EFFORT(@75%) = 75%*3592 = 2695 MAN DAYS

GROSS DEV PRODUCTIVITY = 64,000/2695 = 24 DSI/MD

SCHEDULE = 2.5 * (MM)**.38 = 18 MONTHS = 348 DAYS

AVERAGE STAFF SIZE = 3592/348 = 10

GROSS PRODUCTIVITY INCORPORATES: DEV, FOR QA, & REWORKING
ASSUMING 25% OF EFFORT GOES INTO QA & REWORKING:
    25% OF 2695 MAN DAYS = 674 MAN DAYS

DEVELOPMENT PRODUCTIVITY = 64,000/(2695-674) = 31 DSI/MAN-DAY

ASSUME LOSSES IN PRODUCTIVITY = 50% THEREFORE:
POTENTIAL PRODUCTIVITY = 31 * 2 = APPROX 60 DSI/MD

DEFINE 1 TASK = 60 DSI

DSIPTK=60

~

DSI PER TASK

RJBDSI=64000

~

REAL JOB SIZE IN DSI

UNDEST=0.33

~ FRACTION

~ TASKS UNDERESTIMATION FRACTION
PJBDSSI=INITIAL(RJBDSI*(1-UNDEST))
~
~ PERCEIVED JOB SIZE IN DSI
|

TOTMD=INITIAL(MDSWCH*(((2.4*EXP(1.05*LN(PJBDSI/1000)))*19)*(1-UNDESM)) +
(1-MDSWCH)*TOTMD1)
~
~ TOTAL MAN DAYS
|

UNDESM=0
~ FRACTION
~ MAN-DAYS UNDERESTIMATION FRACTION
|

DEVMD=INITIAL(DEVPRT*TOTMD)
~
~ DEVELOPMENT MAN DAYS
|

MDSWCH=0
~
~ SWITCH 0 OR 1
|

TOTMD1INI=2360
~ MAN DAY
~ INITIAL ESTIMATED TOTAL MAN DAYS
|

TOTMD1=GAME(TOTMD1INI)
~ MAN DAY
~ TOTAL MAN DAYS
|

DEVPRT=0.8
~
~ PERCENT OF EFFORT ASSUMED NEEDED FOR DEVELOPMENT
|

TSTMD=INITIAL((1-DEVPRT)*TOTMD)
~
~ TESTING MAN DAYS
|

69
WFSTRT=WFS1
  ~ MEN
  ~ TEAM SIZE AT BEGINNING OF DESIGN

INUDST=0.5
  ~ DIMENSIONLESS
  ~ INITIAL UNDERSTAFFING FACTOR

TDEV=INITIAL(SCSWCH*((19*2.5*EXP(0.38*LN(TOTMD/19)))*SCHCOM)+
(1-SCSWCH)*TDEV1)
  ~ DAYS
  ~ TOTAL DEVELOPMENT TIME

SCHCOM=1
  ~ DIMENSIONLESS
  ~ SCHEDULE COMPRESSION FACTOR

SCSWCH=0
  ~
  ~ SWITCH 0 OR 1

TDEV1=296.79
  ~ DAYS
  ~ TIME TO DEVELOP

TDEVINI=INTEGER(TDEV1)
  ~ DAYS
  ~ INITIAL TIME TO DEVELOP

TEAMSZ=INITIAL((TOTMD/TDEV)/ADMPPS)
  ~ ~ 1
Control Subsystem

TIME STEP=0.5
   ~ DAYS
   ~ DT
 |

MAXLEN=1000
   ~ ~ |

SAVEPER=40
   ~ DAYS
   ~ CHANGED VALUE TO 40 DAYS VICE 10.
 |

FINAL TIME=IF THEN ELSE(PJBAYK >= 0.995,Time,MAXLEN)
   ~ ~ |

INITIAL TIME=0
   ~
   ~ ASSUMED WITH FINAL_TIME = LENGTH
 |

Equations For Causal Tracing

PRMD=INTEG((TOTDMP-(PRMD/TIME STEP)*M PULSE(Time,1,TIME STEP, TIME STEP,40)).0.1)
   ~ PERSON-DAY
   ~ PERSON DAYS SPENT IN PERIOD
 |

PRDPER=PRTkDV/PRMD
   ~ DSI PER PERSON-DAY
   ~ PRODUCTIVITY IN 40 DAY PERIOD
DYNAMO Project A Peculiar Equations

WFS2=WFS1
  ~ PEOPLE
  ~ WORK FORCE SOUGHT

FRMPQ1=FRMPQA
  ~ PERSONS
  ~ FRACTION OF MANPOWER FOR Q1

PRDFDS=PRERD/(MAX(PRTKDV/1000,0.01))
  ~ DEFECT/KDSI
  ~ PERIOD'S DEFECT DENSITY

CMDSI=CMTKDV*DSIPTK
  ~ TASKS
  ~ CUMULATIVE TASKS DEVELOPED

CRDVWF=TOTDMP-DMPQA
  ~ PERSONS
  ~ CURRENT DEVELOPMENT WORK FORCE

CRQAWF=(INTEGER(100*DMPQA))/100
  ~ PERSONS
  ~ CURRENT QA WORK FORCE

CRRWWF=DMPRW
  ~ PEOPLE
  ~ CURRENT REWORK WORK FORCE

PRCMPL=(CMDSI/PJBSZT)*100
  ~ Percent
  ~ PERCENT COMPLETE

RPPROD=PRDPRD*DSIPTK
  ~ DSI/MAN-DAY
  ~ REPORTED PRODUCTIVITY
FNERG=INTEG((TIME STEP*ERRGRT*IF THEN ELSE(PJBAWK >= 0.995,0,1))/TIME
STEP,0)
  ~ ERRORS
  ~  CUMULATIVE ERRORS GENERATED
 |

FNERD=INTEG((TIME STEP*ERRDRT*IF THEN ELSE(PJBAWK >= 0.995,0,1))/TIME
STEP,0)
  ~ ERRORS
  ~  CUMULATIVE ERRORS DETECTED
 |

FNERES=FNERG-FNERD
  ~ ERRORS
  ~  ERRORS THAT ESCAPED QA
 |

FNPRDT=100*FNERD/MAX(1,FNERG)
  ~ Percent
  ~  PERCENT DETECTED
 |

FNQAMD=INTEG((TIME STEP*DMPQA*IF THEN ELSE(PJBAWK >= 0.995,0,1))/
  TIME STEP,0)
  ~ MAN-DAYS
  ~  CUMULATIVE QA MAN-DAYS
 |

FNTRMD=INTEG((TIME STEP*DMPTMN*IF THEN ELSE(PJBAWK >= 0.995,0,1))/ TIME
STEP,0)
  ~
  ~  CUMULATIVE TRAINING MAN-DAYS
 |

FNRWMD=INTEG((TIME STEP*DMPRW*IF THEN ELSE(PJBAWK >= 0.995,0,1))/TIME
STEP,0)
  ~ MAN-DAYS
  ~  CUMULATIVE REWORK MAN-DAYS
 |

TM=Time
  ~ ~ |

PJBSZT=PJBSZ*DSIPTK
  ~ DSI
  ~  PERCEIVED JOB SIZE IN LINES OF CODE PERCEIVED JOB SIZE IN DSI
 |
IPRJSZ=INITIAL((RJBDSI)*(1-UNDEST))
    ~ DSI
    ~ INITIAL PROJECT SIZE IN DSI
|
FNCOST=INTEG((TIME STEP*TODMP*IF THEN ELSE(PJBAWK >= 0.995,0,1))/TIME STEP,0)
    ~ MAN-DAYS
    ~ FINAL COST IN MAN-DAYS
|
FNTIME=INTEG(IF THEN ELSE(PJBAWK >= 0.995,0,1),0)
    ~ ~ |
FNERR=INTEG(((IF THEN ELSE(PJBAWK >= 0.995,FNERR,ALESER)/TIME STEP)-
    FNERR/TIME STEP),0)
    ~ ~ |
PJBSZT KDSI=PJBSZT/1000
    ~ DSI
    ~ ESTIMATED SYSTEM SIZE (KDSI)
|
PRQAMD PERIOD=PRQAMD/((PRTKDV+0.01)/1000)
    ~ PERSON DAYS
    ~ QA PERSON DAYS PER KDSI DEVELOPED IN PERIOD
|
CUMMD TD=CUMMD-CMQAMD
    ~ PERSON DAYS
    ~ PROGRAMMING PERSON DAYS EXPENDED TO DATE
|
CMDSI KDSI=CMDSI/1000
    ~ TASKS
    ~ TOTAL KDSI COMPLETED
|
CMERD KDSI=CMERD*(1000/(CMDSI+0.01))
    ~ DEFECTS/KDSI
    ~ DEFECT DENSITY PER KDSI
|
FRWFEX PCT=FRWFEX*100
    ~ DIMENSIONLESS
    ~ FRACTION OF WF THAT IS EXPERIENCED
|

74
VENSIM MACRO FOR DYNAMO FUNCTION

This macro definition is for the PULSE function commonly used in DYNAMO, but not directly supported in Vensim.

NOTE: M_PULSE is equivalent to the DYNAMO function PULSE
       M_PULSE takes Time as an argument
       SAMPLE takes Time and TIME STEP as arguments

:MACRO: M_PULSE(time,height,width,first,intvl)

M_PULSE = height* PULSE(pulse_start,width)
       - height
       - Note that the pulse function in Vensim is different from the pulse function in DYNAMO. The argument time needs to be added to get repeated pulses - things may still be misaligned.

pulse_start = IF THEN ELSE(time < first+intvl,first,first + (QUANTUM((time-
                        first)/intvl,1))*intvl)
       - time
       - The pulse start moves forward over time. It makes no sense to have width > interval,
         and interval = 0 will cause an error.

:END OF MACRO:

:MACRO: SAMPLE(Time,TIME_STEP,INITIAL_TIME,X,INTVL,ISAM)

SAMPLE = SAMPLE IF TRUE(Time > next_time,X,X)
       - ~ ~

next_time = INTEG(IF THEN ELSE(Time>next_time,INTVL/TIME_STEP,0),
       INITIAL TIME-TIME STEP/2)
       - Time

:END OF MACRO:

PRQAMD = INTEG((DMPQA-(PRQAMD/TIME STEP)*M_PULSE(Time,1,TIME STEP,TIME
STEP,40)),0)
       - PERSON-DAYS
       - QA PERSON-DAYS IN PERIOD
PRERD=INTEG((ERRDRT-(PRERD/TIME STEP)*M PULSE(Time,1,TIME STEP, TIME STEP,40)),0)
~ ERRORS
~ DETECTED ERRORS DURING PERIOD

TMSTOP=IF THEN ELSE(PJBAWK*M PULSE(Time,1,TIME STEP,TIME STEP,40) >= 0.995,Time,MALLEN)
~
~ FINAL TIME

PRTKDV=INTEG((SDVRT*DSIPTK-(PRTKDV/TIME STEP)*M PULSE(Time,1,TIME STEP,TIME STEP,40)),0.1)

~
~ TASKS DEVELOPED DURING 40 DAY PERIOD
Human Resource Management Subsystem

WFNEW = INT((HIRERT - ASIMRT - NEWTRR), 0)
  ~ PEOPLE
  ~ NEW WORKFORCE

HIRERT = MAX(0, WFGAP / HIREDY)
  ~ PEOPLE/DAY
  ~ HIRING RATE

HIREDY = 40
  ~ DAYS
  ~ HIRING DELAY

WFGAP = WFS - TOTWF
  ~ PEOPLE
  ~ WORKFORCE GAP

NEWTRR = MIN(TRNFRT, WFNEW / TIME STEP)
  ~ PEOPLE/DAY
  ~ NEW EMPLOYEES TRANSFER RATE OUT

TRNFRT = MAX(0, - WFGAP / TRNSDY)
  ~ PEOPLE/DAY
  ~ TRANSFER RATE OF PEOPLE OUT OF PROJECT

TRNSDY = 10
  ~ DAYS
  ~ TIME DELAY TO TRANSFER PEOPLE OUT

ASIMRT = WFNEW / ASIMDY
  ~ PEOPLE/DAY
  ~ ASSIMILATION RATE OF NEW EMPLOYEES

ASIMDY = 80
  ~ DAYS
  ~ AVERAGE ASSIMILATION DELAY
DMPTRN = WFNEW * TRPNHR
   ~ MAN DAYS/DAY
   ~ DAILY MANPOWER FOR TRAINING

TRPNHR = 0.2
   ~ DIMENSIONLESS
   ~ NUMBER OF TRAINERS PER NEW EMPLOYEE

CMTRMD = INTEG(DMPTRN, 0)
   ~
   ~ CUMULATIVE TRAINING MAN-DAYS

WFEXP = INTEG((ASIMRT - EXPTRR - QUITRT), WFSRT)
   ~ PEOPLE
   ~ EXPERIENCED WORKFORCE INITIAL VALUE OF EXPERIENCED WORKFORCE LEVEL

EXPTRR = MIN(WFEXP / TIME STEP, TRNFRT - NEWTRR)
   ~ PEOPLE/DAY
   ~ EXPERIENCED EMPLOYEES TRANSFER RATE

QUITRT = WFEXP / AVEMPT
   ~ PEOPLE/DAY
   ~ EXPERIENCED EMPLOYEES QUIT RATE

AVEMPT = 673
   ~ DAYS
   ~ AVERAGE EMPLOYMENT TIME

FTEXWF = WFEXP * ADMPPS
   ~ MEN
   ~ FULL-TIME-EQUIVALENT EXPERIENCED WF

CELNWH = FTEXWF * MNHPXS
   ~ MEN
   ~ CEILING ON NEW HIREES
MNHPXS=3
  ~ MEN/MEN
  ~ MOST NEW HIREES PER EXPERIENCED STAFF

CELTWF=CELNWH+WFEXP
  ~ PEOPLE
  ~ CEILING ON TOTAL WORKFORCE

WFS=MIN(CELTWF,WFS1/ADMPPS)
  ~ PEOPLE
  ~ WF SOUGHT

WFSINI=2
  ~ PEOPLE
  ~ INITIAL VALUE OF STAFFING LEVEL

WFS1=GAME(WFSINI)
  ~ PEOPLE
  ~ TOTAL REQUESTED STAFFING LEVEL

TOTWF=WFNEW+WFEXP
  ~ PEOPLE
  ~ TOTAL WF LEVEL

FTEQWF=TOTWF*ADMPPS
  ~ PERSONS
  ~ FULL TIME EQUIVALENT WF

FRWFEX=(INTEGER((WFEXP/TOTWF)*10000))/10000
  ~ DIMENSIONLESS
  ~ FRACTION OF WF THAT IS EXPERIENCED
Software Production Subsystem

{(A) MANPOWER ALLOCATION SECTOR}

TOTDMP=TOTWF*ADMPPS
  ~ MAN-DAYS/DAY
  ~ TOTAL DAILY MANPOWER
|

ADMPPS=1
  ~ DAY/DAY
  ~ AVERAGE DAILY MANPOWER PER STAFF
|

CUMMD=INTEG(TOTDMP,0.0001)
  ~ MAN DAYS
  ~ CUMULATIVE MAN-DAYS EXPENDED
|

DMPATR=TOTDMP-DMPTRN
  ~ MAN-DAYS/DAY
  ~ DAILY MANPOWER AVAILABLE AFTER TRAINING
|

AFMPQA=ACTIVE INITIAL(PFMPQA*(1+ADJQA),PFMPQA)
  ~ DIMENSIONLESS
  ~ ACTUAL FRACTION OF MANPOWER FOR QA
|

QO=0
  ~ QUALITY OBJECTIVE ... NORMAL QO = 0
|

PFMPQA=TPFMQA(PJBAWK)*(1+QO/100)
  ~ DIMENSIONLESS
  ~ PLANNED FRACTION OF MANPOWER FOR QA
|

TPFMQA(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,0.15,0.15,0.15,0.15,0.15,0.15,0.15,0.15,0.15,0.15,0.15)
  ~ ~ |

ADJQA=TADJQA(SCHPR)
  ~ Percent
  ~ Percent ADJUSTMENT IN PFMPQA
|

80
TADJQA(0.0.1.0.2.0.3.0.4.0.5,0,-0.025,-0.15,-0.35,-0.475,-0.5)

~ ~ ~

DMPQA=MIN(((FRMPQA/100)*TOTDMP),0.9*DMPATR)

~ MAN-DAYS/DAY
~ DAILY MANPOWER ALLOCATED FOR QA

FRMPQA=GAME(10)

~ PERSONS
~ FRACTION OF MANPOWER ALLOCATED FOR QA

CMQAMD=INTEG(DMPQA,0)

~ MAN-DAYS
~ CUMULATIVE QA MAN-DAYS

DMPSWP=DMPATR-DMPQA

~ MAN-DAYS/DAY
~ DAILY MANPOWER FOR SOFTWARE PRODUCTION

DESECR=ACTIVE INITIAL(DTCERR/DESRWD,0)

~ ERRORS/DAY
~ DESIRED ERROR CORRECTION RATE

DESRWD=15

~ DAYS
~ DESIRED REWORK DELAY

DMPRW=ACTIVE INITIAL(MIN((DESECR*PRWMPE),DMPSWP),0)

~ MAN-DAYS/DAY
~ DAILY MANPOWER ALLOCATED FOR REWORK

PRWMPE=INTEG((RWMPPE-PRWMPE)/TARMPE,0.5)

~ MAN-DAYS/ERROR
~ PERCEIVED REWORK MANPOWER NEEDED PER ERROR

TARMPE=10

~ DAYS
~ TIME TO ADJUST PRWMPE
CMRWMD=INTEG(DMPRW,0)
    ~ MAN DAYS
    ~ CUMULATIVE REWORK MAN-DAYS
 |

DMPDVT=DMPSWP-DMPRW
    ~ MAN-DAYS/DAYS
    ~ DAILY MANPOWER FOR DEVELOPMENT/ TESTING
 |

CMDVMD=INTEG((TIME STEP*DMPDVT*(1-FREFTS))/TIME STEP,0)
    ~ MAN DAYS
    ~ CUMULATIVE DEVELOPMENT MAN-DAYS
 |

{(B) SOFTWARE DEVELOPMENT SECTOR}

SDVRT=ACTIVE INITIAL(MIN((DMPSDV*SDVPRD),TSDKRM/TIME STEP),0)
    ~ TASKS/DAY
    ~ SOFTWARE DEVELOPMENT RATE
 |

DMPSDV=DMPDVT*(1-FREFTS)
    ~ MAN-DAYS/DAY
    ~ DAILY MANPOWER FOR SOFTWARE DEVELOPMENT
 |

INSPRD=(SDVRT*TIME STEP)/TOTDMP
    ~ INSTANTANEOUS PRODUCTIVITY
 |

FREFTS=TFEFTS(TSKPRM/PJBSZ)
    ~ DIMENSIONLESS
    ~ FRACTION OF EFFORT FOR SYSTEM TESTING
 |

TFEFTS(0.0,0.04,0.08,0.12,0.16,0.2,1,0.5,0.28,0.15,0.05,0)
    ~ ~
 |

SDVPRD=POTPRD*MPDMCL
    ~ TASKS/MAN-DAY
    ~ SOFTWARE DEVELOPMENT PRODUCTIVITY
 |

POTPRD=ANPPRD*MPPTPD
    ~ TASKS/MAN-DAY
    ~ POTENTIAL PRODUCTIVITY
 |

82
ANPPRD = FRWFEX*NPWPEX+(1-FRWFEX)*NPWPNE
   ~ TASKS/MAN-DAY
   ~ AVERAGE NOMINAL POTENTIAL PRODUCTIVITY

NPWPEX = 1
   ~ TSK/M-D
   ~ NOMINAL POTENTIAL PRODUCTIVITY OF EXP EMPLOYEE

NPWPNE = 0.5
   ~ TSK/M-D
   ~ NOMINAL POTENTIAL PROD OF NEW EMPL.

MPPTPD = TMPTPD(PJBAWK)
   ~ DIMENSIONLESS
   ~MULTIPLIER TO POTENTIAL PRODUCTIVITY DUE TO LEARNING

TMPTPD(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,1,1.0125,1.0325,1.055,1.09,1.15,1.2,1.22, 1.245,1.25,1.25)
   ~ ~ ~

MPDMCL = AFMDPJ*(1-COMMOH)
   ~ DIMENSIONLESS
   ~MULTIPLIER TO PRODUCTIVITY DUE TO MOTIVATION & COMM LOSSES

COMMOH = TCOMOH(TOTWF)
   ~ DIMENSIONLESS
   ~ COMMUNICATION OVERHEAD

TCOMOH(0.5,10,15,20,25,30,0,0.015,0.06,0.135,0.24,0.375,0.54)
   ~ ~ ~

NFMDPJ = 0.6
   ~ DIMENSIONLESS
   ~ NOMINAL FRACTION OF A MAN-DAY ON PROJECT

AFMDPJ = INTEG(WRADJR,NFMDPJ)
   ~ DIMENSIONLESS
   ~ ACTUAL FRACTION OF A MAN-DAY ON PROJECT
WRADJR=(WKRTS-AFMDPJ)/WKRADY
  ~ 1/DAY
  ~ WORK RATE ADJUSTMENT RATE

WKRADY=NWRADY*EWKRTS
  ~ DAYS
  ~ WORK RATE ADJUSTMENT DELAY

NWRADY=TNWRAD(TIMERM)
  ~ DAYS
  ~ NORMAL WORK RATE ADJUSTMENT DELAY

TNWRAD(0.5,10,15,20,25,30,2.3.5,5,6.5,8,9.5,10)
  ~ ~ /

EWKRTS=IF THEN ELSE(WKRTS >= AFMDPJ,1,0.75)
  ~ DIMENSIONLESS
  ~ EFFECT OF WORK RATE SOUGHT

WKRTS=(1+PBWKRS)*NFMDPJ
  ~ DIMENSIONLESS
  ~ WORK RATE SOUGHT

MAXMHR=INITIAL(1)
  ~ DIMENSIONLESS
  ~ MAXIMUM BOOST IN MAN-HOURS

PBWKRS=IF THEN ELSE(PMDSHR >= 0,(MDHDL/(FTEQWF*(OVWDT+0.0001))),
  (MDHDL/(TMDPSN-MDHDL+0.0001)))
  ~ Percent
  ~ Percent BOOST IN WORK RATE SOUGHT

MDHDL=IF THEN ELSE(PMDSHR >= 0,MIN(MAXSHR,PMDSHR),-EXSABS)*CTRLSW
  ~ MAN-DAYS
  ~ MAN-DAYS THAT WILL BE HANDLED OR ABSORBED

CTRLSW=1
  ~ Zero or One
  ~ CONTROL SWITCH ... ALLOWS US TO TEST POLICY OF NO OVERWORK

84
EXSABS=MAX(0,(TEXABS(TMPSN/MDRM)*MDRM-TMPSN))
    ~ MAN-DAYS
    ~ MAN-DAY EXCUSES THAT WILL BE ABSTRACTED

TEXABS(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,0,0.2,0.4,0.55,0.7,0.8,0.9,0.95,1,1,1)
    ~ ~ ~

MAXSHR=(OVWDT*(FTEQWF*MAXMHK)*WTOVWK)
    ~ MAN-DAYS
    ~ MAXIMUM SHORTAGE IN MAN-DAYS THAT CAN BE HANDLED

WTOVWK=IF THEN ELSE(Time >= BRKDTM+RLXTC,1,0)
    ~ Zero or One
    ~ WILLINGNESS TO OVERWORK

BRKDTM=INTEG((TIME STEP/(TIME STEP))*(MAX(BRKDTM,IF THEN ELSE
    (OVWDT = 0, (TIME + TIME STEP,0)) - BRKDTM)/TIME STEP,-1)
    ~ ~ ~ TIME OF LAST EXHAUSTION BREAKDOWN

RLXTC=INTEG(IF THEN ELSE((EXHLEV/MXEXH) >= 0.1,1,-RLXTC/TIME STEP)-
    ((1/TIME STEP)*RLXTC*IF THEN ELSE(OVWDT = 0,1,0))),0)
    ~ ~ ~ VARIABLE THAT CONTROLS TIME TO DE-EXHAUST

OVWDT=NOVWDT*MODTEX
    ~ DAYS
    ~ OVERWORK DURATION THRESHOLD

NOVWDT=TNOWDT(TIMERM)
    ~ DAYS
    ~ NOMINAL OVERWORK DURATION THRESHOLD

TNOWDT(0,10,20,30,40,50,0,10,20,30,40,50)
    ~ ~ ~

MODTEX=TMODEX(EXHLEV/MXEXH)
    ~ DIMENSIONLESS
    ~ EFFECT OF EXHAUSTION ON OVERWORK DURATION THRESHOLD

85
TMODEX=(0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.1,1.0,9.0,8.0,7.0,6.0,5.0,4.0,3.0,2.0,1.0)
    ~ ~ ~
EXHLEV=INTEG((RIEXHL-RDEXHL),0)
    ~ EXHAUST UNITS
    ~ EXHAUSTION LEVEL

RIEXHL=TRIXHL((1-AFMDPJ)/(1-NFMDPJ))
    ~ EXHAUST UNITS/DAY
    ~ RATE OF INCREASE IN EXHAUSTION LEVEL

TRIXHL=(-0.5,-0.4,-0.3,-0.2,-0.1,1.745058e-009,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.25,
        2.2,1.9,1.6,1.3,1.15,0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.0)
    ~ ~ ~
RDEXHL=IF THEN ELSE(0 >= RIXHL,EXHLEV/EXHDDY,0)
    ~ EXHAUST UNITS/DAY
    ~ RATE OF DEPLETION IN EXHAUSTION LEVEL

EXHDDY=20
    ~ DAYS
    ~ EXHAUSTION DEPLETION DELAY TIME

MXEXHT=50
    ~ EXHAUST UNITS
    ~ MAXIMUM TOLERABLE EXHAUSTION

{(C) QUALITY ASSURANCE AND REWORK SECTOR}

QART=DELAY3(SDVRT,AQADLY)
    ~ TASKS/DAY
    ~ FOR QA RATE

TSKWK=INTEG((SDVRT-QART),0)
    ~ TASKS
    ~ TASKS WORKED

AQADLY=10
    ~ DAYS
    ~ AVERAGE DELAY FOR QA
CUMTQA=INTEG((QART-TSRATE),0)
   ~ TASKS
   ~ CUMULATIVE TASKS QA'ED

ANERPT=MAX(PTDTER/(TSKWK+0.0001),0)
   ~ ERRORS/TASK
   ~ AVERAGE # OF ERRORS PER TASK

QAMPNE=NQAMPE*(1/MPDMCL)*MDEFED
   ~ MAN-DAYS/ERROR
   ~ QA MANPOWER NEEDED TO DETECT AVERAGE ERROR

NQAMPE=TQQAPE(PJBAWK)
   ~ MAN-DAYS/ERROR
   ~ NOMINAL QA MANPOWER NEEDED TO DETECT AVERAGE ERROR

TNQAPE(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,4.0.4,0.39,0.375,0.35,0.3,0.25,0.225,0.21,
   0.2,0.2)
   ~ ~ ~

MDEFED=TMDGED(ERRDSY)
   ~ DIMENSIONLESS
   ~ MULTIPLIER TO DETECTION EFFORT DUE TO ERROR DENSITY

TMDGED(0,1,2,3,4,5,6,7,8,9,10,50,36,26,17.5,10,4,1.75,1,2,1,1,1) ~ ~ ~
ERRDSY=ANERPT*1000/DSIPTK
   ~ ERRORS/KDSI
   ~ ERROR DENSITY

PERDRT=DMPQA/QAMPNE
   ~ ERRORS/DAY
   ~ POTENTIAL ERROR DETECTION RATE

ERRDRT=MIN(PERDRT,PTDTER/TIME STEP)
   ~ ERRORS/DAY
   ~ ERROR DETECTION RATE

CMERD=INTEG(ERRDRT,0)
   ~ ERRORS
   ~ CUMULATIVE ERRORS DETECTED
PRCTDT=100*CMERD/(CUMERG+0.001)
  ~ Percent
  ~ PERCENT ERRORS DETECTED

ERRSRT=QART*ANERT
  ~ ERRORS/DAY
  ~ ERROR ESCAPE RATE

CMERES=INTEG(ERRSRT,0)
  ~ ERRORS
  ~ CUMULATIVE ERRORS THAT ESCAPED

PTDTER=INTEG((ERRGRT-ERRDRT-ERRSRT),0)
  ~ ERRORS
  ~ POTENTIALLY DETECTABLE ERRORS

ERRGRT=SDVRT*ERRPTK
  ~ ERRORS/DAY
  ~ ERROR GENERATION RATE

ERRPTK=NERPTK*MERGSP*MERGW
  ~ ERRORS/TASK
  ~ ERRORS PER TASK

NERPTK=NERPK*DSIPTK/1000
  ~ ERRORS/TASK
  ~ NOMINAL # OF ERRORS COMMITTED PER TASK

NERPK=TNERPK(PJBAWK)
  ~ ERRORS/KDSI
  ~ NOMINAL # OF ERRORS COMMITTED PER KDSI

TNERPK(0,0.2,0.4,0.6,0.8,1,2,2.5,23,23.86,21.59,15.9,13.6,12.5)
  ~ ~  ~

MERGSP=TMERGE(SCHPR)
  ~ DIMENSIONLESS
  ~ MULTIPLIER TO ERROR GENERATION DUE TO SCHEDULE PRESSURE
TMEGSP(0.4, 0.2, 0.6, 0.8, 1.0, 0.9, 1.0, 0.94, 1.1, 1.05, 1.14, 1.24, 1.36, 1.5)
    ~   ~    ~

MERGWM = TMEGWM(FRFEX)
    ~ DIMENSIONLESS
    ~ MULTIPLIER TO ERROR GENERATION DUE TO WORKFORCE MIX
    ~

TMEGWM(0.0, 0.2, 0.4, 0.6, 0.8, 1.2, 1.8, 1.6, 1.4, 1.2, 1)
    ~   ~    ~

CUMERG = INTEG(ERRGRT, 0)
    ~ ERRORS
    ~ CUMULATIVE ERRORS GENERATED DIRECTLY DURING WORKING

DTCERR = INTEG((ERRDRT - RWRATE), 0)
    ~ ERRORS
    ~ DETECTED ERRORS

RWRATE = DMPRW/RWMPPE
    ~ ERRORS/DAY
    ~ REWORK RATE

RWMPPE = NRWMPE/MPDMCL
    ~ MAN-DAYS/ERROR
    ~ REWORK MANPOWER NEEDED PER ERROR

NRWMPE = TNRWME(PJBAWK)
    ~ MAN-DAYS/ERROR
    ~ NOMINAL REWORK MANPOWER NEEDED PER ERROR

TNRWME(0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 0.6, 0.575, 0.5, 0.4, 0.325, 0.3)
    ~   ~    ~

CMRWED = INTEG(RWRATE, 0)
    ~ ERRORS
    ~ CUMULATIVE REWORKED ERRORS DURING DEVELOPMENT

{(D) SYSTEM TESTING SECTOR}

UDAVER = INTEG((AEGRT + AERGRT - AERRRT - DCRTAE), 0)
    ~ ERRORS
    ~ UNDETECTED ACTIVE ERRORS
AEGRT = (ERRSRT + BDFXGR) * FRAERR
  ~ ERRORS/DAY
  ~ ACTIVE ERRORS GENERATION RATE

BDFXGR = RWRATE * PBADFX
  ~ ERRORS/DAY
  ~ BAD FIXES GENERATE RATE

PBADFX = 0.075
  ~ FRACTION
  ~ PERCENT BAD FIXES

FRAERR = TFRAER(PJBAWK)
  ~ DIMENSIONLESS
  ~ FRACTION OF ESCAPING ERRORS THAT WILL BE ACTIVE

TFRAER(0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 0.95, 0.85, 0.5, 0.2, 0.075, 0.0)
  ~ ~

AERGRT = SDVRT * SMOOTH(AERRDS, TSAEDS) * MAERED
  ~ ERRORS/DAY
  ~ ACTIVE ERRORS REGENERATION RATE

MAERED = TMERED(SMOOTH(AERRDS * 1000 / DSIPTK, TSAEDS))
  ~ DIMENSIONLESS
  ~ MULTIPLIER TO ACTIVE ERROR REGENERATION DUE TO ERROR DENSITY

TMERED(0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.325, 1.45, 1.6, 2.2, 2.5, 3.25, 4.35, 6)
  ~ ~

TSAEDS = 40
  ~ DAYS
  ~ TIME TO SMOOTH ACTIVE ERROR DENSITY (AERRDS)

AERRDS = UDAVER / (CUMTQA + 0.1)
  ~ ERRORS/TASK
  ~ ACTIVE ERROR DENSITY

90
AERRRT=UDAVER*AERRFR
   - ERRORS/DAY
   - ACTIVE ERRORS RETIRING RATE

AERRFR=TERMFRT(PJBAWK)
   - 1/DAYS
   - ACTIVE ERRORS RETIRING FRACTION

TERMFRT(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,0,0,0,0,0,0.01,0.02,0.03,0.04,0.1,0.3,1)
   ~ ~

DCRTAE=MIN(TSRATE*AERRDS,UDAVER/TIME STEP)
   - ERRORS/DAY
   - DETECTION/CORRECTION RATE OF ACTIVE ERRORS

UDPVER=INTEG((PEGRT+AERRRT-DCRTPE),0)
   - ERRORS
   - UNDETECTED PASSIVE ERRORS

PEGRT=(ERRSRT+BDFXGR)*(1-FRAERR)
   - ERRORS/DAY
   - PASSIVE ERRORS GENERATION RATE

DCRTPE=MIN(TSRATE*PERRDS,UDPVER/TIME STEP)
   - ERRORS/DAY
   - DETECT/CORRECT RATE OF PASSIVE ERRORS

CMRWET=INTEG((DCRTPE+DCRTAE),0)
   - ERRORS
   - CUMULATIVE ERRORS REWORKED IN TESTING PHASE

ALESER=UDAVER+UDPVER+CMRWET
   - ERRORS
   - ALL ERRORS THAT ESCAPED AND WERE GENERATED

DMPTST=DMPDVT*FREFTS
   - MAN DAYS/DAY
   - DAILY MANPOWER FOR TESTING
CMTSMD = INTEG(DMPTST, 0)
   ~ MAN DAYS
   ~ CUMULATIVE TESTING MAN-DAYS

TSRATE = MIN(CUMTQA / TIME STEP, DMPTST / TMPNPT)
   ~ TASKS/DAY
   ~ TESTING RATE

TMPNPT = (TSTOVH * (DSIPTK / 1000) + TMPNPE * (PERRDS + AERRDS)) / MPDMCL
   ~ MAN-DAYS/TASK
   ~ TESTING MANPOWER NEEDED PER TASK

TSTOVH = 1
   ~ MAN-DAYS/KDSI
   ~ TESTING EFFORT OVERHEAD

TMPNPE = 0.15
   ~ MAN-DAY/ERROR
   ~ TESTING MANPOWER NEEDED PER ERROR

PTKTST = CUMTKT / PJBSZ
   ~ PERCENT
   ~ PERCENT OF TASKS TESTED

PERRDS = UDPVER / (CUMTQA + 0.0001)
   ~ ERRORS/TASK
   ~ PASSIVE ERROR DENSITY

CUMTKT = INTEG(TSRATE, 0)
   ~ TASKS
   ~ CUMULATIVE TASKS TESTED

ALLERR = PTDTER + DTCERR + CMRWED + UDAVER + UDPVER + CMRWET
   ~ ERRORS
   ~ ALL ERRORS

ALLRWK = CMRWED + CMRWET
   ~ ERRORS
   ~ ALL ERRORS REWORKED ... IN DEVELOPMENT AND TESTING
CMTKDV=INTEG(SDVRT,0)
  ~ TASKS
  ~ CUMULATIVE TASKS DEVELOPED
 |

PJBWK=CMTKDV/RJBSZ
  ~ PERCENT
  ~ PERCENT OF JOB ACTUALLY WORKED
 |

PJPRD=TSKPRM/(MDPRNT+0.1)
  ~ TASKS/MAN-DAY
  ~ PROJECTED DEVELOPMENT PRODUCTIVITY
 |

MDPRNT=MAX(0,MDRM-MDPNRW-MDPNTS)
  ~ MAN-DAYS
  ~ MAN DAYS PERCEIVED REMAINING FOR NEW TASKS
 |

MDPNRW=DTCERR*PRWMPE
  ~ MAN DAYS PERCEIVED NEEDED FOR REWORKING ALREADY DETECTED ERRORS (MD
 |

ASSPRD=PJPRD*WTPJDP+PRDPD*(1-WTPJDP)
  ~ TASKS/MAN-DAY
  ~ ASSUMED PRODUCTIVITY
 |

PRDPD=CMTKDV/(CUMMD-CMTSMD)
  ~ TASKS/MAN-DAY
  ~ PERCEIVED DEVELOPMENT PRODUCTIVITY
 |

WTPJDP=MPWDEV*MPWREX
  ~ DIMENSIONLESS
  ~ WEIGHT TO PROJECTED DEVELOPMENT PRODUCTIVITY
 |

MPWDEV=TMPDEV(PJBPWK/100)
  ~ DIMENSIONLESS
  ~ MULTIPLIER TO PRODUCTIVITY WEIGHT DUE TO DEVELOPMENT
 |

TMPDEV(0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,1,1,1,1,1,0.975,0.9,0.75,0.5,0)
  ~ ~ 1
MPWREX = TMPREX((1-MDPRNT/(JBSZMD-TSSZMD)))
   ~ DIMENSIONLESS
   ~ MULTIPLIER TO PRODUCTIVITY WEIGHT DUE TO RESOURCE EXPENDITURE

TMPREX(0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1, 1, 1, 1, 1, 1, 0.9, 0.9, 0.9, 0.75, 0.75, 0.5, 0.0)
   ~ ~ ~

MDPNNT = TSKPRM/ASSPRD
   ~ MAN-DAYS
   ~ MAN DAYS PERCEIVED STILL NEEDED FOR NEW TASKS

TMDPSN = MDPNNT + MDPNTS + MDPNRW
   ~ MAN-DAYS
   ~ TOTAL MAN DAYS PERCEIVED STILL NEEDED

MDPNTS = TSTPRM/PRTPRD
   ~ MAN-DAYS
   ~ MAN DAYS PERCEIVED STILL NEEDED FOR TESTING

TSTPRM = PJBSZ-CUMTKT
   ~ TASKS
   ~ TASKS REMAINING TO BE TESTED

PRTPRD = SMOOTH((IF THEN ELSE(0 >= CUMTKT, PLTSPD, ACTSPD)), TSTSPD)
   ~ TASKS/MAN-DAY
   ~ PERCEIVED TESTING PRODUCTIVITY

TSTSPD = 50
   ~ DAYS
   ~ TIME TO SMOOTH TESTING PRODUCTIVITY

PLTSPD = PJBSZ/TSSZMD
   ~ TASKS/MAN-DAY
   ~ PLANNED TESTING PRODUCTIVITY

ACTSPD = CUMTKT/(CMTSMD+0.001)
   ~ TASKS/MAN-DAY
   ~ ACTUAL TESTING PRODUCTIVITY
PMDSHR=TMDSRN-MDRT
  ~ MAN DAYS
  ~ PERCEIVED SHORTAGE IN MAN DAYS

SHRRPT=PMDSHR-MDHDL
  ~ MAN DAYS
  ~ SHORTAGE REPORTED

MDRPTN=MDRM+SHRRPT
  ~ MAN DAYS
  ~ MAN DAYS REPORTED STILL NEEDED

SCHPR=(TMDSRN-MDRM)/MDRM
  ~ DIMENSIONLESS
  ~ SCHEDULE PRESSURE

PTRPTC=ACTIVE INITIAL(SMOOTH((100-(MDRPTN/JBSZMD))*100,RPTDLY),0)
  ~ PERCENT
  ~ PERCENT OF TASKS REPORTED COMPLETE

RPTDLY=10
  ~ DAYS
  ~ REPORTING DELAY

PDEVRC=ACTIVE INITIAL(SMOOTH(MAX(100-(MDRPTN-MDPNTS)/(JBSZMD-TSSZMD)*100),PDEVRC),RPTDLY),0)
  ~ PERCENT DEVELOPMENT PERCEIVED COMPLETE %

UNDJTK=INTEG((-RTDSTK),RJBSZ-PJBSZ)
  ~ TASKS
  ~ UNDISCOVERED JOB TASKS

RJBSZ=INITIAL(RJBDSI/DSIPTK)
  ~ TASKS
  ~ REAL JOB SIZE IN TASKS

RTDSTK=UNDJTK*PUTDPTD/100
  ~ TASKS/DAY
  ~ RATE OF DISCOVERING TASKS
PUTDPD=TPUTDD(PJBPKW)
   ~ 1/DAY
   ~ PERCENT OF UNDISCOVERED TASKS DISCOVERED PER DAY
   
TPUTDD(0,20,40,60,80,100,0,0.4,2.5,5,10,100)
   ~ ~

PJBPKW=(CMTKDV/PJBSZ)*100
   ~ PERCENT
   ~ PERCENT OF JOB PERCEIVED WORKED
   
RTINCT=DELAY3(RTDSTK,DLINCT)
   ~ TASKS/DAY
   ~ RATE OF INCORPORATING DISCOVERED TASKS INTO PROJECT
   
TKDSCV=INTEG((TIME STEP*(1/TIME STEP)*((-1)*TKDSCV+MAX((TKDSCV+TIME
STEP*(RTDSTK-RTINCT),0))/TIME STEP,0)))/TIME STEP,0)
   ~ TASKS
   ~ TASKS DISCOVERED
   
DLINCT=10
   ~ DAYS
   ~ AVERAGE DELAY IN INCORPORATING DISCOVERED TASKS
   
PJBSZ=INTEG(RTINCT,PJBDSI/DSIPTK)
   ~ TASKS
   ~ CURRENTLY PERCEIVED JOB SIZE
   
TSKPRM=PJBSZ-CMTKDV
   ~ TASKS
   ~ NEW TASKS PERCEIVED REMAINING
   
PSZDCT=TKDSCV/ASSPRD
   ~ MAN-DAYS
   ~ PERCEIVED SIZE OF DISCOVERED TASKS IN MAN DAYS
   
RSZDCT=PSZDCT/(MDPRNT+0.0001)
   ~ DIMENSIONLESS
   ~ RELATIVE SIZE OF DISCOVERED TASKS
   
96
FADHWO=TFAHWO(RSZDCT/(MSZTWO+0.001))
   ~
   ~ FRACTION OF ADDITIONAL TASKS ADDING TO MAN-DAYS

TFAHWO(0,0,2,0,4,0,6,0,8,1,1,2,1,4,1,6,1,8,2,0,0,0,0,0,0,0,7,0,9,0,975,1,1)
   ~  ~
MSZTWO=0.01
~
~ MAXIMUM RELATIVE SIZE OF ADDITIONS TOLERATED W/O ADDING TO PROJECT

IRDVDT=(RTNCT/ASSPRD)*(FADHWO)
   ~
   ~ RATE OF INCREASE IN DEVELOPMENT MAN-DAYS DUE TO DISCOVERED TASKS

TSSZMD=INTEG((IRTSDT+(1/TIME STEP)*ARTJBM*IF THEN ELSE(FREFTS >=
   0,9,1,0)),TSTMD)
   ~
   ~ PLANNED TESTING SIZE IN MAN-DAYS ... BEFORE WE START TESTING

IRTSRT=(RTNCT/PRTPRD)*(FADHWO)
   ~ MD/D
   ~ RATE OF INCREASE IN TESTING MAN DAYS DUE TO DISCOVERED TASKS

JBSZMD=TOTMD1
   ~ MAN DAYS
   ~ TOTAL JOB SIZE IN MAN DAYS

ARTJBM=(MDRPTN+CUMMD-JBSZMD)/DAJBMD
   ~ MAN-DAYS/DAY
   ~ RATE OF ADJUSTING THE JOB SIZE IN MAN-DAYS

DAJBMD=TDAJMD(TIMERM)
   ~ DAYS
   ~ DELAY IN ADJUSTING JOB'S SIZE IN MAN DAYS

TDAJMD(0,20,0,5,3) ~ ~

MDRM=MAX(0.0001,JBSZMD-CUMMD) ~ ~
Planning and Man Days remaining

\[
\text{TIMEPR} = \text{MDRM}/(\text{WF} \times \text{ADMPPS})
\]

\[
\sim \text{DAYS}
\]

\[
\sim \text{TIME PERCEIVED STILL REQUIRED}
\]

\[
\text{INDCDT} = \text{Time} + \text{TIMEPR}
\]

\[
\sim \text{INDICATED COMPLETION DATE}
\]

\[
\text{SCHCDT} = \text{PROJDR}
\]

\[
\sim \text{SCHEDULE COMPLETION DATE}
\]

\[
\text{PROJDR} = \text{GAME}(\text{TDEVINI})
\]

\[
\sim \text{DAYS}
\]

\[
\sim \text{PROJECT DURATION}
\]

\[
\text{SCHADT} = \text{TSHADT}(\text{TIMERM})
\]

\[
\sim \text{DAYS}
\]

\[
\sim \text{SCHEDULE ADJUSTMENT TIME}
\]

\[
\text{TSHADT} = (0.5,0.5,5)
\]

\[
\sim \sim
\]

\[
\text{TIMERM} = \text{MAX}(\text{SCHCDT} - \text{Time}, 0)
\]

\[
\sim \text{DAYS}
\]

\[
\sim \text{TIME REMAINING}
\]

\[
\text{WFINDC} = \text{MDRM}/(\text{TIMERM} + 0.001)/\text{ADMPPS}
\]

\[
\sim \text{PEOPLE}
\]

\[
\sim \text{INDICATED WORKFORCE}
\]

\[
\text{WFNEED} = \text{MIN}((\text{WCWF} \times \text{WFINDC} + (1 - \text{WCWF}) \times \text{TOTWF}), \text{WFINDC})
\]

\[
\sim \text{PEOPLE}
\]

\[
\sim \text{WORKFORCE LEVEL NEEDED}
\]
WCWF = MAX(WCWF1, WCWF2)
    ~ DIMENSIONLESS
    ~ WILLINGNESS TO CHANGE WORKFORCE LEVEL

WCWF1 = TWCFW1(TIMERM/WCWFTP)
    ~ DIMENSIONLESS
    ~ WILLINGNESS TO CHANGE WORKFORCE (1)

TWCFW1(0, 0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1, 2.4, 2.7, 3, 0, 0.1, 0.4, 0.85, 1, 1, 1, 1, 1, 1)
    ~ ~ ~

WCWFTP = IF THEN ELSE(TMPRMR = 0, HIREDY + ASIMDY, TMPRMR)
    ~
    ~ TIME PARAMETER DAYS

TMPRMR = 0
    ~
    ~ TIME PARAMETER (= HIREDY + ASIMDY) DAYS

WCWF2 = TWCFW2(SCHCDT/MXTLCD)
    ~ DIMENSIONLESS
    ~ WILLINGNESS TO CHANGE WF (2)

TWCFW2(0.86, 0.88, 0.9, 0.92, 0.94, 0.96, 0.98, 1, 0.1, 0.2, 0.35, 0.6, 0.7, 0.77, 0.8)
    ~ ~ ~

MXTLCD = INITIAL(MXSCDX*TDEV)
    ~ DAYS
    ~ MAXIMUM TOLERABLE COMLETION DATE

MXSCDX = 1e+006
    ~ DIMENSIONLESS
    ~ MAX SCHEDULE COMPLETION DATE EXTENSION
Sketch Information

\(// Sketch information - do not modify anything except names
V160 Do not put anything below this section - it will be ignored
*Logo
$Tms Rmn|10|10-0-0-0-0-0
|0,vsr0000.bmp,285,144,256,128,0,5,0,0,0,-1--1,-1--1
\(-/-// Sketch information - do not modify anything except names
V160 Do not put anything below this section - it will be ignored
*1
$Times New Roman|8|10-0-0-255|10-0-0-0-0
|0,"
Human Resources Mgmt
,246,54,30,30,2,131,0,6,0,0,Times New Roman|8|10-0-0-255-128-64,0-0
|11,"
Production
,247,166,32,32,2,131,0,6,0,1,Times New Roman|8|164-0-0-128-255-128-64,0-0
|12,"
Work Force Available
,242,102,55,7,0,3,0,3,-1,0,Arial|8|10-0-0-0-0-1--1
|13,"
Planning
,367,232,30,30,2,131,0,6,0,0,Times New Roman|8|10-0-0-255-255,-1--1
|14,"
Controlling
,128,231,31,31,2,131,0,6,0,0,Times New Roman|8|10-0-0-255-128-192,-1--1
|15,"
SOFTWARE DEVELOPMENT MANAGEMENT MODEL
,257,308,235,11,0,3,0,2,-1,0,Times New Roman|14|BU0-64-128,-1--1
|16,"
Progress Status
,163,143,42,7,0,3,0,3,-1,0,Arial|8|10-0-0-0-0-1--1
|17,"
Work Force Needed
,325,141,50,7,0,3,0,3,-1,0,Arial|8|10-0-0-0-0-1--1
|18,"
Tasks Completed
,210,222,44,7,0,3,0,3,-1,0,Arial|8|10-0-0-0-0-1--1
|19,"
Schedule
,298,223,25,7,0,3,0,3,-1,0,Arial|8|10-0-0-0-0-1--1
|20,"
Effort Remaining
,250,263,43,7,0,3,0,3,-1,0,Arial|8|10-0-0-0-0-1--1

-11
>0,1,0,0,0,0,-1--1,1l(246,101)l
>3,1,0,0,0,0,-1--1,1l(314,202)l
!!---/// Sketch information - do not modify anything except names
V160  Do not put anything below this section - it will be ignored
*3
$Times New Roman10\ﻙ0-0-0-0-0-0-0-0-0-0-0-0
|0, DMPQA, 216, 88, 20, 20, 2, 0, 6, 0, 0, Arial8|B|0-0-0-0-255-255-255-128-0-0-0-0
|1, DMPATR, 346, 192, 20, 20, 2, 0, 7, 0, 0, Arial8|B|0-0-0-0-255-255-255-128-0-0-0-0
|2, FRMPQA, 216, 16, 24, 7, 0, 0, 0, 2, -1, 0, Arial8|B|0-0-0-0-255-255-255-0-0-0-0
|3, TOTDMP, 345, 87, 20, 20, 2, 0, 7, 0, 0, Arial8|B|0-0-0-0-255-255-255-0-0-0-0
|4, ADMPPS, 344, 17, 24, 7, 0, 0, 2, -1, 0, Arial8|B|0-0-0-0-255-255-255-0-0-0-0

102
Manpower Allocation Sector

318,367,130,13,0,3,0,2,-1,0,Times New Roman16iBUl0-0-160,-1--1--1
>1,0,0,0,0,0,0,-1--1--1,1I(286,144)|
>2,0,0,0,0,0,0,-1--1--1,1I(216,38)|
>3,0,0,0,0,0,0,-1--1--1,1I(287,87)|
>4,3,0,0,0,0,0,-1--1--1,1I(344,38)|
>5,3,0,0,0,0,0,-1--1--1,1I(417,87)|
>6,1,0,0,0,0,0,-1--1--1,1I(419,192)|
>3,1,0,0,0,0,0,-1--1--1,1I(345,132)|
>9,7,1,0,0,0,0,-1--1--1,1I(289,192)|
>9,0,0,0,0,0,0,-1--1--1,1I(216,133)|
>11,8,0,0,0,0,0,-1--1--1,1I(350,253)|
>12,8,1,0,0,0,0,-1--1--1,1I(424,297)|
>13,10,1,0,0,0,0,-1--1--1,1I(199,281)|
>14,10,0,0,0,0,0,-1--1--1,1I(122,182)|
>8,7,1,0,0,0,0,-1--1--1,1I(291,297)|
\/-// Sketch information - do not modify anything except names

V160 Do not put anything below this section - it will be ignored

*4

$Times New Romanl0iBu0-0-000-000-000-0

10,EXHLEV,457,206,24,10,3,0,0,6,0,0,Times New Roman6iBu0-0-0-0-255-255,0-0-0
11,RDEXHL,543,96,15,15,2,0,0,7,0,0,Times New Roman6iBu0-0-0-255-255-128,0-0-0
12,PBWKRS,273,225,15,15,2,0,0,7,0,0,Times New Roman6iBu0-0-0-255-255-128,0-0-0
13,AFMDPI,369,199,26,10,3,0,0,7,0,0,Times New Roman6iBu0-0-0-255-255,-1--1--1
14,NFMDPI,436,138,19,6,0,0,0,2,0,0,1,Times New Roman6iBu0-0-0-1--1--1
15,48,535,35,8,8,0,132,0,2,0,1,Times New Roman6iBu0-0-0-0-0-0

-6

7,1,475,70,6,8,39,131,0,0,0,0,4--1--1,0,1-0-160
8,RIEXHL,498,70,17,6,32,0,0,7,1,0,Times New Roman6iBu0-0-160,0-255-255,0-0-160

-9

10,EXHDDY,534,206,20,6,0,0,0,3,-1,0,Times New Roman6iBu0-0-0-0-0-0
11,WKRAYD,280,171,16,16,2,1,0,7,0,0,Times New Roman6iBu0-0-0-255-255-232,0-0-0
12,WKRTS,402,243,15,15,2,0,0,7,0,0,Times New Roman6iBu0-0-0-255-255-0-0-0
13,1,350,177,6,8,39,131,0,0,0,0,4--1--1--1,0-0-255
14,WRAIDR,375,177,19,6,32,0,0,3,-1,0,Times New Roman6iBu0-0-160,0-0-255
15,FTEQWF,216,288,16,16,2,1,0,7,0,0,Times New Roman6iBu0-0-0-255-255-232,0-0-0
Quality Assurance Sector

System Testing Sector

106
27,13,1,0,0,0,0,-1--1,1(175,162)
44,32,1,0,0,0,0,-1--1,1(458,274)
3,32,1,0,0,0,0,-1--1,1(359,297)
21,32,1,0,0,0,0,-1--1,1(413,283)
42,32,1,0,0,0,0,-1--1,1(536,304)
5,0,1,0,0,0,0,-1--1,1(181,105)
46,43,4,0,0,12,1,0-0-0-0(73,354)(559,196)(552,196)(539,196)
46,12,3,0,0,12,1,0-0-0,1(71,278)
46,1,3,0,0,12,1,0-0-0,1(106,309)
\--/// Sketch information - do not modify anything except names
V160 Do not put anything below this section - it will be ignored
*7
$Arial8\|0-0-0|0-0|0-0\|0-0$
10,ASSPRD,52,219,17,17,2,0,0,5,0,0,0,Arial8\|BI0-0-0-0,255-255-0,0-0-0-0
11,MDPNR,179,158,18,18,2,0,0,5,0,0,0,Arial8\|BI0-0-0-0,255-255-0,0-0-0-0
12,TISKRM,116,176,22,22,2,1,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
13,PRMD,292,172,23,8,0,1,1,1,-1,0,128-128-128,128-128-128
14,MDHDL,316,106,20,20,2,1,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
15,MDRPNT,379,73,18,18,2,0,0,5,0,0,0,Arial8\|BI0-0-0,255-255-128,0-0-0-0
16,MDRM,240,130,19,19,2,1,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
17,SHRPTD,305,55,18,18,2,0,0,5,0,0,0,Arial8\|BI0-0-0,255-255-255,0-0-0-0
18,PMDSHR,235,47,17,17,2,0,0,5,0,0,0,Arial8\|BI0-0-0,255-255-255,0-0-0-0
19,MDPNRT,56,148,18,18,2,0,0,5,0,0,0,Arial8\|BI0-0-0,255-255-255,0-0-0-0
10,MDPRTD,91,57,17,17,2,0,0,5,0,0,0,Arial8\|BI0-0-0,255-255-255,0-0-0-0
11,DTCE,257,135,87,19,19,2,1,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
12,MDPNTS,475,55,18,18,2,0,0,5,0,0,0,Arial8\|BI0-0-0,255-255-255,0-0-0-0
13,PRWMPE,197,92,22,22,2,1,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
14,JPRDPD,205,221,20,20,2,0,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
15,PRDPD,251,285,20,20,2,0,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
16,WTIPJD,53,299,18,18,2,0,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
17,MDPRNT,283,228,20,20,2,0,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
18,JPBPWK,377,321,29,80,1,1,-1,-1,0,128-128-128,128-128-128
19,CMKT,153,284,21,21,2,1,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
20,CMTSM,344,219,24,24,2,1,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
21,CUMD,313,298,22,22,2,1,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
22,MPDEV,207,311,18,18,2,0,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
23,MPRER,96,264,22,22,2,1,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
24,PRTPRD,497,192,32,12,3,0,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
25,STSPRM,548,305,20,20,2,0,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
26,ACTSPD,430,133,20,20,2,0,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
27,CUMT,366,303,24,24,2,1,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
28,PLTSDP,464,257,20,20,2,0,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
29,TSTSPD,483,122,21,7,0,0,0,2,-1,0,0,Arial8\|BI0-0-0,0-0-0-0
30,"
Controlling Subsystem
31,359,102,13,0,3,0,2,-1,0,Times New Roman16\|BI0-0-160,-1--1--1
31,PJBSZ,422,293,20,20,2,1,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
32,TSZMD,531,238,20,20,2,1,0,5,0,0,0,Arial8\|BI0-0-0,255-255-232,0-0-0-0
-33
\textbackslash n\textbackslash n>6,5,1,0,0,0,0,-1--1,1(269,144)\]  
>7,5,1,0,0,0,0,-1--1,1(342,39)\]  
>4,7,1,0,0,0,0,-1--1,1(273,86)\]  
>8,7,1,0,0,0,0,-1--1,1(269,35)\]  
>6,8,1,0,0,0,0,-1--1,1(246,89)\]  
>10,8,1,0,0,0,-1--1,1(158,55)\]  
>9,10,1,0,0,0,-1--1,1(63,93)\]  
>1,10,1,0,0,0,0,-1--1,1(105,124)\]  
>12,10,1,0,0,0,0,-1--1,1(332,18)\]  
>11,1,1,0,0,0,0,-1--1,1(144,115)\]  
>13,1,1,0,0,0,0,-1--1,1(180,108)\]  
>0,9,0,0,0,0,-1--1,1(52,190)\]  
>2,9,1,0,0,0,0,-1--1,1(98,167)\]  
>14,0,1,0,0,0,0,-1--1,1(149,212)\]  
>15,0,1,0,0,0,0,-1--1,1(188,255)\]  
>16,0,1,0,0,0,0,-1--1,1(52,252)\]  
>17,14,0,0,0,0,0,-1--1,1(250,225)\]  
>2,14,0,0,0,0,0,-1--1,1(154,194)\]  
>19,15,1,0,0,0,0,-1--1,1(200,279)\]  
>20,15,1,0,0,0,0,-1--1,1(297,268)\]  
>21,15,1,0,0,0,0,-1--1,1(267,322)\]  
>22,16,1,0,0,0,0,-1--1,1(132,323)\]  
>23,16,1,0,0,0,0,-1--1,1(85,296)\]  
>24,12,1,0,0,0,0,-1--1,1(515,139)\]  
>25,12,1,0,0,0,0,-1--1,1(546,117)\]  
>1,17,0,0,0,0,0,-1--1,1(223,188)\]  
>12,17,1,0,0,0,0,-1--1,1(445,72)\]  
>6,17,1,0,0,0,0,-1--1,1(271,187)\]  
>18,22,0,1,0,0,0,-1--1,1(292,316)\]  
>26,24,0,0,0,0,0,-1--1,1(458,158)\]  
>27,24,1,0,0,0,0,-1--1,1(424,205)\]  
>28,24,1,0,0,0,0,-1--1,1(493,227)\]  
>29,24,1,0,0,0,0,-1--1,1(484,130)\]  
>27,25,1,0,0,0,0,-1--1,1(411,322)\]  
>31,25,1,0,0,0,0,-1--1,1(481,305)\]  
>31,28,1,0,0,0,0,-1--1,1(423,262)\]  
>32,28,1,0,0,0,0,-1--1,1(501,280)\]  
>20,26,1,0,0,0,0,-1--1,1(372,157)\]  
>27,26,1,0,0,0,0,-1--1,1(396,180)\]  
\textbackslash n\textbackslash noutdir// Sketch information - do not modify anything except names  
V160 Do not put anything below this section - it will be ignored  
*8  
$Arial8$\textbackslash l0,WFPS,388,45,20,2,0,0,5,0,0,\textit{Arial8}\textbackslash l1,ADMPSS,243,82,24,7,0,0,2,-1,0,\textit{Arial8}\textbackslash l2,CELTWF,193,45,20,2,1,0,5,0,0,\textit{Arial8}\textbackslash l3,WFSS,131,156,22,8,0,1,1,1,-1,0,128-128,128,128-128  
$WF\textbackslash n\textbackslash e\textbackslash n,199,124,20,20,2,0,0,5,0,0,\textit{Arial8}\textbackslash l5,TOTWF,107,50,20,2,1,0,5,0,0,\textit{Arial8}\textbackslash l0,255-255-232,0-0-0-0-0-0-0-0
Planning Subsystem

.294,329,91,13,0,3,0,2,-1,0,Times New Roman16Bi0-0-160,-1--1--1
.24,TDEVIN,509,296,29,8,0,1,1,1,-1,0,128-128-128,128-128-128
>1,0,1,0,0,0,0,-1--1--1,1(305,72)
>2,0,0,0,0,0,-1--1--1,1(283,45)
>3,0,0,1,0,0,-1--1--1,1(343,110)
>5,4,0,0,0,0,-1--1--1,1(147,82)
>6,4,0,0,0,0,-1--1--1,1(134,121)
>7,4,0,0,0,0,-1--1--1,1(271,121)
>8,1,0,0,0,0,-1--1--1,1(289,90)
>9,7,0,0,0,0,-1--1--1,1(383,120)
>12,7,1,0,0,0,-1--1--1,1(347,158)
>8,6,0,0,0,0,-1--1--1,1(157,157)
>11,6,0,0,0,0,-1--1--1,1(82,207)
>12,8,0,0,0,0,-1--1--1,1(267,188)
>13,8,1,0,0,0,-1--1--1,1(175,227)
>14,11,0,0,0,0,-1--1--1,1(176,262)
>15,11,1,0,0,0,-1--1--1,1(238,278)
>16,15,0,0,0,0,-1--1--1,1(451,260)
>15,12,0,0,0,0,-1--1--1,1(355,239)
>10,12,0,0,0,0,-1--1--1,1(363,188)
>17,10,0,0,0,0,-1--1--1,1(452,202)
>18,9,0,0,0,0,-1--1--1,1(461,145)
>19,9,0,0,0,0,-1--1--1,1(454,78)
>20,13,0,1,0,0,-1--1--1,1(143,165)
>21,13,0,1,0,0,-1--1--1,1(168,207)
>22,13,0,1,0,0,-1--1--1,1(186,197)
>24,16,0,1,0,0,-1--1--1,1(501,278)
Sketch information - do not modify anything except names
V160 Do not put anything below this section - it will be ignored
*System
$Tm: Rmn11010-0-010-0010-0-0
10, vskt0001.bmp, 298, 174, 205, 174, 0, 5, 0, 0, 0, -1--1--1, -1--1--1
11,
System Dynamics Primer
,302, 371, 160, 16, 0, 3, 0, 2, -1, 0, Arial20|Bl0-64-128, -1--1--1
-2
APPENDIX B.

SMFS VENSIM GRAPH DESCRIPTION (VGD) FILE

This appendix provides a listing of the pre-defined reports, help, and information screens and graphs available in the SMFS User Interface. These reports, graphs, and screens are contained in the Causal.vgd file. A Vensim graph description file allows the creation of customized output formats. It is a file that contains a series of keywords and values that describe what is to be displayed in a graph, table, or text report. The .vgd file structure and syntax depend on the commands used within the file, since each tool can contain different information. Further information about the various commands available can be found in Ventana [1994, p. 259-270]. The listing that follows provides the elements used within the SMFS .vgd file to display information and simulation results.

:REPORT SYSTEM
:TITLE System Dynamics Primer
:LOCATION -1,-1
:SIZE 64, 26

System Dynamics is a methodology for studying problems arising from the dynamic behavior of management and socioeconomic systems. The principal concern of a System Dynamics study is to understand the forces operating in a system in order to:

a. Determine the forces that influence on the stability of the system.
b. Determine the structure of the system as it relates to dynamic behavior such as growth, decline, or oscillation of central forces of interest.
c. Determine how the structure of a system can be modified to avoid unwanted situations or potentially dangerous conditions, and promote a better behavior.

The basic structure of a system dynamic model consists of a number of reservoirs or levels, interconnected by arrows representing information flow paths. The rate of information flow is controlled by functions that depend upon conditions in the system. The following are basic elements of a system dynamics model:

a. Levels or reservoirs - represented in diagrams as a boxes. Level variables are determined through the accumulation of information flow. They are always dependent upon rate variables. They represent an accumulation, or integration over time of flows or changes that come into and go out of the level.

b. Rates - rate variables represent the instantaneous flow to or from a level variable. They
are represented by a valve symbol to indicate control. Rate variable equations determine how the flow rates depends upon levels.

c. Auxiliary variables - represent intermediary variables that can occur anywhere in a path directed from a level to a rate. They represent intermediate values used in computing level and rate dependencies. Another view of auxiliary variables are as the combinations of information inputs into concepts. They are represented in diagrams as bubbles or circles.

d. Constants - are used to define fixed parameters in a model. They are represented as text labels.

e. Information flow - lines with arrow heads indicate the flow of information from one point to another. They represent a cause and effect between two variables.

f. Sources and Sinks - are represented by clouds. They represent the origin or termination of flows without any interest or consequence to the system model.

These basic elements are employed in developing feedback or causal-loop diagrams. The diagram is an aid used to determine the system structure and the presence of feedback loops. Feedback loops are either positive or negative. A positive feedback loop is usually associated with system growth, while negative feedback loops tend to stabilize the system and attain a particular state.

Causal-Loop Diagrams are the graphical equivalents of a system dynamics model set in mathematical equations. These diagrams have the advantage of showing system structure clearly, while emphasizing the presence and effect of feedback loops.

**END-OF-REPORT**

**REPORT INFO**

**TITLE** Model Overview

**LOCATION** 1,1

**SIZE** 54,15

This model represents a comprehensive system-dynamics model of the management process for software development. System dynamics is a methodology for studying the dynamic behavior of systems. It prescribes a set of concepts and procedures to assist in creating and testing system policies within the model. The model emphasizes the dependence of dynamic behavior upon information feedback within the cause and effect structure of the dynamics of the software development and management processes.

The model integrates four major subsystems covering the typical management functions of organizing, planning and controlling resources with the software production activities of design, coding and testing. The figure is a high level view of the four subsystems and their interconnections.

For an in-depth detail description of the model and its subsystems, please refer to "Software Project Dynamics, An integrated Approach", by Tarek K. Abdel-Hamid and Stuart E. Madnick.

**END-OF-REPORT**
This subsystem captures the hiring, training, assimilation, and transfer of people within the organization. The diagram illustrates the project's human resources as composed of two major work force levels named WFNEW and WEXP, for new hired work force and the experienced work force, respectively. Additionally, all the activities that affect these levels, as well as their feedback loops are shown in the diagram.

A more detail description of the subsystem can be found in “Software Project Dynamics, An Integrated Approach”, by Tarek K. Abdel-Hamid and Stuart E. Madnick.

This diagram represents the manpower allocations in the areas of quality assurance, production workload, and retesting. This sector is one of four sectors that comprise the Software production subsystem, one of the major activities of a software development project. The diagram depicts not only the information variables, but also their feedback flows.

A more detail description of the sector and related subsystems can be found in “Software Project Dynamics, An integrated Approach”, by Tarek K. Abdel-Hamid and Stuart E. Madnick.

This diagram illustrates the productivity subsector of software production as a function of complex factors affecting how much manpower is used and how productive they are. This sector is one of four sectors that comprise the Software production subsystem, one of the major activities of a software development project. The diagram depicts not only the information variables, but also their feedback flows.

A more detail description of the sector and related subsystems can be found in “Software Project Dynamics, An integrated Approach”, by Tarek K. Abdel-Hamid and Stuart E. Madnick.
This diagram illustrates the software quality assurance sector of software production as a function of complex factors in the generation, detection, and correction of errors during the development phase.

The sector includes activities related to the design, coding, reviewing, and testing, but excludes requirement definition activities, as they are assumed completed. This sector is one of four sectors that comprise the Software production subsystem, one of the major activities of a software development project. The diagram depicts not only the information variables, but also their feedback flows and interrelationships with other subsystems.

A more detailed description of the sector and related subsystems can be found in “Software Project Dynamics, An integrated Approach”, by Tarek K. Abdel-Hamid and Stuart E. Madnick.

This diagram illustrates the system testing sector of software production as a function of two processes: the growth of the undetected error populations and the system testing that result in the detection and correction of those errors.

This sector is one of four sectors that comprise the Software production subsystem, one of the major activities of a software development project. The diagram depicts not only the information variables, but also their feedback flows and interrelationships with other subsystems.

A more detailed description of the sector and related subsystems can be found in “Software Project Dynamics, An integrated Approach”, by Tarek K. Abdel-Hamid and Stuart E. Madnick.

This diagram illustrates the controlling subsystem of the model as a function of three major functions: measurement of what is happening in the activity being controlled, evaluating the significance of the measurements reported against a set of standards, and reporting what has been measured and assessed so corrective action can take place, if needed.

This sector is one of four subsystems that comprise the software development and management model. The diagram depicts not only the information variables, but also their feedback flows and interrelationships with other sectors and subsystems.
A more detailed description of the sector and related subsystems can be found in “Software Project Dynamics, An integrated Approach”, by Tarek K. Abdel-Hamid and Stuart E. Madnick.

END-OF-REPORT

REPORT INFO?
TITLE Planning Subsystem
LOCATION -1,-1
SIZE 54,12

This diagram illustrates the planning subsystem of the model as a function of initial estimates which are revised through the life of the project. It illustrates a causal-loop structure of the adjustments to workforce and schedule.

This subsystem is one of four subsystems that comprise the software development and management model. The diagram depicts not only the information variables, but also their feedback flows and interrelationship with other sectors and subsystems.

A more detail description of the sector and related subsystems can be found in “Software Project Dynamics, An integrated Approach”, by Tarek K. Abdel-Hamid and Stuart E. Madnick.

END-OF-REPORT

REPORT HELP1
TITLE Post-Analysis Help
LOCATION 30,75
SIZE 52,20

In order to conduct post-analysis of games, the user must load a previously run game. Once it is loaded, the user can easily perform analysis of the output. The user will load in the previous games by clicking on the Load button. This action will display a Load and Reorder Previous Scenarios window, which displays the names of previously saved games.

The Load and reorder command will put up a window with two lists. On the left is a list of loaded scenarios and on the right is a list of scenarios that have been run.

The << button loads scenarios. Click on the scenario you want to load in the right hand list, and then click the << button. If you do not have a mouse available, use the TAB and Return (or Enter) key combination to move among the boxed selection items and to make your selection.

The >> button unloads scenarios. Click on the scenario you want to unload in the left hand list, and the click on the >> button. If the list on the right is empty, you will need to run some simulation scenarios. You can do that by exiting to the main menu and selecting to run a game.

You can load up to 8 scenarios at a time, but it is recommended that no more than four be loaded at any given time since it becomes difficult to read graphs with more than four scenarios loaded.
The Select option allows the user to select a variable of interest for analysis. A Variable Selection control window that contains a list of variable names in the current model will be displayed. The user can make his selection by double-clicking on any variable on the list, by clicking on the variable of interest and then clicking on the SELECT button, or by typing in the space provided the name of the variable and then clicking the SELECT button.

Causal tracing is a process that allows you to determine the underlying causes of model behavior, and the differences in behavior between different scenarios. There are four options available for doing causal tracing. They are:

1) The Tree option enables the display of causes of the variable of interest, as a tree that branches from the right. Variables that are shown between parenthesis indicate that they appear somewhere else in the tree. Also, this option displays a graphical representation of the variable of interest. An option menu is provided for the user to continue further analysis.

2) The graph option is a refinement of the Tree option. It allows the display of the equation for the variable of interest and other pertinent information, and the graphical display of a predetermined set of variable levels affecting the variable of interest. An option menu is provided for the user to continue further analysis.

3) The Uses option enables the display of a uses of a variable as a tree branching from the left. Variables that are shown between parenthesis indicate that they appear somewhere else in the tree. Also, this option displays a graphical representation of the variable of interest. An option menu is provided for the user to continue further analysis.

4) The Loop option displays a list of all feedback loops passing through the variable of interest. The list is ordered from the shortest loop (the one involving the least number of variables) to the longest loop. Loops provides useful information about model interactions. Also, this option provides useful information about the variable of interest. An option menu is provided for the user to continue further analysis.

The List Differences option allows comparison of the constants used in the first scenario run to those in the second loaded scenario in a table format.

The Help option will produce this dialog window when selected. It can be locked in place, printed, copied into the Windows clipboard, or deleted when no longer needed.

The Return to Option will place the user at the main menu of the simulation.

The Exit option will terminate the simulation and close the program.

END-OF-REPORT

REPORT HELP2
TITLE Simulation Analysis Help
LOCATION 30,75
SIZE 52,22

You can perform analysis of games at any time during its execution.
The Select option allows the user to select a variable of interest for analysis. A variable Selection control window that contains a list of variable names in the current model will be displayed. The user can make his selection by double-clicking on any variable on the list, by clicking on the variable of interest and then clicking on the SELECT button, or by typing in the space provided the name of the variable and then clicking the SELECT button.

Causal tracing is a process that allows you to determine the underlying causes of model behavior, and the differences in behavior between different scenarios. There are four options available for doing causal tracing. They are:

1) The Tree option enables the display of causes of the variable of interest, as a tree that branches from the right. Variables that are shown between parenthesis indicate that they appear somewhere else in the tree. Also, this option displays a graphical representation of the variable of interest. An option menu is provided for the user to continue further analysis.

2) The graph option is a refinement of the Tree option. It allows the display of the equation for the variable of interest and other pertinent information, and the graphical display of a predetermined set of variable levels affecting the variable of interest. An option menu is provided for the user to continue further analysis.

3) The Uses option enables the display of the uses of a variable as a tree branching from the left. Variables that are shown between parenthesis indicate that they appear somewhere else in the tree. Also, this option displays a graphical representation of the variable of interest. An option menu is provided for the user to continue further analysis.

4) The Loop option displays a list of all feedback loops passing through the variable of interest.

The list is ordered from the shortest loop (the one involving the least number of variables) to the longest loop. Loops provides useful information about model interactions. Also, this option provides useful information about the variable of interest. An option menu is provided for the user to continue further analysis.

The Help option will produce this dialog window when selected. It can be locked in place, printed, copied into the Windows clipboard, or deleted when no longer needed.

The Exit to Simulation option, when selected, will place the user back to the game control center to continue playing the game.

END-OF-REPORT

REPORT HELP3
:TITLE Causal Tracing Help
:LOCATION 12,30
:SIZE 56,12

Causal tracing is a process that allows you to determine the underlying causes of model behavior, and the differences in behavior between different scenarios. Several screens provide several methods to study the causes of a variable of interest. The functions can be executed by pointing and clicking the respective buttons, or by pressing in the keyboard “ALT” and the first
letter of the word in the label indicated in the button. The following are options available for analysis:

1) The Trace on Highlight option allows the user to select a new variable of interest from the displayed tree diagram by clicking to highlight the desired new variable. Click on the “Yes” button to execute. The screen will automatically display the new variable of interest, its tree, and applicable graph. This function is executed also by pointing and double clicking the mouse left button on the variable of interest. This function is not available using the keyboard.

2) The Trace Loops option displays a list of all feedback loops passing through the variable of interest. The list is ordered from the shortest loop (the one involving the least number of variables) to the longest loop. Loops provides useful information about model interactions. Also, this option provides useful information about the variable of interest. An option menu is provided for the user to continue further analysis.

3) The Tree option enables the display of a causes of the variable of interest, as a tree that branches from the right. Variables that are shown between parenthesis indicate that they appear somewhere else in the tree. Also, this option displays a graphical representation of the variable of interest. An option menu is provided for the user to continue further analysis.

4) The Graph option is a refinement of the Tree option. It displays the equation containing the variable of interest and other pertinent information and the graphical display of a predetermined set of variable levels affecting the variable of interest. An option menu is provided for the user to continue further analysis.

5) The Uses option enables the display of the uses of a variable as a tree branching from the left. Variables that are shown between parenthesis indicate that they appear somewhere else in the tree. Also, this option displays a graphical representation of the variable of interest. An option menu is provided for the user to continue further analysis.

6) The Print options, depending of the options menu, will print the tree diagram, graph(s), loops involving the variable of interest, or the equation containing the variable of interest to the selected printer.

7) The Data Table option enables the display a window of the causes of the variable of interest in a tabulated table of values. This window can be locked in place, printed, copied into the Windows clipboard, or deleted when no longer needed.

8) The Select a New Variable function allows users to change the variable of interest among a given set of variables in the simulation. Users, using the trace on highlight option, can navigate and explore other variables within in the model. This option will let the user return back to the given set of variables to re-select among the given set after exploring other variables.

9) The Help option will produce this dialog window when selected. It can be locked in place, printed, copied into the Windows clipboard, or deleted when no longer needed.

10) The Exit to Simulation option, when selected, will place the user back to the game control center to continue playing the game.

:END-OF-REPORT
Project Cost: Enter your estimates for total Project Cost in Person-Days.
Project Duration: Enter your updated estimate for the Project Duration in days.
Staffing Level: Enter your total requested Staffing Level.
Pct Alloc to QA: Enter the desired percent of personnel allocated to Quality Assurance as a number from 0 to 100.

At time = \( \text{TM/ days} \)
\( \text{Current Total Staff Size} \)
\( \text{Staff Allocated to Programming} \)
\( \text{Staff Allocated to QA} \)
\( \text{Percent of Workforce that is Experienced} \)

Updated Estimates
\( \text{New Est of System Size due to changes in requirements} \)
\( \text{Your Last Est of Programming Phase Cost} \)
\( \text{Your Last Est of Programming Phase Duration} \)
\( \text{Time Remaining} \)

\( \text{\( \text{PJBSZT/ DSI} \)} \)
\( \text{\( \text{JBSZMD/ Person-Days} \)} \)
\( \text{\( \text{SCHCMT/ Days} \)} \)
\( \text{\( \text{TIMEIRM/ Days} \)} \)
REPORTED PROGRESS
Percent DSI Reported Complete
Total DSI Reported Complete to Date
Total Person-Days Expended to Date
Reported Productivity
DSI/Person-Days

:END-OF-REPORT

:REPORT RPT3
:TITLE Defect Report
:LOCATION -1,-1
:SIZE 45,16

At Time = \(\text{TM} / \text{days}\)
CUMULATIVE STATISTICS FROM START OF PROJECT
- TOTAL Person Days Expended to Date
- Programming Person Days Expended to Date
- QA Person-Days Expended to Date

- TOTAL Defects Detected
- TOTAL KDSI Completed
- Defect Density

STATISTICS FOR THE LAST 40 DAY PERIOD ONLY
- QA Person Days Expended
- Defects Detected
- Density of Defects Detected

:END-OF-REPORT

:GRAPH STATUS_GRAPH1
:TITLE Project Size and Status Graph
:X-LABEL Time in Days
:X-MIN 0
:X-MAX 600
:SCALE
:VAR JBSZMDI Estimated Programming Cost
:LINE-WIDTH 2
:UNITS Person-Days
:Y-MIN 0
:Y-MAX 4000
:SCALE
:VAR PJBSZT KDSI Estimated System Size
:LINE-WIDTH 2
:UNITS KDSI
:Y-MIN 0
:Y-MAX 64
:SCALE
:VAR FTEQWF Total Staff
:LINE-WIDTH 2
:UNITS Persons
:Y-MIN 0
:Y-MAX 24

:GRAPH UNSTACKED_2A
:TITLE Estimated Programming Cost
:X-LABEL Time in Days
:X-MIN 0
:X-MAX 600
:SCALE
:VAR JBSZMD
:LINE-WIDTH 2
:UNITS Person-Days
:Y-MIN 0
:Y-MAX 4000

:GRAPH UNSTACKED_2B
:TITLE Estimated System Size
:X-LABEL Time in Days
:X-MIN 0
:X-MAX 600
:SCALE
:VAR PJBSZT KDSI
:LINE-WIDTH 2
:UNITS KDSI
:Y-MIN 0
:Y-MAX 64

:GRAPH UNSTACKED_2C
:TITLE Total Staffing Level
:X-LABEL Time in Days
:X-MIN 0
:X-MAX 600
:SCALE
:VAR FTEQWF
:LINE-WIDTH 2
:UNITS Persons
:Y-MIN 0
:Y-MAX 24

:GRAPH DEFECTS
:TITLE Total Defects
:X-LABEL Time in Days
:X-MIN 0
:X-MAX 600
:WIP
:SCALE
:VAR PRQAMD PERIODQA Person-Days per KDSI
:LINE-WIDTH 2
APPENDIX C.

SMFS CUSTOM DESCRIPTION (VCD) FILE

This appendix contains a listing of the SMFS Vensim application (Venapps). The scripts that follow make possible the use of the Causal.vmf model and Causal.vgd file to form an easy-to-use interface application that facilitates the use and the interpretation of results from the software development and management model. The listing provides a series of menus and sequence of screens that allow the user of the SMFS to use and analyze the software development and management model in a straightforward and meaningful way. These sequence of screens is stored in a .vcd file. The SMFS Venapps is controlled using the file Causal.vcd. This file defines the appearance and behavior of the application using a simple scripting language. The structure and syntax of a Venapp screen follows this convention: Name, Text, Position, Justification, Accelerator/Range, Command, Shift screen. An example is shown in Figure 27.

![Venapps Screen Format](image)

**Figure 27.** Venapps Screen Format [Ventana, 1994, p. 316]

Controls form the basic components in a Venapps application screen. There are four types of controls available in the Vensim scripting language. They are:

- Output - Places information on the screen
- Tool - Places the output of Vensim tools on the screen
- Input - Allows the user input to the screen
- Command - Determines the logical flow and transition between screens.

The contents in a Venapps screen are built up from a combination of the different controls available. The Venapps scripting language allows to draw objects and text, display variable values, change variables, and display the output of Vensim tools. Every screen created requires a name in order to be a valid reference in the .vcd file. In order to form a valid screen, the first line must begin with the keyword :SCREEN, followed by the screen name, as in :SCREEN Welcome. Following this statement, any valid command or control statement can be used to define the appearance and behavior of the screen. The following screens are contained in the Causal.vcd file.

:SCREEN WELCOME
SCREENFONT, Times New Roman 12|0-0-0
COMMAND,"",0,0,0,0,,SPECIAL>SETTITLE,"Software Management Flight Simulator"
COMMAND,"",0,0,0,0,,SPECIAL>LOADMODEL,causal.vmf
COMMAND,"",0,0,0,0,,SPECIAL>READCUSTOM,causal.vgd
COMMAND,"",0,0,0,0,,SPECIAL>LOADTOOLSET,causal.vts
SKETCH,"Logo",5,15,90,65,,1,
BUTTON,"Start",22,85,22,0,L,R,,CHOICES
BUTTON,"Exit",56,85,22,0,L,E,E,X,X,SPECIAL>ASKYESNO\nDo you want to exit the simulator?&MENU>EXIT,

:SCREEN CHOICES
SCREENFONT, Times New Roman 12|0-0-0
TEXTONLY,"Software Management Flight Simulator",0,15,100,0,CI,Arial12|22|Bl|255-25-0-l
LINE,"",10,30,80,0,CIllll|0-125-125,
TEXTONLY,"System Dynamics Primer",15,36,45,0,CI,Arial14|Bl|l0-30-125l,
BUTTON,"System",60,36,20,0,L,S,,SYSTEM_DYNAMICS
TEXTONLY,"Simulator Model Overview",15,46,45,0,CI,Arial14|Bl|l0-30-125l,
BUTTON,"Overview",60,46,20,0,L,O,,MODELS
TEXTONLY,"Play a New Game",15,56,45,0,CI,Arial14|Bl|l0-30-125l,
BUTTON,"Game",60,56,20,0,L,G,,GAME_INTRO
TEXTONLY,"Analyze Previous Run Games",15,66,45,0,CI,Arial14|Bl|l0-30-125l,
BUTTON,"Analyze",60,66,20,0,L,A,,POST_ANALYZE
TEXTONLY,"Exit Flight Simulator",15,76,45,0,CI,Arial14|Bl|l0-30-125l,
BUTTON,"Exit",60,76,20,0,L,E,E,X,X,SPECIAL>ASKYESNO\nDo you want to exit the simulator?&MENU>EXIT,
LINE,"",10,87,80,0,CIllll|0-125-125,
:SCREEN SYSTEM_DYNAMICS
SKETCH,"System",0,0,100,90,,,10
BUTTON,"Return to Options",3,92,22,0,L99,CHOICES
BUTTON,"Primer",27,92,22,0,L99,CUSTOM>SYSTEM
BUTTON,"Model Overview",51,92,22,0,L99,MODEL
BUTTON,"Exit",75,92,22,0,L99,SPECIAL>ASKYESNO\Do you want to exit the simulator?\&MENU>EXIT,

:SCREEN GAME_INTRO
TEXTONLY,"Game Instructions",0,5,100,0,L99,Arial20\B\255-25-0I,
TEXTONLY, "You are not allowed to discuss this exercise with anyone other than the lab attendant.",5,15,100,0,L
TEXTONLY, "Please refrain from discussing this with members in the other class until they have completed.",5,20,100,0,L
TEXTONLY, "the exercise.",5,25,100,0,L
TEXTONLY, \"\",5,30,100,0,L
TEXTONLY, "The system will show you the size of the initial core team of software developers who have just completed the requirements/design specifications. The system will then advance to the programming phase where you will simulate the first 40 working day time period. You will\",5,45,100,0,L
TEXTONLY, "be allowed to view the various reports, graphs, perform analysis, and then update your estimates\",5,50,100,0,L
TEXTONLY, "for the project cost and duration and change your staffing levels.\",5,55,100,0,L
TEXTONLY, \"\",5,60,100,0,L
TEXTONLY, "Record your decisions for each interval on the documentation sheet provided before\",5,65,100,0,L
TEXTONLY, "proceeding to the next level.\",5,70,100,0,L
TEXTONLY, \"\",5,75,100,0,L
TEXTONLY, "THE LAB ATTENDANT MUST VERIFY YOUR FINAL RESULTS. GOOD LUCK!\",5,74,100,0,L\llib\255-0I,
BUTTON,"Continue",19,85,20,0,L99,SIMULATE>RUNNAME!?NAME FOR NEW GAME OUTPUT (NOT BASE!!),STARTGAME
BUTTON,"Previous Menu",40,85,20,0,L99,CUSTOM>CHOICES
BUTTON,"Exit",61,85,20,0,L99,SPECIAL>ASKYESNO\Do you want to exit the simulator?\&MENU>EXIT,

:SCREEN STARTGAME
COMMAND,\",\",0,0,0,0,SPECIAL>CLEARRUNS
COMMAND,\",\",0,0,0,0,SIMULATE>BASED\|
COMMAND,\",\",0,0,0,0,SIMULATE>RESUME10
COMMAND,\",\",0,0,0,0,GAME>GAMEINTERVAL40
COMMAND,\",\",0,0,0,0,MENU>GAME10
COMMAND,\",\",0,0,0,0,SIMULATE>CHGFILE\|
COMMAND,\",\",0,0,0,0,SIMULATE>BASED\|
COMMAND,\",\",0,0,0,0,SIMULATE>RESUME10
CLOSESCREEN,\",\",0,0,0,0,INITIAL_ESTIMATE

129
:SCREEN INITIAL_ESTIMATE
TEXTONLY,"Initial Estimates",0,5,100,0,Calibri20|B|255-25-0l,
TEXTONLY,"The initial core team of software developers have just completed the requirements
and",11,15,100,0,L
TEXTONLY,"design specifications. Your task is to take over as manager of the programming
phase.",7,20,100,0,L
TEXTONLY,"Shown below are the Initial Project Estimates:",7,25,0,0,L
TEXTONLY,"System Size in DSI",22,35,0,0,L||0-0-125
SHOWVAR,"IPRJSZ",73,35,0,0,R||0-0-125
TEXTONLY,"Cost of Programming Phase in Man-Days",22,40,0,0,L||0-0-125
SHOWVAR,"TOMD1INIT",73,40,0,0,R||0-0-125
TEXTONLY,"Duration of Programming Phase, days",22,45,0,0,L||0-0-125
SHOWVAR,"TDEVINI",73,45,0,0,R||0-0-125
TEXTONLY,"Initial Development Team, Men",22,50,0,0,L||0-0-125
SHOWVAR,"WFS1",73,50,0,0,R||0-0-125
TEXTONLY,"Pct of Staff Allocated to QA, (%Men)",22,55,0,0,L||0-0-125
SHOWVAR,"FRMPPQA",73,55,0,0,R||0-0-125
TEXTONLY,"SMFS estimate for the percent of the total staff to allocate to QA is shown
above.",11,65,100,0,L
TEXTONLY,"Remember, SMFS has not yet been calibrated to your environment. This estimate
is",7,70,100,0,L
TEXTONLY,"merely illustrative. It may or may not be appropriate for your unique project. At
this",7,75,100,0,L
TEXTONLY,"point, you need to make two decisions based on this information.",7,80,100,0,L
TEXTONLY,"Press Any Key to Continue",0,90,100,0,Calibri14|B|0-0-125l,
ANYKEY,",",0,0,0,0,0,0,,FIRST_DECISION

:SCREEN FIRST_DECISION
TEXTONLY,"Initial Programming Phase Decisions",0,4,100,0,Calibri20|B|255-25-0l,
TEXTONLY,"Initial Project Estimates",0,14,100,0,Calibri14|B|0-0-125l,
TEXTONLY,"System Size in DSI",22,20,0,0,L
SHOWVAR,"IPRJSZ",73,20,0,0,R
TEXTONLY,"Cost of Programming Phase in Man-Days",22,25,0,0,L
SHOWVAR,"TOMD1INIT",73,25,0,0,R
TEXTONLY,"Duration of Programming Phase, days",22,30,0,0,L
SHOWVAR,"TDEVINI",73,30,0,0,R
TEXTONLY,"Initial Development Team, Men",22,35,0,0,L
SHOWVAR,"WFS1",73,35,0,0,R
TEXTONLY,"Pct of Staff Allocated to QA, (%Men)",22,40,0,0,L
SHOWVAR,"FRMPPQA",73,40,0,0,R
TEXTONLY,"FIRST DECISION: Determine the Total Staff Level for the Programming \nPhase.",0,48,100,0,C112|B|l
TEXTONLY,"Staffing Level",35,54,0,0,L||0-0-125l
MODVAR,"WFS1",57,54,7,5,0
TEXTONLY,"SECOND DECISION: Determine the % of Personnel Allocated to \nQuality Assurance.",0,62,100,0,C112|B|l
TEXTONLY,"% Alloc to QA",35,68,0,0,L||0-0-125l
MODVAR,"FRMPPQA",57,68,7,5,0,0100
TEXTONLY,"IMPORTANT!!",0,80,100,0,C112|B|255-25-0l,
This is Your FINAL Opportunity to Change These Values.

Control Center

Project Estimates

System Size in DSI

Cost of Programming, Man-Days

Duration of Programming, Days

Initial Development Team, Men

New System Size in DSI

Total DSI Reported Completed

Reported Productivity, DSI/Man-Days

Time in Days

Input Variables

Project Cost, Man-Days

Project Duration, Days

Staffing Level, Men

Staffing Report

Staffing Graph

Defect Report

Defect Graph

Analyze Scenario

End Simulation

Advance Time

Transit

Continue
BRANCH,"END",0,0,0,0,,GAME>ENDGAME

:SCREEN PROJ_STATUS_GRAPH1
TOOL,"GR1",0,0,100,90,,CUSTOM>STATUS_GRAPH1
BUTTON,"Go Back",5,93,25,0,L,Gg,,CONTROL
BUTTON,"Unstack Graphs",37,93,25,0,L,Uu,,UNSTACKED_MENU
BUTTON,"Print Current View",70,93,25,0,L,Pp,PRINT>GR1

:SCREEN UNSTACKED_MENU
TEXTONLY,"Listed below are the three individual graphs which are plotted on
the",0,70,100,0,CnI44I255-0-0l,
TEXTONLY,"Project Status Graph. Select the desired graph or table of numeric
values.",0,75,100,0,CnI44I255-0-0l,
TOOL,"GR1",0,0,100,65,,CUSTOM>STATUS_GRAPH1
BUTTON,"Pgm @ost Graph",2,86,20,0,L,Cc,,UNSTACKED_PGM_COST
BUTTON,"Pgm(m) Cost Table",2,93,20,0,L,Mrn,,PGM_COST_TABLE
BUTTON,"Sys Size Graph",27,86,20,0,L,Mr,,UNSTACKED_SYS_SIZE
BUTTON,"(y)s Size Table",27,93,20,0,L,Yy,,SYS_SIZE_TABLE
BUTTON,"S(t)affing Graph",52,86,20,0,L,Tt,,UNSTACKED_TOT_STAFF
BUTTON,"St(t)affing Table",52,93,20,0,L,At,,TOT_STAFF_TABLE
BUTTON,"Go Back",78,86,20,0,L,Gg,,PROJ_STATUS_GRAPH1
BUTTON,"Print Current View",78,93,20,0,L,Pp,PRINT>GR1

:SCREEN UNSTACKED_PGM_COST
TOOL,"GR1A",0,0,100,80,,CUSTOM>UNSTACKED_2A
BUTTON,"Pgm @ost Graph",2,86,20,0,L,Cc,,UNSTACKED_PGM_COST
BUTTON,"Pgm(m) Cost Table",2,93,20,0,L,Mrn,,PGM_COST_TABLE
BUTTON,"Sys Size Graph",27,86,20,0,L,Mr,,UNSTACKED_SYS_SIZE
BUTTON,"(y)s Size Table",27,93,20,0,L,Yy,,SYS_SIZE_TABLE
BUTTON,"S(t)affing Graph",52,86,20,0,L,Tt,,UNSTACKED_TOT_STAFF
BUTTON,"St(t)affing Table",52,93,20,0,L,At,,TOT_STAFF_TABLE
BUTTON,"Go Back",78,86,20,0,L,Gg,,PROJ_STATUS_GRAPH1
BUTTON,"Print Current View",78,93,20,0,L,Pp,PRINT>GR1A

:SCREEN PGM_COST_TABLE
TEXTONLY,"Estimated Programming Cost",0,5,100,0,CnTimes New Roman|l6l
COMMAND,"",0,0,0,0,,SPECIAL>SETWBITEM|BSZMD
TOOL,"GR1A",0,30,100,40,,WORKBENCH>CAUSES TAB
BUTTON,"Pgm @ost Graph",2,86,20,0,L,Cc,,UNSTACKED_PGM_COST
BUTTON,"Pgm(m) Cost Table",2,93,20,0,L,Mrn,,PGM_COST_TABLE
BUTTON,"Sys Size Graph",27,86,20,0,L,Mr,,UNSTACKED_SYS_SIZE
BUTTON,"(y)s Size Table",27,93,20,0,L,Yy,,SYS_SIZE_TABLE
BUTTON,"S(t)affing Graph",52,86,20,0,L,Tt,,UNSTACKED_TOT_STAFF
BUTTON,"St(t)affing Table",52,93,20,0,L,At,,TOT_STAFF_TABLE
BUTTON,"Go Back",78,86,20,0,L,Gg,,PROJ_STATUS_GRAPH1
BUTTON,"Print Current View",78,93,20,0,L,Pp,PRINT>GR1A

:SCREEN UNSTACKED_SYS_SIZE
TOOL,"GR2A",0,0,100,80,,CUSTOM>UNSTACKED_2B

132
BUTTON,"View Table",37,93,25,0,L,Vv,DEFECT_TABLE
BUTTON,"Print Current View",70,93,25,0,L,Pp,PRINT>DR1

:SCREEN DEFECT_TABLE
TEXTONLY,"Total Defects",0,5,100,0,CI_TimesNewRoman16l
COMMAND,"",0,0,0,0, SPECIAL>SETWBITEMIPRQAMD PERIOD
TOOL,"GR4",0,30,100,40,, WORKBENCH>CAUSES TAB
COMMAND,"",1,0,0,0, SPECIAL>SETWBITEMIPRDFDS
TOOL,"GR4",0,70,100,40,, WORKBENCH>CAUSES TAB
BUTTON,"Go Back",17,93,25,0,L,Gg,DEFECT_GRAPH
BUTTON,"Print Current View",60,93,25,0,L,Pp,PRINT>GR4

:SCREEN STAFFING_GRAPH
TOOL,"SG1",0,0,100,90,, CUSTOM>STAFFING_LEVEL
BUTTON,"Go Back",5,93,25,0,L,Gg,CUSTOM
BUTTON,"View Table",37,93,25,0,L,Vv,STAFFING_TABLE
BUTTON,"Print Current View",70,93,25,0,L,Pp,PRINT>SG1

:SCREEN STAFFING_TABLE
TEXTONLY,"Total Staff Composition",0,5,100,0,CI_TimesNewRoman16l
COMMAND,"",1,0,0,0, SPECIAL>SETWBITEMIFTEQWF
TOOL,"GR5",0,20,100,40,, WORKBENCH>CAUSES TAB
COMMAND,"",1,0,0,0, SPECIAL>SETWBITEMICRQAWF
TOOL,"GR5",0,55,100,40,, WORKBENCH>CAUSES TAB
COMMAND,"",0,0,0,0, SPECIAL>SETWBITEMICRDVWF
TOOL,"GR5",0,85,100,30,, WORKBENCH>CAUSES TAB
BUTTON,"Go Back",17,93,25,0,L,Gg,STAFFING_GRAPH
BUTTON,"Print Current View",60,93,25,0,L,Pp,PRINT>GR5

:SCREEN MODELS
TEXTONLY,"Simulator Model Structures",28,5,45,0,CIArial26|Bl|255-0-0l,
TEXTONLY,"Model Overview",10,20,45,0,LIArial14|B|l0-125-0l,
BUTTON,"Overview",68,20,20,0,L,Oo,OVERVIEW
TEXTONLY,"Human Resources Subsystem",10,28,45,0,LIArial14|B|l0-30-125l,
BUTTON,"HR",68,28,20,0,L,Hh,HUMAN_RESOURCES_OV
TEXTONLY,"Manpower Allocation Sector",10,36,0,0,LIArial14|B|l0-30-125l,
BUTTON,"Manpower",68,36,20,0,L,Mm,MAN_POWER_OV
TEXTONLY,"SW Development Productivity Subsector",10,44,45,0,LIArial14|B|l0-30-125l,
BUTTON,"Software",68,44,20,0,L,Ss,SW_DEV_PROD_OV
TEXTONLY,"Quality Assurance Sector",10,52,45,0,LIArial14|B|l0-30-125l,
BUTTON,"QA",68,52,20,0,L,Qq,QUALITY_ASSURANCE_OV
TEXTONLY,"System Testing Sector",10,60,45,0,LIArial14|B|l0-30-125l,
BUTTON,"Testing",68,60,20,0,L,Tt,SYSTESTING_OV
TEXTONLY,"Controlling Subsystem",10,68,45,0,LIArial14|B|l0-30-125l,
BUTTON,"Control",68,68,20,0,L,Cc,CONTROL_SYS_OV
TEXTONLY,"Planning Subsystem",10,76,45,0,LIArial14|B|l0-30-125l,
BUTTON,"Planning",68,76,20,0,L,Pp,PLANNING_SYS_OV
BUTTON,"Return to Options",28,90,20,0,L,Rr,CHOICES
BUTTON,"Exit",52,90,20,0,L,EeXx,SPECIAL>ASKYESNO\
Do you want to exit the simulator? & MENU > EXIT

:SCREEN OVERVIEW
TEXTONLY,"Simulator Main Block Components",0,5,100,0,C128B12S5-0-0
SKETCH,"SK1",10,15,80,75,,2
BUTTON,"Info",2,92,11,0,L,li,CUSTOM>INFO
BUTTON,"Print View",15,92,22,0,L,Pp,PRINT>SK1
BUTTON,"Model Components",39,92,22,0,L,Mm,,MODELS
BUTTON,"Return to Options",63,92,22,0,L,Rr,,CHOICES
BUTTON,"Exit",87,92,11,0,L,EeXx, SPECIAL>ASKYESNO
Do you want to exit the simulator? & MENU > EXIT

:SCREEN HUMAN_RESOURCES_OV
SKETCH,"SK2",0,0,100,90,,3
BUTTON,"Info",2,92,11,0,L,li,CUSTOM>INFO1
BUTTON,"Print View",15,92,22,0,L,Pp,PRINT>SK2
BUTTON,"Model Components",39,92,22,0,L,Mm,,MODELS
BUTTON,"Return to Options",63,92,22,0,L,Rr,,CHOICES
BUTTON,"Exit",87,92,11,0,L,EeXx, SPECIAL>ASKYESNO
Do you want to exit the simulator? & MENU > EXIT

:SCREEN MAN_POWER_OV
SKETCH,"SK3",0,0,100,90,,4
BUTTON,"Info",2,92,11,0,L,li,CUSTOM>INFO2
BUTTON,"Print View",15,92,22,0,L,Pp,PRINT>SK3
BUTTON,"Model Components",39,92,22,0,L,Mm,,MODELS
BUTTON,"Return to Options",63,92,22,0,L,Rr,,CHOICES
BUTTON,"Exit",87,92,11,0,L,EeXx, SPECIAL>ASKYESNO
Do you want to exit the simulator? & MENU > EXIT

:SCREEN SW_DEV_PROD_OV
SKETCH,"SK4",0,0,100,90,,5
BUTTON,"Info",2,92,11,0,L,li,CUSTOM>INFO3
BUTTON,"Print View",15,92,22,0,L,Pp,PRINT>SK4
BUTTON,"Model Components",39,92,22,0,L,Mm,,MODELS
BUTTON,"Return to Options",63,92,22,0,L,Rr,,CHOICES
BUTTON,"Exit",87,92,11,0,L,EeXx, SPECIAL>ASKYESNO
Do you want to exit the simulator? & MENU > EXIT

:SCREEN QUALITY_ASSURANCE_OV
SKETCH,"SK5",0,0,100,90,,6
BUTTON,"Info",2,92,11,0,L,li,CUSTOM>INFO4
BUTTON,"Print View",15,92,22,0,L,Pp,PRINT>SK5
BUTTON,"Model Components",39,92,22,0,L,Mm,,MODELS
BUTTON,"Return to Options",63,92,22,0,L,Rr,,CHOICES
BUTTON,"Exit",87,92,11,0,L,EeXx, SPECIAL>ASKYESNO
Do you want to exit the simulator? & MENU > EXIT

135
Do you want to exit the simulator? & MENU>EXIT,

Do you want to exit the simulator? & MENU>EXIT,

Do you want to exit the simulator? & MENU>EXIT,

Do you want to exit the simulator? & MENU>EXIT,

TEXTONLY,"SIMULATION POST-ANALYSIS",0,6,100,0,Calibri|20|B|255-0-0|,
SETWB,"-0,0,0,0,,SPECIAL>SETWBITEMIPRDPDRD,
TEXTONLY,"Load, Unload, and Reorder Previously Run Scenarios",30,20,0,0,Calibri|12|B|0-0-125|,
BUTTON,"Load",5,20,20,0,Li,MENU>LOAD_RUN,
TEXTONLY,"Select Variable as a Start Point for Further Analysis",30,30,0,0,Calibri|12|B|0-0-125|,
BUTTON,"Select",5,30,20,0,Ls,SPECIAL>SETWBITEMIPRDPDRD,
&SPECIAL>VARSELECT|Select a New Variable to Trace, 
TEXTONLY,"Trace Underlying Causes Using Trees",30,40,0,0,Calibri|12|B|0-0-125|,
BUTTON,"Trees",5,40,20,0,L,Tt,,POST_CAUSE1
TEXTONLY,"Trace Underlying Causes Using Graphs",30,50,0,0,Calibri|12|B|0-0-125|,
BUTTON,"Graphs",5,50,20,0,L,Gg,,POST_CAUSE2
TEXTONLY,"Trace the Uses of a Variable",30,60,0,0,Calibri|12|B|0-0-125
BUTTON,"Uses",5,60,20,0,L,Uu,,POST_USE
TEXTONLY,"Trace the Feedback Loops of a Variable",30,70,0,0,Calibri|12|B|0-0-125|,
BUTTON,"Feedback",5,70,20,0,L,Ff,,POST_LOOPS
TEXTONLY,"List Differences of First Two Loaded Scenarios",30,80,0,0,Calibri|12|B|0-0-125|,
BUTTON,"Differences",5,80,20,0,L,Dd,,POST_DIFF
BUTTON,"Help",10,90,25,0,L,Hh,CUSTOM>HELP1
BUTTON,"Return to Options",37,90,25,0,L,Rr,SPECIAL>SETWBITEMIPRDPDRD,CHOICES
136
Do you want to exit the simulator?

:SCREEN POST_CAUSE1

TEXTONLY, "TREE CAUSAL TRACING -\".20,2,0,0,Lt16Bl255-0-0,1,
WBVAR,\"35,2,0,0,Lt16Bl255-0-0-125,1,
TOOL,\"TR1\",2,7,96,42,..WORKBENCH>CAUSES TREE

TEXTONLY, "OPTIONS",22,50,0,0,Lt16Bl255-0-0,
TEXTONLY, "Trace on Highlight",3,56,0,0,Lt16Bl255-0-0-125,1,

BUTTON,\"Yes\",43,56,10,5,Lt16Bl10Bl,Yy,..SPECIAL>SECONDCLICK\TR1,

TEXTONLY, "Trace Loops",3,62,0,0,Lt16Bl12Bl0-0-125,1,

BUTTON,\"Loops\",43,62,10,5,Lt16Bl12Bl0-0-125,1,

TEXTONLY, "Graph",3,68,10,5,Lt16Bl12Bl0-0-125,1,

BUTTON,\"Graph\",43,68,10,5,Lt16Bl12Bl0-0-125,1,

TEXTONLY, "Change to Graph Based",3,74,0,0,Lt16Bl12Bl0-0-125,1,

BUTTON,\"Print\",16,74,10,5,Lt16Bl12Bl0-0-125,1,

TEXTONLY, "Print Graph",28,74,0,0,Lt16Bl12Bl0-0-125,1,

BUTTON,\"OK\",43,74,10,5,Lt16Bl12Bl0-0-125,1,

TEXTONLY, "Data Table",3,80,0,0,Lt16Bl12Bl0-0-125,1,

BUTTON,\"Data\",43,80,10,5,Lt16Bl12Bl0-0-125,1,

WORKBENCH>CAUSES TAB

TEXTONLY, \"Select a New Variable to Trace\",3,86,0,0,Lt16Bl12Bl0-0-125,1,

BUTTON,\"Select\",43,86,10,5,Lt16Bl12Bl0-0-125,1,

SPECIAL>VARSELECT\New Variable for\n
Texting

TOOL,\"GR1\",55,50,43,49,..WORKBENCH>STRIP GRAPH

BUTTON,\"Help\",59,20,5,Lt16Bl12Bl0-0-125,1,

CUSTOM>HELP3

BUTTON,\"Exit to Analysis\",30,93,20,5,Lt16Bl12Bl0-0-125,1,

SPECIAL>SETWBITEM|PRDPRD,

POST_ANALYZE

SETWB,\"",0,0,0,0,..POST_CAUSE1

:SCREEN POST_CAUSE2

TEXTONLY, "GRAPH CAUSAL TRACING -\".1,2,50,0,Lt16Bl255-0-0,
WBVAR,\"35,2,0,0,Lt16Bl255-0-0-125,1,
TEXTONLY, "OPTIONS",21,50,0,0,Lt16Bl255-0-0-125,1,

TEXTONLY, "Trace on Highlight",3,56,0,0,Lt16Bl12Bl0-0-125,1,

BUTTON,\"Yes\",43,56,10,5,Lt16Bl12Bl0-0-125,1,

TEXTONLY, "Trace Loops",3,62,0,0,Lt16Bl12Bl0-0-125,1,

BUTTON,\"Loops\",43,62,10,5,Lt16Bl12Bl0-0-125,1,

TEXTONLY, "Change to Tree Based",3,68,0,0,Lt16Bl12Bl0-0-125,1,

BUTTON,\"Tree\",43,68,10,5,Lt16Bl12Bl0-0-125,1,

TEXTONLY, "Print Graph",3,74,0,0,Lt16Bl12Bl0-0-125,1,

BUTTON,\"Print\",18,74,10,5,Lt16Bl12Bl0-0-125,1,

TEXTONLY, "Print Eqn",30,74,0,0,Lt16Bl12Bl0-0-125,1,

BUTTON,\"OK\",43,74,10,5,Lt16Bl12Bl0-0-125,1,

TEXTONLY, "Data Table",3,80,0,0,Lt16Bl12Bl0-0-125,1,

BUTTON,\"Data\",43,80,10,5,Lt16Bl12Bl0-0-125,1,

WORKBENCH>CAUSES TAB

TEXTONLY, \"Select a New Variable to Trace\",3,86,0,0,Lt16Bl12Bl0-0-125,1,

BUTTON,\"Select\",43,86,10,5,Lt16Bl12Bl0-0-125,1,

SPECIAL>VARSELECT\New Variable for\n
Texting

TOOL,\"TR1\",55,1,44,98,..WORKBENCH>CAUSES STRIP

137
TOOL,"GR1",1,8,53,40,,WORKBENCH>DOCUMENT
BUTTON,"Help",5,93,20,5,LIArial110BI,Hh,CUSTOM>HELP3
BUTTON,"Exit to Analysis",30,93,20,5,LIArial110BI,EeXx,SPECIAL>SETWBITEMIPRDPDPR,
POST_ANALYZE
SETWB,,",0,0,0,0,,POST_CAUSE2

:SCREEN POST_USE
TEXTONLY,"USES of -",36,2,0,0,LI16BI255-0-0I,
WBVAR,",",51,2,0,0,LIArial16BI0-125-125I,
TEXTONLY,"OPTIONS",21,50,0,0,LIArial16BI255-0-0I,
TEXTONLY,"Trace on Highlight",3,56,0,0,LIArial12BI0-0-125I,
BUTTON,"Yes",43,56,10,5,LIArial10BI,Yy,SPECIAL>SECONDCLICKTR1,
TEXTONLY,"Trace Loops",3,62,0,0,LIArial12BI0-0-125I,
BUTTON,"Loops",43,62,10,5,LIArial10BI,Ll,,POST_LOOPS
TEXTONLY,"Change to Tree Based",3,68,0,0,LIArial12BI0-0-125I,
BUTTON,"Tree",43,68,10,5,LIArial10BI,Tt,,POST_CAUSE1
TEXTONLY,"Change to Graph Based",3,74,0,0,LIArial12BI0-0-125I,
BUTTON,"Graph",43,74,10,5,LIArial10BI,Gg,,POST_CAUSE2
TEXTONLY,"Print Tree",3,80,0,0,LIArial12BI0-0-125I,
BUTTON,"Print",16,80,10,5,LIArial10BI,Pp,PRINT>TR1,
TEXTONLY,"Print Graph",28,80,0,0,LIArial12BI0-0-125I,
BUTTON,"OK",43,80,10,5,LIArial10BI,Oo,PRINT>GR1
TEXTONLY,",Select a New Variable to Trace",3,86,0,0,LIArial12BI0-0-125I,
BUTTON,"Select",43,86,10,5,LIArial10BI,Ss,SPECIAL>VARSELECT\New Variable for\Tracing

TOOL,"TR1",2,7,96,42,,WORKBENCH>USES TREE
TOOL,"GR1",55,50,43,49,,WORKBENCH>STRIP GRAPH
BUTTON,"Help",5,93,20,5,LIArial10BI,Hh,CUSTOM>HELP3
BUTTON,"Exit to Analysis",30,93,20,5,LIArial110BI,EeXx,SPECIAL>SETWBITEMIPRDPDPR,
POST_ANALYZE
SETWB,,",0,0,0,0,,POST_USE

:SCREEN POST_LOOPS
TEXTONLY,"LOOPS CAUSAL TRACING -",1,2,50,0,LI16BI255-0-0,
WBVAR,",",35,2,0,0,LIArial12BI0-125-125I,
TEXTONLY,"OPTIONS",21,50,0,0,LIArial16BI255-0-0I,
TEXTONLY,"Trace on Highlight",3,56,0,0,LIArial12BI0-0-125I,
BUTTON,"Yes",43,56,10,5,LIArial10BI,Yy,SPECIAL>SECONDCLICKTR1,
TEXTONLY,"Change to Tree Based",3,62,0,0,LIArial12BI0-0-125I,
BUTTON,"Tree",43,62,10,5,LIArial10BI,Tt,,POST_CAUSE1
TEXTONLY,"Change to Graph Based",3,68,0,0,LIArial12BI0-0-125I,
BUTTON,"Graph",43,68,10,5,LIArial10BI,Gg,,POST_CAUSE2
TEXTONLY,"Print Loops",3,74,0,0,LIArial12BI0-0-125I,
BUTTON,"Print",18,74,10,5,LIArial10BI,Pp,PRINT>TR1,
TEXTONLY,"Print Eqn",30,74,0,0,LIArial12BI0-0-125I,
BUTTON,"OK",43,74,10,5,LIArial10BI,Oo,PRINT>GR1
TEXTONLY,"Find Variable Uses",3,80,0,0,LIArial12BI0-0-125I,
BUTTON,"Uses",43,80,10,5,LIArial10BI,Uu,,POST_USE
TEXTONLY,"Select a New Variable to Trace",3,86,0,0,LIArial12BI0-0-125I,
:SCREEN SIM_CAUSE2
TEXTONLY, "GRAPH CAUSAL TRACING:1,2,50,0,LLBIBI255-0-0
WBVAR,"",35,2,0,0,LLARIAL12BIB0-125-125L
TEXTONLY, "OPTIONS";21,50,0,0,LLARIAL16BIB255-0-0L
TEXTONLY, "Trace on Highlight";3,56,0,0,LLARIAL12BIB0-125-125L
BUTTON, "Yes";43,56,10,5,LLARIAL10BILY,SPECIAL>SECONDCLICK|TR1
TEXTONLY, "Trace Loops";3,62,0,0,LLARIAL12BIB0-125-125L
BUTTON, "Loops";43,62,10,5,LLARIAL10BIL,LL,,SIM_LOOPS
TEXTONLY, "Change to Tree Based";3,68,0,0,LLARIAL12BIB0-0-125L
BUTTON, "Tree";43,68,10,5,LLARIAL10BILT,TS,,SIM_CAUSE1
TEXTONLY, "Print Graph";3,74,0,0,LLARIAL12BIB0-0-125L
BUTTON, "Graph";18,74,10,5,LLARIAL10BILG,G,PRINT>TR1
TEXTONLY, "Print Eqn";30,74,0,0,LLARIAL12BIB0-0-125L
BUTTON, "OK";43,74,10,5,LLARIAL10BILO,PRINT>GR1
TEXTONLY, "Data Table";3,80,0,0,LLARIAL12BIB0-0-125L
BUTTON, "Data";43,80,10,5,LLARIAL10BILD,WORKBENCH>USES TAB
TEXTONLY, "Select a New Variable to Trace";3,86,0,0,LLARIAL12BIB0-0-125L
BUTTON, "Select";43,86,10,5,LLARIAL10BILS,S,,VARIABLES_2
TOOLT,"TR1";55,1,44,98,,WORKBENCH>USES STRIP
TOOLT,"GR1";1,8,53,40,,WORKBENCH>DOCUMENT
BUTTON, "Help";5,93,20,5,LLARIAL10BILH,H,CUSTOM>HELP3
BUTTON, "Exit to Simulation";30,93,20,5,LLARIAL10BILX,EXX,SPECIAL>SETWBITEM|PRDPRD,
CONTROL
SETWB,"",0,0,0,0,,SIM_CAUSE2

:SCREEN SIM_USE
TEXTONLY, "USES of ";36,2,0,0,LL16BIB255-0-0L
WBVAR,"",51,2,0,0,LLARIAL16BIB0-125-125L
TEXTONLY, "OPTIONS";21,50,0,0,LLARIAL16BIB255-0-0L
TEXTONLY, "Trace on Highlight";3,56,0,0,LLARIAL12BIB0-125-125L
BUTTON, "Yes";43,56,10,5,LLARIAL10BILY,SPECIAL>SECONDCLICK|TR1
TEXTONLY, "Trace Loops";3,62,0,0,LLARIAL12BIB0-125-125L
BUTTON, "Loops";43,62,10,5,LLARIAL10BIL,LL,,SIM_LOOPS
TEXTONLY, "Change to Tree Based";3,68,0,0,LLARIAL12BIB0-0-125L
BUTTON, "Tree";43,68,10,5,LLARIAL10BILT,TS,,SIM_CAUSE1
TEXTONLY, "Change to Graph Based";3,74,0,0,LLARIAL12BIB0-0-125L
BUTTON, "Graph";43,74,10,5,LLARIAL10BILG,,SIM_CAUSE2

140
TEXTONLY,"Print Tree",3,80,0,0,LIArial12|BI0-0-125l,
BUTTON,"Print",16,80,10,5,LIArial10|Bl,Pp,PRINT>TR1,
TEXTONLY,"Print Graph",28,80,0,0,LIArial12|BI0-0-125l,
BUTTON,"OK",43,80,10,5,LIArial10|BI0,PRINT>GR1
TEXTONLY,"Select a New Variable to Trace",3,86,0,0,LIArial12|BI0-0-125l,
BUTTON,"Select",43,86,10,5,LIArial10|Bl,SS,,VARIABLES_3
TOOL,"TR1",2,796,42,,WORKBENCH>USES TREE
TOOL,"GR1",55,50,43,49,,WORKBENCH>STRIP GRAPH
BUTTON,"Help",5,93,20,5,LIArial10|Bl,Hh,CUSTOM>HELP3
BUTTON,"Exit to Simulation",30,93,20,5,LIArial10|Bl,EeXx,SPECIAL>SETWBITEM|PRDPRD,\CONTROL
SETWB,"",0,0,0,0,,SIM_USE

:SCREEN SIM_LOOPS
TEXTONLY,"LOOPS CAUSAL TRACING - ",1,2,50,0,LI|Bl255-0-0
WBVAR,"",35,2,0,0,LIArial12|Bl0-125-125l,
TEXTONLY,"OPTIONS",21,50,0,0,LIArial16|Bl255-0-0l,
TEXTONLY,"Trace on Highlight",3,56,0,0,LIArial12|BI0-0-125l,
BUTTON,"Yes",43,56,10,5,LIArial10|Bl,Yy,SPECIAL>SECONDCLICK|TR1,
TEXTONLY,"Change to Tree Based",3,62,0,0,LIArial12|BI0-0-125l,
BUTTON,"Tree",43,62,10,5,LIArial10|Bl,Tt,,SIM_CAUSE1
TEXTONLY,"Change to Graph Based",3,68,0,0,LIArial12|BI0-0-125l,
BUTTON,"Graph",43,68,10,5,LIArial10|Bl,Gg,,SIM_CAUSE2
TEXTONLY,"Print Loops",3,74,0,0,LIArial12|BI0-0-125l,
BUTTON,"Print",18,74,10,5,LIArial10|Bl,Pp,PRINT>TR1,
TEXTONLY,"Print Eqn",30,74,0,0,LIArial12|BI0-0-125l,
BUTTON,"OK",43,74,10,5,LIArial10|Bl,Oo,PRINT>GR1
TEXTONLY,"Find Variable Uses",3,80,0,0,LIArial12|BI0-0-125l,
BUTTON,"Uses",43,80,10,5,LIArial10|Bl,Uu,,SIM_USE
TEXTONLY,"Select a New Variable to Trace",3,86,0,0,LIArial12|BI0-0-125l,
BUTTON,"Select",43,86,10,5,LIArial10|Bl,SS,,VARIABLES_4
TOOL,"TR1",55,1,44,98,,WORKBENCH>LOOPS
TOOL,"GR1",1,8,53,40,,WORKBENCH>DOCUMENT
BUTTON,"Help",5,93,20,5,LIArial10|Bl,Hh,CUSTOM>HELP3
BUTTON,"Exit to Simulation",30,93,20,5,LIArial10|Bl,EeXx,SPECIAL>SETWBITEM\PRDPRD,CONTROL
SETWB,"",0,0,0,0,,SIM_LOOPS

:SCREEN SIM_DIFF
TEXTONLY,"Differences Between First Two Loaded Scenarios"\,0,5,100,0,CIArial12|BI0255-0-0l,
 TOOL,"TR1",5,15,90,70,,WORKBENCH>RUNS COMPARE
BUTTON,"Exit to Analysis",20,90,25,0,L,EeXx,,CONTROL,
BUTTON,"Print",55,90,25,0,L,Pp,PRINT>T1,COUNTROL,

:SCREEN VARIABLES
TEXTONLY,"VARIABLE SELECTION",30,2,0,0,LI|Bl18|BI255-0-0l,
TEXTONLY,"OPTIONS",42,10,0,0,LIArial16|BI0-125-125l,
TEXTONLY,"Instantaneous Productivity",25,18,0,0,LiArial12|Bi0-0-125l,
BUTTON,"INSPRD",10,18,12,5,LiArial10|Bi,li,SPECIAL>SETWBITEM|INSPRD,\nSIM_ANALYZE,
TEXTONLY,"Perceived Development Productivity",25,24,0,0,LiArial12|Bi0-0-125l,
BUTTON,"PRDPRD",10,24,12,5,LiArial10|Bi,Pp,SPECIAL>SETWBITEM|PRDPRD,\nSIM_ANALYZE,
TEXTONLY,"Productivity in 40 Days Period",25,30,0,0,LiArial12|Bi0-0-125l,
BUTTON,"P@DPER",10,30,12,5,LiArial10|Bi,Dd,SPECIAL>SETWBITEM|PDPER,\nSIM_ANALYZE,
TEXTONLY,"Tasks Developed in 40 Days Period",25,36,0,0,LiArial12|Bi0-0-125l,
BUTTON,"PR(T)KDV",10,36,12,5,LiArial10|Bi,Tt,SPECIAL>SETWBITEM|PRTKDV,\nSIM_ANALYZE,
TEXTONLY,"Person Days Spent in Period",25,42,0,0,LiArial12|Bi0-0-125l,
BUTTON,"PR(M)D",10,42,12,5,LiArial10|Bi,Mm,SPECIAL>SETWBITEM|PRMD,\nSIM_ANALYZE,
TEXTONLY,"Fraction of WF Experienced",25,48,0,0,LiArial12|Bi0-0-125l,
BUTTON,"FRWFEX",10,48,12,5,LiArial10|Bi,Ff,SPECIAL>SETWBITEM|FRWFEX,\nSIM_ANALYZE,
TEXTONLY,"Actual Fraction of Man-Days on Project",25,54,0,0,LiArial12|Bi0-0-125l,
BUTTON,"AFMDPJ",10,54,12,5,LiArial10|Bi,Aa,SPECIAL>SETWBITEM|AFMDPJ,\nSIM_ANALYZE,
TEXTONLY,"Communications Overhead",25,60,0,0,LiArial12|Bi0-0-125l,
BUTTON,"COMMHO",10,60,12,5,LiArial10|Bi,Cc,SPECIAL>SETWBITEM|COMMHO,\nSIM_ANALYZE,
TEXTONLY,"SW Development Productivity",25,66,0,0,LiArial12|Bi0-0-125l,
BUTTON,"SDVPRD",10,66,12,5,LiArial10|Bi,ss,SPECIAL>SETWBITEM|SDVPRD,\nSIM_ANALYZE,
TEXTONLY,"Daily Manpower for Training",25,72,0,0,LiArial12|Bi0-0-125l,
BUTTON,"DMPTR(N)",10,72,12,5,LiArial10|Bi,Nn,SPECIAL>SETWBITEM|DMPTRN,\nSIM_ANALYZE,
TEXTONLY,"Cumulative Rework Man-Days",25,78,0,0,LiArial12|Bi0-0-125l,
BUTTON,"CMR(W)MD",10,78,12,5,LiArial10|Bi,Ww,SPECIAL>SETWBITEM|CMRMD,\nSIM_ANALYZE,
TEXTONLY,"Period's Defect Density",25,84,0,0,LiArial12|Bi0-0-125l,
BUTTON,"PR(D)FDS",10,84,12,5,LiArial10|Bi,Dd,SPECIAL>SETWBITEM|PRDFDS,\nSIM_ANALYZE,
TEXTONLY,"Error Generation Rate",25,90,0,0,LiArial12|Bi0-0-125l,
BUTTON,"ERR(G)RT",10,90,12,5,LiArial10|Bi,Ll,SPECIAL>SETWBITEM|ERRGRT,\nSIM_ANALYZE,
BUTTON,"CANCE(L)",70,75,20,8,LiArial10|Bi,Ll,SIM_ANALYZE,
BUTTON,"Exit to Simulation",70,85,20,8,LiArial10|Bi,EeXx,,CONTROL

:SCREEN VARIABLES_1
TEXTONLY,"VARIABLE SELECTION",30,2,0,0,Li18|Bi255-0-0l,
TEXTONLY,"OPTIONS",42,10,0,0,LiArial16|Bi0-125-125l,
TEXTONLY,"Instantaneous Productivity",25,18,0,0,LiArial12|Bi0-0-125l,
BUTTON,"INSPRD",10,18,12,5,LiArial10|Bi,li,SPECIAL>SETWBITEM|INSPRD,\nSIM_CAUSE1,
TEXTONLY,"Perceived Development Productivity",25,24,0,0,LiArial12|Bi0-0-125l,
BUG TON,"PRDPRD",10,24,12,5,LIArial10|B,|Pp,SPECIAL>SETWITEM|PRDPRD,\nSIM_CAUSE1,
TEXTONLY,"Productivity in 40 Days Period",25,30,0,0,LIArial12|B|0-0-125|l,
BUTTON, "P|DPER",10,30,12,5,LIArial10|B|Dd,SPECIAL>SETWITEM|PDPER,\nSIM_CAUSE1,
TEXTONLY,"Tasks Developed in 40 Days Period",25,36,0,0,LIArial12|B|0-0-125|l,
BUTTON, "P(R)TDV",10,36,12,5,LIArial10|B|Tt,SPECIAL>SETWITEM|PTDVTDV,\nSIM_CAUSE1,
TEXTONLY,"Person Days Spent in Period",25,42,0,0,LIArial12|B|0-0-125|l,
BUTTON, "P(R)MD",10,42,12,5,LIArial10|B|Mm,SPECIAL>SETWITEM|PRMD,\nSIM_CAUSE1,
TEXTONLY,"Fraction of WF Experienced",25,48,0,0,LIArial12|B|0-0-125|l,
BUTTON, "FRWFEX",10,48,12,5,LIArial10|B|Ff,SPECIAL>SETWITEM|FRWFEX,\nSIM_CAUSE1,
TEXTONLY,"Actual Fraction of Man-Days on Project",25,54,0,0,LIArial12|B|0-0-125|l,
BUTTON, "AFMDPJ",10,54,12,5,LIArial10|B|Aa,SPECIAL>SETWITEM|AFMDPJ,\nSIM_CAUSE1,
TEXTONLY,"Communications Overhead",25,60,0,0,LIArial12|B|0-0-125|l,
BUTTON, "COMMOH",10,60,12,5,LIArial10|B|Cc,SPECIAL>SETWITEM|COMMOH,\nSIM_CAUSE1,
TEXTONLY,"SW Development Productivity",25,66,0,0,LIArial12|B|0-0-125|l,
BUTTON, "SDVPRD",10,66,12,5,LIArial10|B|Ss,SPECIAL>SETWITEM|SDVPRD,\nSIM_CAUSE1,
TEXTONLY,"Daily Manpower for Training",25,72,0,0,LIArial12|B|0-0-125|l,
BUTTON, "DMPTR(N)",10,72,12,5,LIArial10|B|Nn,SPECIAL>SETWITEM|DMPTRN,\nSIM_CAUSE1,
TEXTONLY,"Cumulative Rework Man-Days",25,78,0,0,LIArial12|B|0-0-125|l,
BUTTON, "CMR(W)MD",10,78,12,5,LIArial10|B|Ww,SPECIAL>SETWITEM|CMRWM,\nSIM_CAUSE1,
TEXTONLY,"Period’s Defect Density",25,84,0,0,LIArial12|B|0-0-125|l,
BUTTON, "PR(D)DFS",10,84,12,5,LIArial10|B|Dd,SPECIAL>SETWITEM|PRDFS,\nSIM_CAUSE1,
TEXTONLY,"Error Generation Rate",25,90,0,0,LIArial12|B|0-0-125|l,
BUTTON, "ERR(G)RT",10,90,12,5,LIArial10|B|Ll,SPECIAL>SETWITEM|ERRGRT,\nSIM_CAUSE1,
BUTTON, "CAN(E)L",70,75,20,8,LIArial10|B|Ll,SPECIAL>SETWITEM|CANEL,\nBUTTON, "Exit to Simulation",70,85,20,8,LIArial10|B|EeXX,"CONTROL

:SCREEN VARIABLES_ 2
TEXTONLY,"VARIABLE SELECTION",30,2,0,0,LI|18|B|255-0-0l,
TEXTONLY,"OPTIONS",42,10,0,0,LIArial16|B|125-125|l,
TEXTONLY,"Instantaneous Productivity",25,18,0,0,LIArial12|B|0-0-125|l,
BUTTON, "INSPRD",10,18,12,5,LIArial10|B|Ii,SPECIAL>SETWITEM|INSPRD,\nSIM_CAUSE2,
TEXTONLY,"Perceived Development Productivity",25,24,0,0,LIArial12|B|0-0-125|l,
BUTTON, "PRDPRD",10,24,12,5,LIArial10|B|Pp,SPECIAL>SETWITEM|PRDPRD,\nSIM_CAUSE2,
TEXTONLY,"Productivity in 40 Days Period",25,30,0,0,LIArial12|B|0-0-125|l,
BUTTON, "P|DPER",10,30,12,5,LIArial10|B|Dd,SPECIAL>SETWITEM|PDPER,\nSIM_CAUSE2,
SIM_CAUSE2,
TEXTONLY,"Tasks Developed in 40 Days Period",25,36,0,0,Li,arial12|Bl0-0-125l,
BUTTON,"PR(T)KDV",10,36,12,5,Li,arial10|Bl,tt,SPECIAL>SETWBITEM|PRTKDV,\nSIM_CAUSE2,
TEXTONLY,"Person Days Spent in Period",25,42,0,0,Li,arial12|Bl0-0-125l,
BUTTON,"PR(M)D",10,42,12,5,Li,arial10|Bl,mm,SPECIAL>SETWBITEM|PRMD,\nSIM_CAUSE2,
TEXTONLY,"Fraction of WF Experienced",25,48,0,0,Li,arial12|Bl0-0-125l,
BUTTON,"FRWFEX",10,48,12,5,Li,arial10|Bl,ff,SPECIAL>SETWBITEM|FRWFEX,\nSIM_CAUSE2,
TEXTONLY,"Actual Fraction of Man-Days on Project",25,54,0,0,Li,arial12|Bl0-0-125l,
BUTTON,"AFMDPJ",10,54,12,5,Li,arial10|Bl,aa,SPECIAL>SETWBITEM|AFMDPJ,\nSIM_CAUSE2,
TEXTONLY,"Communications Overhead",25,60,0,0,Li,arial12|Bl0-0-125l,
BUTTON,"COMMOH",10,60,12,5,Li,arial10|Bl,cc,SPECIAL>SETWBITEM|COMMOH,\nSIM_CAUSE2,
TEXTONLY,"SW Development Productivity",25,66,0,0,Li,arial12|Bl0-0-125l,
BUTTON,"SDVPRD",10,66,12,5,Li,arial10|Bl,ss,SPECIAL>SETWBITEM|SDVPRD,\nSIM_CAUSE2,
TEXTONLY,"Daily Manpower for Training",25,72,0,0,Li,arial12|Bl0-0-125l,
BUTTON,"DMPTR(N)",10,72,12,5,Li,arial10|Bl,nn,SPECIAL>SETWBITEM|DMPTRN,\nSIM_CAUSE2,
TEXTONLY,"Cummulative Rework Man-Days",25,78,0,0,Li,arial12|Bl0-0-125l,
BUTTON,"CMR(W)MD",10,78,12,5,Li,arial10|Bl,ww,SPECIAL>SETWBITEM|CMRWMD,\nSIM_CAUSE2,
TEXTONLY,"Period's Defect Density",25,84,0,0,Li,arial12|Bl0-0-125l,
BUTTON,"PR(D)FDS",10,84,12,5,Li,arial10|Bl,dd,SPECIAL>SETWBITEM|PRDFDS,\nSIM_CAUSE2,
TEXTONLY,"Error Generation Rate",25,90,0,0,Li,arial12|Bl0-0-125l,
BUTTON,"ERR(G)RT",10,90,12,5,Li,arial10|Bl,rr,SPECIAL>SETWBITEM|ERRGRT,\nSIM_CAUSE2,
BUTTON,"CANCE(L)",70,75,20,8,Li,arial10|Bl,li,SIM_CAUSE2,
BUTTON,"Exit to Simulation",70,85,20,8,Li,arial10|Bl,exx,,CONTROL

:SCREEN VARIABLES_3
TEXTONLY,"VARIABLE SELECTION",30,2,0,0,Li,arial16|Bl0-0-125l,
TEXTONLY,"OPTIONS",42,10,0,0,Li,arial16|Bl0-125-125l,
TEXTONLY,"Instantaneous Productivity",25,18,0,0,Li,arial12|Bl0-0-125l,
BUTTON,"INSPRD",10,18,12,5,Li,arial10|Bl,li,SPECIAL>SETWBITEM|INSPRD,\nSIM_USE,
TEXTONLY,"Perceived Development Productivity",25,24,0,0,Li,arial12|Bl0-0-125l,
BUTTON,"PRDPDRD",10,24,12,5,Li,arial10|Bl,p,SPECIAL>SETWBITEM|PRDPDRD,\nSIM_USE,
TEXTONLY,"Productivity in 40 Days Period",25,30,0,0,Li,arial12|Bl0-0-125l,
BUTTON,"P®DPER",10,30,12,5,Li,arial10|Bl,dd,SPECIAL>SETWBITEM|PDPDPER,\nSIM_USE,
TEXTONLY,"Tasks Developed in 40 Days Period",25,36,0,0,Li,arial12|Bl0-0-125l,
BUTTON,"PR(T)KDV",10,36,12,5,Li,arial10|Bl,tt,SPECIAL>SETWBITEM|PRTKDV,\nSIM_USE,
TEXTONLY,"Person Days Spent in Period",25,42,0,0,LtArial|12|B|0-0-125l,
BUTTON,"PR(M)D",10,42,12,5,LtArial|10|B|l|Mm,SPECIAL>SETWBITEM|PRMDl,
SIM_USE,
TEXTONLY,"Fraction of WF Experienced",25,48,0,0,LtArial|12|B|0-0-125l,
BUTTON,"FRWFEX",10,48,12,5,LtArial|10|B|l|Ff,SPECIAL>SETWBITEM|FRWFEXl,
SIM_USE,
TEXTONLY,"Actual Fraction of Man-Days on Project",25,54,0,0,LtArial|12|B|0-0-125l,
BUTTON,"AFMDPJ",10,54,12,5,LtArial|10|B|l|Aa,SPECIAL>SETWBITEM|AFMDPJl,
SIM_USE,
TEXTONLY,"Communications Overhead",25,60,0,0,LtArial|12|B|0-0-125l,
BUTTON,"COMMOH",10,60,12,5,LtArial|10|B|l|Cc,SPECIAL>SETWBITEM|COMMOHl,
SIM_USE,
TEXTONLY,"SW Development Productivity",25,66,0,0,LtArial|12|B|0-0-125l,
BUTTON,"SDVPRD",10,66,12,5,LtArial|10|B|l|Ss,SPECIAL>SETWBITEM|SDVPRDl,
SIM_USE,
TEXTONLY,"Daily Manpower for Training",25,72,0,0,LtArial|12|B|0-0-125l,
BUTTON,"DMPTR(N)",10,72,12,5,LtArial|10|B|l|Nn,SPECIAL>SETWBITEM|DMPTRNl,
SIM_USE,
TEXTONLY,"Cumulative Rework Man-Days",25,78,0,0,LtArial|12|B|0-0-125l,
BUTTON,"CMR(W)MD",10,78,12,5,LtArial|10|B|l|Ww,SPECIAL>SETWBITEM|CMRWMDl,
SIM_USE,
TEXTONLY,"Period's Defect Density",25,84,0,0,LtArial|12|B|0-0-125l,
BUTTON,"PR(D)FDS",10,84,12,5,LtArial|10|B|l|Dd,SPECIAL>SETWBITEM|PRDFDSl,
SIM_USE,
TEXTONLY,"Error Generation Rate",25,90,0,0,LtArial|12|B|0-0-125l,
BUTTON,"ERR(G)RT",10,90,12,5,LtArial|10|B|l|Li,SPECIAL>SETWBITEM|ERRGRTl,
SIM_USE,
BUTTON,"CANCE(L)",70,75,20,8,LtArial|10|B|l|Li,,SIM_USE,
BUTTON,"Exit to Simulation",70,85,20,8,LtArial|10|B|l|EeXx,,CONTROL

:SCREEN VARIABLES_4
TEXTONLY,"VARIABLE SELECTION",30,2,0,0,Lt18|B|255-0-0l|
TEXTONLY,"OPTIONS",42,10,0,0,Lt16|B|125-125l|
TEXTONLY,"Instantaneous Productivity",25,18,0,0,LtArial|12|B|0-0-125l,
BUTTON,"INSPRD",10,18,12,5,LtArial|10|B|l|II,SPECIAL>SETWBITEM|INSPRDl,
SIM_LOOPS,
TEXTONLY,"Perceived Development Productivity",25,24,0,0,LtArial|12|B|0-0-125l,
BUTTON,"PRDPRD",10,24,12,5,LtArial|10|B|l|Pp,SPECIAL>SETWBITEM|PRDPRDl,
SIM_LOOPS,
TEXTONLY,"Productivity in 40 Days Period",25,30,0,0,LtArial|12|B|0-0-125l,
BUTTON,"P©PER",10,30,12,5,LtArial|10|B|l|Dd,SPECIAL>SETWBITEM|PRDPERl,
SIM_LOOPS,
TEXTONLY,"Tasks Developed in 40 Days Period",25,36,0,0,LtArial|12|B|0-0-125l,
BUTTON,"PR(T)KDV",10,36,12,5,LtArial|10|B|l|Tt,SPECIAL>SETWBITEM|PRTKDVl,
SIM_LOOPS,
TEXTONLY,"Person Days Spent in Period",25,42,0,0,LtArial|12|B|0-0-125l,
BUTTON,"PR(M)D",10,42,12,5,LtArial|10|B|l|Mm,SPECIAL>SETWBITEM|PRMDl,
SIM_LOOPS,
TEXTONLY,"Fraction of WF Experienced",25,48,0,0,LtArial|12|B|0-0-125l,
APPENDIX D.

SMFS VENSIM MODEL EQUATIONS

This appendix provides an enumerated and alphabetized listing of the equations that comprise the SMFS model as converted from Dynamo into Vensim, and as modified to allow user interaction. The equation listed follows the Vensim equation structure to facilitate their understanding. Appendix A provides an explanation of the structure and syntax of Vensim equations. The following list of variables is currently not in use during simulations (as reported by Vensim during model compilation) using the current version of the interface.

1. AFMPQA  
2. ALLERR  
3. ALLRWK  
4. CMDASI KDSI  
5. CMDVMD  
6. CMERD KDSI  
7. CMERES  
8. CMRWMD  
9. CMTRMD  
10. CRDVWF  
11. CROAWF  
12. CRRWWF  
13. CUMMD TD  
14. DEVMD  
15. FNCOST  
16. FNRES  
17. FNERR  
18. FNPRDT  
19. FNQAMD  
20. FNRWMD  
21. FNTIME  
22. FNTRMD  
23. FRMPQ1  
24. FRWFEX PCT  
25. INDCDT  
26. INSPRD  
27. INUDST  
28. IPRJSZ  
29. IRDVDT  
30. PIBSZT KDSI  
31. PRCMPL  
32. PRCTDT  
33. PRDFDS  
34. PRDPER  
35. PRQAMD PERIOD  
36. PTKTST  
37. PTRPTC  
38. RPRPROD  
39. SCHADT  
40. TEAMSZ  
41. TM  
42. TMSTTP  
43. WFNEED  
44. WFS2
**Causal Variable Listing**

VERSION 1
October 1995

1. ACTSPD=CUMTKT/(CMTSMD+0.001)
   ~ TASKS/MAN-DAY
   ~ ACTUAL TESTING PRODUCTIVITY

2. ADJQA=TADJQA(SCHPR)
   ~ PERCENT
   ~ PERCENT ADJUSTMENT IN PFMPQA

3. ADMPPS=1
   ~ DAY/DAY
   ~ AVERAGE DAILY MANPOWER PER STAFF

4. AEGRT=(ERRSRT+BDFXGR)*FRAERR
   ~ ERRORS/DAY
   ~ ACTIVE ERRORS GENERATION RATE

5. AERGRT=SDVRT*SMOOTH(AERRDS,TSAEDS)*MAERED
   ~ ERRORS/DAY
   ~ ACTIVE ERRORS REGENERATION RATE

6. AERRDS=UDAVER/(CUMTQA+0.1)
   ~ ERRORS/TASK
   ~ ACTIVE ERROR DENSITY

7. AERRFR=TERMFR(PJBAWK)
   ~ 1/DAYS
   ~ ACTIVE ERRORS RETIRING FRACTION

8. AERRRT=UDAVER*AERRFR
   ~ ERRORS/DAY
   ~ ACTIVE ERRORS RETIRING RATE
9. AFMDPJ=INTEG(WRADJR,NFMDPJ)
~ DIMENSIONLESS
~ ACTUAL FRACTION OF A MAN-DAY ON PROJECT

10. AFMPQA=ACTIVE INITIAL(PFMPQA*(1+ADJQA),PFMPQA)
~ DIMENSIONLESS
~ ACTUAL FRACTION OF MANPOWER FOR QA

11. ALESER=UDAVE+UDPVER+CMRWET
~ ERRORS
~ ALL ERRORS THAT ESCAPED AND WERE GENERATED

12. ALLERR=PTDTER+DTCERR+CMRWED+UDAVE+UDPVER+CMRWET
~ ERRORS
~ ALL ERRORS

13. ALLRWK=CMRWED+CMRWET
~ ERRORS
~ ALL ERRORS REWORKED ... IN DEVELOPMENT AND TESTING

14. ANERPT=MAX(PTDTER/(TSKWK+0.0001),0)
~ ERRORS/TASK
~ AVERAGE # OF ERRORS PER TASK

15. ANPPRD=FRWFEX*NWPPEX+(1-FRFEX)*NPWPNE
~ TASKS/MAN-DAY
~ AVERAGE NOMINAL POTENTIAL PRODUCTIVITY

16. AQADLY=10
~ DAYS
~ AVERAGE DELAY FOR QA

17. ARTJBM=(MDRPTN+CUMMD-JBSZMD)/DAJMBD
~ MAN-DAYS/DAY
~ RATE OF ADJUSTING THE JOB SIZE IN MAN-DAYS

18. ASIMDLY=80
~ DAYS
~ AVERAGE ASSIMILATION DELAY
19. ASIMRT=WFNEW/ASIMDY
   ~ PEOPLE/DAY
   ~ ASSIMILATION RATE OF NEW EMPLOYEES

20. ASSPRD=PJDPRD*WTPJDP+PRDPRD*(1-WTPJDP)
   ~ TASKS/MAN-DAY
   ~ ASSUMED PRODUCTIVITY

21. AVEEMP=673
   ~ DAYS
   ~ AVERAGE EMPLOYMENT TIME

22. BDFXGR=RWRATE*PBADFX
   ~ ERRORS/DAY
   ~ BAD FIXES GENERATE RATE

23. BRKDTM=INTEG((TIME STEP*(1/TIME STEP))*(MAX(BRKDTM,IF THEN ELSE
   OVDTH=0,(TIME+TIME STEP,0))-BRKDTM))/TIME STEP,-1)
   ~ TIME OF LAST EXHAUSTION BREAKDOWN

24. CELNWH=FTEXWF*MNHPXS
   ~ MEN
   ~ CEILING ON NEW HIREES

25. CELTWF=CELNWH+WFEWP
   ~ PEOPLE
   ~ CEILING ON TOTAL WORKFORCE

26. CMDSI=CMTKDV*DSIPRTK
   ~ TASKS
   ~ CUMULATIVE TASKS DEVELOPED

27. CMDSI KDSI=CMDSI/1000
   ~ TASKS
   ~ TOTAL KDSI COMPLETED

28. CMDVMD=INTEG((TIME STEP*DMPDVT*(1-FREFTS))/TIME STEP,0)
   ~ MAN DAYS
   ~ CUMULATIVE DEVELOPMENT MAN-DAYS
29. CMERD=INTEG(ERRDRT,0)
   ~ ERRORS
   ~ CUMULATIVE ERRORS DETECTED

30. CMERD KDSI=CMERD*(1000/(CMDSI+0.01))
    ~ DEFECTS/KDSI
    ~ DEFECT DENSITY PER KDSI

31. CMERES=INTEG(ERRSRT,0)
    ~ ERRORS
    ~ CUMULATIVE ERRORS THAT ESCAPED

32. CMQAMD=INTEG(DMPQA,0)
    ~ MAN-DAYS
    ~ CUMULATIVE QA MAN-DAYS

33. CMRWED=INTEG(RWRATE,0)
    ~ ERRORS
    ~ CUMULATIVE REWORKED ERRORS DURING DEVELOPMENT

34. CMRWET=INTEG((DCRTP+DCRTAE),0)
    ~ ERRORS
    ~ CUMULATIVE ERRORS REWORKED IN TESTING PHASE

35. CMRWMD=INTEG(DMPRW,0)
    ~ MAN DAYS
    ~ CUMULATIVE REWORK MAN-DAYS

36. CMTKDV=INTEG(SDVRT,0)
    ~ TASKS
    ~ CUMULATIVE TASKS DEVELOPED

37. CMTRMD=INTEG(DMPTRN,0)
    ~ CUMULATIVE TRAINING MAN-DAYS

38. CMTSMD=INTEG(DMPTST,0)
    ~ MAN DAYS
    ~ CUMULATIVE TESTING MAN-DAYS

151
39. COMMOH=TCOMOH(TOTWF)
   ~ DIMENSIONLESS
   ~ COMMUNICATION OVERHEAD
   
40. CRDVWF=TOTDMP-DMPQA
    ~ PERSONS
    ~ CURRENT DEVELOPMENT WORK FORCE
    
41. CRQAWF=(INTEGER(100*DMPQA))/100
    ~ PERSONS
    ~ CURRENT QA WORK FORCE
    
42. CRRWWF=DMPRW
    ~ PEOPLE
    ~ CURRENT REWORK WORK FORCE
    
43. CTRLSW=1
    ~ Zero or One
    ~ CONTROL SWITCH ... ALLOWS US TO TEST POLICY OF NO
      OVERWORK
    
44. CUMERG=INTEG(ERRGRT,0)
    ~ ERRORS
    ~ CUMULATIVE ERRORS GENERATED DIRECTLY DURING WORKING
    
45. CUMMD=INTEG(TOTDMP,0.0001)
    ~ MAN DAYS
    ~ CUMULATIVE MAN-DAYS EXPENDED
    
46. CUMMD TD=CUMMD-CMQAMD
    ~ PERSON DAYS
    ~ PROGRAMMING PERSON DAYS EXPENDED TO DATE
    
47. CUMTKT=INTEG(TSRATE,0)
    ~ TASKS
    ~ CUMULATIVE TASKS TESTED
48. CUMTQA=INTEG((QART-TSRATE),0)
   ~ TASKS
   ~ CUMULATIVE TASKS QA'ED

49. DAJBMD=TDAJMD(TIMERMB)
   ~ DAYS
   ~ DELAY IN ADJUSTING JOB'S SIZE IN MAN DAYS

50. DCRTAE=MIN(TSRATE*AERRDS,UDAVER/TIME STEP)
    ~ ERRORS/DAY
    ~ DETECTION/CORRECTION RATE OF ACTIVE ERRORS

51. DCRTPE=MIN(TSRATE*PERRDS,UDPVER/TIME STEP)
    ~ ERRORS/DAY
    ~ DETECT/CORRECT RATE OF PASSIVE ERRORS

52. DESECR=ACTIVE INITIAL(DTCERR/DESRWD,0)
    ~ ERRORS/DAY
    ~ DESIRED ERROR CORRECTION RATE

53. DESRWD=15
    ~ DAYS
    ~ DESIRED REWORK DELAY

54. DEVMD=INITIAL(DEVPRT*TOTMD)
    ~ DEVELOPMENT MAN DAYS

55. DEVPRT=0.8
    ~ PERCENT OF EFFORT ASSUMED NEEDED FOR DEVELOPMENT

56. DLINCT=10
    ~ DAYS
    ~ AVERAGE DELAY IN INCORPORATING DISCOVERED TASKS

57. DMPATR=TOTDMP-DMPTRN
    ~ MAN-DAYS/DAY
    ~ DAILY MANPOWER AVAILABLE AFTER TRAINING
58. DMPDVT = DMPSWP - DMPRW
   ~ MAN-DAYS/DAYS
   ~ DAILY MANPOWER FOR DEVELOPMENT/TESTING

59. DMPQA = MIN(((FRMPQA/100)*TOLDMP),0.9*DMPATR)
   ~ MAN-DAYS/DAY
   ~ DAILY MANPOWER ALLOCATED FOR QA

60. DMPRW = ACTIVE INITIAL(MIN((DESECR*PRWMPE),DMPSWP),0)
   ~ MAN-DAYS/DAY
   ~ DAILY MANPOWER ALLOCATED FOR REWORK

61. DMPSDV = DMPDVT*(1-FREFTS)
   ~ MAN-DAYS/DAY
   ~ DAILY MANPOWER FOR SOFTWARE DEVELOPMENT

62. DMPSWP = DMPATR - DMPQA
   ~ MAN-DAYS/DAY
   ~ DAILY MANPOWER FOR SOFTWARE PRODUCTION

63. DMPTRN = WFNEW*TRPNHR
   ~ MAN DAYS/DAY
   ~ DAILY MANPOWER FOR TRAINING

64. DMPTST = DMPDVT*FREFTS
   ~ MAN DAYS/DAY
   ~ DAILY MANPOWER FOR TESTING

65. DSIPTK = 60
   ~ DSI PER TASK

66. DTCERR = INTG((ERRDRT-RWRATE),0)
   ~ ERRORS
   ~ DETECTED ERRORS

67. ERRDRT = MIN(PERDRT,PTDTER/TIME STEP)
   ~ ERRORS/DAY
   ~ ERROR DETECTION RATE
68. ERRDSY=ANERPT*1000/DSIPTK
   ~ ERRORS/KDSI
   ~ ERROR DENSITY

69. ERRGRT=SDVRT*ERRPTK
   ~ ERRORS/DAY
   ~ ERROR GENERATION RATE

70. ERRPTK=NERPTK*MERGSP*MERGWM
   ~ ERRORS/TASK
   ~ ERRORS PER TASK

71. ERRSRT=QART*ANERPT
   ~ ERRORS/DAY
   ~ ERROR ESCAPE RATE

72. EWKRTS=IF THEN ELSE(WKRTS >= AFMDPJ,1,0.75)
   ~ DIMENSIONLESS
   ~ EFFECT OF WORK RATE SOUGHT

73. EXHDDY=20
   ~ DAYS
   ~ EXHAUSTION DEPLETION DELAY TIME

74. EXHLEV=INTEG((RIEXHL-RDEXHL),0)
   ~ EXHAUST UNITS
   ~ EXHAUSTION LEVEL

75. EXPTRR=MIN(WFEXP/TIME STEP,TRNFRT-NEWTRR)
   ~ PEOPLE/DAY
   ~ EXPERIENCED EMPLOYEES TRANSFER RATE

76. EXSABS=MAX(0,(TEXABS(TMDPSN/MDRM)*MDRM-TMDPSN))
   ~ MAN-DAYS
   ~ MAN-DAY EXCESSES THAT WILL BE ABSESSED

77. FADHWO=TFAHWO(RSZDCT/(MSZTWO+0.001))
   ~ FRACTION OF ADDITIONAL TASKS ADDING TO MAN-DAYS
78. FINAL TIME=IF THEN ELSE(PJBAWK >= 0.995,Time,MAXLEN)
   ~ ~ | 

79. FNOCOST=INTEG((TIME STEP*TOTDMP*IF THEN ELSE(PJBAWK >= 0.995,0,1))/TIME STEP,0)
    ~ MAN-DAYS
    ~ FINAL COST IN MAN-DAYS
    | 

80. FNERD=INTEG((TIME STEP*ERRDRT*IF THEN ELSE(PJBAWK >= 0.995,0,1))/TIME STEP,0)
    ~ ERRORS
    ~ CUMULATIVE ERRORS DETECTED
    | 

81. FNERES=FNERG-FNERD
    ~ ERRORS
    ~ ERRORS THAT ESCAPED QA
    | 

82. FNERG=INTEG((TIME STEP*ERRGRT*IF THEN ELSE(PJBAWK >= 0.995,0,1))/TIME STEP,0)
    ~ ERRORS
    ~ CUMULATIVE ERRORS GENERATED
    | 

83. FNERR=INTEG(((IF THEN ELSE(PJBAWK >= 0.995,FNERR,ALESER)/TIME STEP)-FNERR/TIME STEP),0)
    ~ ~ | 

84. FNPRDT=100*FNERD/Max(1,FNERG)
    ~ Percent
    ~ PERCENT DETECTED
    | 

85. FNQAMD=INTEG((TIME STEP*DMPQA*IF THEN ELSE(PJBAWK >= 0.995,0,1))/TIME STEP,0)
    ~ MAN-DAYS
    ~ CUMULATIVE QA MAN-DAYS
    | 

86. FNRWMD=INTEG((TIME STEP*DMPRW*IF THEN ELSE(PJBAWK >= 0.995,0,1))/TIME STEP,0)
    ~ MAN-DAYS
    ~ CUMULATIVE REWORK MAN-DAYS
    | 

87. FNTIME=INTEG(IF THEN ELSE(PJBAWK >= 0.995,0,1),0)
    ~ ~ | 

156
88. FNTRMD=INTEG((TIME STEP*DMPTRN*IF THEN ELSE(PJBAWK >= 0.995,0,1))/TIME STEP,0)
   ~
   ~ CUMULATIVE TRAINING MAN-DAYS
          |

89. FRAERR=TFRAER(PJBAWK)
~ DIMENSIONLESS
~ FRACTION OF ESCAPING ERRORS THAT WILL BE ACTIVE
          |

90. FREFTS=TFEFTS(TSKPRM/PJBSZ)
~ DIMENSIONLESS
~ FRACTION OF EFFORT FOR SYSTEM TESTING
          |

91. FRMPQ1=FRMPQA
~ PERSONS
~ FRACTION OF MANPOWER FOR Q1
          |

92. FRMPQA=GAME(10)
~ PERSONS
~ FRACTION OF MANPOWER ALLOCATED FOR QA
          |

93. FRWFEX=(INTEGR((WFEXP/TOTWF)*10000))/10000
~ DIMENSIONLESS
~ FRACTION OF WF THAT IS EXPERIENCED
          |

94. FRWFEX PCT=FRWFEX*100
~ DIMENSIONLESS
~ FRACTION OF WF THAT IS EXPERIENCED
          |

95. FTEQWF=TOTWF*ADMPPS
~ PERSONS
~ FULL TIME EQUIVALENT WF
          |

96. FTEXWF=WFEXP*ADMPPS
~ MEN
~ FULL-TIME-EQUIVALENT EXPERIENCED WF
          |
97. HIREDY=40
   ~ DAYS
   ~ HIRING DELAY
   |

98. HIRERT=MAX(0,WFGAP/HIREDY)
   ~ PEOPLE/DAY
   ~ HIRING RATE
   |

99. INDCDT=Time+TIMEPR
   ~
   ~ INDICATED COMPLETION DATE
   |

100. INITIAL TIME=0
   ~
   ~ ASSUMED WITH FINAL_TIME = LENGTH
   |

101. INSPRD=(SDVRT*TIME STEP)/TOTDMP
   ~
   ~ INSTANTANEOUS PRODUCTIVITY
   |

102. INUDST=0.5
   ~ DIMENSIONLESS
   ~ INITIAL UNDERSTAFFING FACTOR
   |

103. IPRJSZ=INITIAL((RJBDSI)*(1-UNDEST))
   ~ DSI
   ~ INITIAL PROJECT SIZE IN DSI
   |

104. IRDVDT=(RTINCT/ASSPRD)*(FADHWO)
   ~
   ~ RATE OF INCREASE IN DEVELOPMENT MAN- DAYS DUE TO DISCOVERED TASKS
   |

105. IRTSDT=(RTINCT/PRTPRD)*(FADHWO)
   ~ MD/D
   ~ RATE OF INCREASE IN TESTING MAN DAYS DUE TO DISCOVERED TASKS
   |

106. JBSZMD=TOTMD1
   ~ MAN DAYS
   ~ TOTAL JOB SIZE IN MAN DAYS
   |
107. MAERED=TMERED(SMOOTH(AERRDS*1000/DSIPTK,TSAEDS))
    ~
    ~ MULTIPLIER TO ACTIVE ERROR REGENERATION DUE TO
    ~ ERROR DENSITY (DIMEN
    ~
108. MAXLEN=1000
    ~ ~
109. MAXMHR=INITIAL(1)
    ~ DIMENSIONLESS
    ~ MAXIMUM BOOST IN MAN-HOURS
110. MAXSHR=(OVWDTH*FTEQWF*MAXMHR)*WTOVWK
    ~ MAN-DAYS
    ~ MAXIMUM SHORTAGE IN MAN-DAYS THAT CAN BE HANDLED
111. MDEFED=TMDFED(ERRDSY)
    ~ DIMENSIONLESS
    ~ MULTIPLIER TO DETECTION EFFORT DUE TO ERROR DENSITY
112. MDHDL=IF THEN ELSE(PMDSHR >= 0,MIN(MAXSHR,PMDSHR),-EXSABS)*
    ~ CTRLSW
    ~ MAN-DAYS
    ~ MAN-DAYS THAT WILL BE HANDLED OR ABSORBED
113. MDPNNT=TSKPRM/ASSPRD
    ~ MAN-DAYS
    ~ MAN DAYS PERCEIVED STILL NEEDED FOR NEW TASKS
114. MDPNRW=DTCERR*PRWMP
    ~
    ~ MAN DAYS PERCEIVED NEEDED FOR REWORKING ALREADY
    ~ DETECTED ERRORS (MD
115. MDPNTS=TSTPRM/PRTPRD
    ~ MAN-DAYS
    ~ MAN DAYS PERCEIVED STILL NEEDED FOR TESTING
116. MDPRNT=MAX(0,MDRM-MDPNRW-MDPNTS)
    ~ MAN-DAYS
    ~ MAN DAYS PERCEIVED REMAINING FOR NEW TASKS
117. \[ MDRM = \max(0.0001JBSZMD-CUMMD) \]

\[ \sim \sim \sim \]

118. \[ MDRPTN = MDRM + SHRRPT \]
\[ \sim \text{ MAN DAYS} \]
\[ \sim \text{ MAN DAYS REPORTED STILL NEEDED} \]

\[ \sim \]

119. \[ MDSWCH = 0 \]

\[ \sim \]
\[ \sim \text{ SWITCH 0 OR 1} \]

\[ \sim \]

120. \[ MERGSP = TMEGSP(SCHPR) \]

\[ \sim \]
\[ \sim \text{ MULTIPLIER TO ERROR GENERATION DUE TO SCHEDULE PRESSURE (DIMENSIONLESS)} \]

\[ \sim \]

121. \[ MERGW = TMEGW(FRWFEX) \]

\[ \sim \text{ DIMENSIONLESS} \]

\[ \sim \text{ MULTIPLIER TO ERROR GENERATION DUE TO WORKFORCE MIX} \]

\[ \sim \]

122. \[ MNHPXS = 3 \]

\[ \sim \text{ MEN/MEN} \]

\[ \sim \text{ MOST NEW HIREES PER EXPERIENCED STAFF} \]

\[ \sim \]

123. \[ MODTEX = TMODEX(EXHLEV/MXEXHT) \]

\[ \sim \text{ DIMENSIONLESS} \]

\[ \sim \text{ EFFECT OF EXHAUSTION ON OVERWORK DURATION THRESHOLD} \]

\[ \sim \]

124. \[ MPDMCL = AFMDPJ*(1-COMMOH) \]

\[ \sim \] \text{ MULTIPLIER TO PRODUCTIVITY DUE TO MOTIVATION & COMM LOSSES (DIMENSIONLESS)}

\[ \sim \]

125. \[ MPPTPD = TMPTPD(PJBAWK) \]

\[ \sim \text{ DIMENSIONLESS} \]

\[ \sim \text{ MULTIPLIER TO POTENTIAL PRODUCTIVITY DUE TO LEARNING} \]

126. \[ MPWDEV = TMPDEV(PJBPWK/100) \]

\[ \sim \text{ DIMENSIONLESS} \]

\[ \sim \text{ MULTIPLIER TO PRODUCTIVITY WEIGHT DUE TO DEVELOPMENT} \]

160
127. MPWREX=TMPREX((1-MDPRNT/(JBSZMD-TSSZMD)))
~ DIMENSIONLESS
~ MULTIPLIER TO PRODUCTIVITY WEIGHT DUE TO RESOURCE EXPENDITURE

128. MSZTWO=0.01
~ MAXIMUM RELATIVE SIZE OF ADDITIONS TOLERATED W/O ADDING TO PROJECT

129. MXEXHT=50
~ EXHAUST UNITS
~ MAXIMUM TOLERABLE EXHAUSTION

130. MXSCDX=1e+006
~ DIMENSIONLESS
~ MAX SCHEDULE COMPLETION DATE EXTENSION

131. MXTLCD=INITIAL(MXSCDX*TDEV)
~ DAYS
~ MAXIMUM TOLERABLE COMPLETION DATE

132. NERPK=TNERPK(PJBAWK)
~ ERRORS/KDSI
~ NOMINAL # OF ERRORS COMMITTED PER KDSI

133. NERPTK=NERPK*DSIPTK/1000
~ ERRORS/TASK
~ NOMINAL # OF ERRORS COMMITTED PER TASK

134. NEWTRR=MIN(TRNFRT,WFNEW/TIME STEP)
~ PEOPLE/DAY
~ NEW EMPLOYEES TRANSFER RATE OUT

135. NFMDPJ=0.6
~ DIMENSIONLESS
~ NOMINAL FRACTION OF A MAN-DAY ON PROJECT

137. NOWWDT=TNOWWDT(TIMERM)
~ DAYS
~ NOMINAL OVERWORK DURATION THRESHOLD

161
138. NPWPEX=1
   ~ TSK/M-D
   ~ NOMINAL POTENTIAL PRODUCTIVITY OF EXP EMPLOYEE

139. NPWPNE=0.5
   ~ TSK/M-D
   ~ NOMINAL POTENTIAL PROD OF NEW EMPL.

140. NQAMPE=TNQAPE(PJBAWK)
    ~ MAN-DAYS/ERROR
    ~ NOMINAL QA MANPOWER NEEDED TO DETECT AVERAGE ERROR

141. NRWMPE=TNRWME(PJBAWK)
    ~ MAN-DAYS/ERROR
    ~ NOMINAL REWORK MANPOWER NEEDED PER ERROR

142. NWRADY=TNWRAD(TIMERM)
    ~ DAYS
    ~ NORMAL WORK RATE ADJUSTMENT DELAY

143. OVWDTH=NOVWDT*MODTEX
    ~ DAYS
    ~ OVERWORK DURATION THRESHOLD

144. PBADFX=0.075
    ~ FRACTION
    ~ PERCENT BAD FIXES

145. PBWKRS=IF THEN ELSE(PMDSHR >= 0,(MDHDL/(FTEQWF*(OVWDTH + 0.0001)),(MDHDL/(TMDPSN-MDHDL+0.0001))))
    ~ PERCENT
    ~ PERCENT BOOST IN WORK RATE SOUGHT

146. PDEVRC=ACTIVE INITIAL(SMOOTH(MAX(((100-((MDRPTN-MDPNTS)/(JBSZMD-TSSZMD))*100),PDEVRC),RPTDLY),0)
    ~ PERCENT DEVELOPMENT PERCEIVED COMPLETE %

162
147. PEGRT=(ERRSRT+BDFXGR)*(1-FRAERR)
~ ERRORS/DAY
~ PASSIVE ERRORS GENERATION RATE

148. PERDRT=DMPQA/QAMPNE
~ ERRORS/DAY
~ POTENTIAL ERROR DETECTION RATE

149. PERRDS=UDPVER/(CUMTQA+0.0001)
~ ERRORS/TASK
~ PASSIVE ERROR DENSITY

150. PFMPQA=TPFMQA(PJBAWK)*(1+QO/100)
~ DIMENSIONLESS
~ PLANNED FRACTION OF MANPOWER FOR QA

151. PJBAWK=CMTKV/RJBSZ
~ PERCENT
~ PERCENT OF JOB ACTUALLY WORKED

152. PJBDIS=INITIAL(RJBDSI*(1-UNDEST))
~ PERCEIVED JOB SIZE IN DSI

153. PJBPWK=(CMTKV/PJBSZ)*100
~ PERCENT
~ PERCENT OF JOB PERCEIVED WORKED

154. PJBSZ=INTEG(RTINCT,PJBDIS/DSPDK)
~ TASKS
~ CURRENTLY PERCEIVED JOB SIZE

155. PJBSZT=PJBSZ*DSPIK
~ DSI
~ PERCEIVED JOB SIZE IN LINES OF CODE PERCEIVED JOB SIZE IN DSI

156. PJBSZT KDSI=PJBSZT/1000
~ DSI
~ ESTIMATED SYSTEM SIZE (KDSI)
157. PJPRD=TSKPRM/(MDPRNT+0.1)
    ~ TASKS/MAN-DAY
    ~ PROJECTED DEVELOPMENT PRODUCTIVITY

158. PLTSPD=PJSZ/TSSZMD
    ~ TASKS/MAN-DAY
    ~ PLANNED TESTING PRODUCTIVITY

159. PMDSHR=TMPSN-MDRM
    ~ MAN DAYS
    ~ PERCEIVED SHORTAGE IN MAN DAYS

160. POTPRD=ANPPRD*MPPTPD
    ~ TASKS/MAN-DAY
    ~ POTENTIAL PRODUCTIVITY

161. PRCMPL=(CMDSI/PJSZ)*100
    ~ PERCENT
    ~ PERCENT COMPLETE

162. PRCTDT=100*CMERD/(CUMERG+0.001)
    ~ PERCENT
    ~ PERCENT ERRORS DETECTED

163. PRDFDS=PRERD/(MAX(PRTKDV/1000,0.01))
    ~ DEFECT/KDSI
    ~ PERIOD’S DEFECT DENSITY

164. PRDPER=PRTKDV/PRMD
    ~ DSI PER PERSON-DAY
    ~ PRODUCTIVITY IN 40 DAY PERIOD

165. PRDPRD=CMTKDV/(CUMMD-CMTSMD)
    ~ TASKS/MAN-DAY
    ~ PERCEIVED DEVELOPMENT PRODUCTIVITY

166. PRERD=INTEG((ERRDRT-(PRERD/TIME STEP)*M PULSE(Time,
    1,TIME STEP,TIME STEP,40)),0)
    ~ ERRORS
    ~ DETECTED ERRORS DURING PERIOD
168. PRMD=INTEG((TOTDMP-(PRMD/TIME STEP)*M PULSE(Time, 1,TIME STEP,TIME STEP,40)),0.1)  
~ PERSON-DAY  
~ PERSON DAYS SPENT IN PERIOD

169. PROJDR=GAME(TDEVINI)  
~ DAYS  
~ PROJECT DURATION

170. PRQAMD=INTEG((DMPQA-(PRQAMD/TIME STEP)*M PULSE(Time, 1,TIME STEP,TIME STEP,40)),0)  
~ PERSON-DAYS  
~ QA PERSON-DAYS IN PERIOD

171. PRQAMD PERIOD=PRQAMD/((PRTKDV+0.01)/1000)  
~ PERSON DAYS  
~ QA PERSON DAYS PER KDSI DEVELOPED IN PERIOD

172. PRTKDV=INTEG((SDVRT*DSIPTK-(PRTKDV/TIME STEP)*M PULSE(Time, 1,TIME STEP,TIME STEP,40)),0.1)  
~ TASKS DEVELOPED DURING 40 DAY PERIOD

173. PRTPRD=SMOOTH((IF THEN ELSE(0 >= CUMTKT,PLTSPD,ACTSPD)),TSTSPD)  
~ TASKS/MAN-DAY  
~ PERCEIVED TESTING PRODUCTIVITY

174. PRWMP=INTEG((RWMPPE-PRWMP)/TARMPE,0.5)  
~ MAN-DAYS/ERROR  
~ PERCEIVED REWORK MANPOWER NEEDED PER ERROR

175. PSZDCT=TKDCV/ASSPRD  
~ MAN-DAYS  
~ PERCEIVED SIZE OF DISCOVERED TASKS IN MAN DAYS

176. PTDTER=INTEG((ERRGRT-ERRDRT-ERRSRT),0)  
~ ERRORS  
~ POTENTIALLY DETECTABLE ERRORS

177. PTKTST=CUMTKT/PJBSZ  
~ PERCENT  
~ PERCENT OF TASKS TESTED
178. PTRPTC=ACTIVE INITIAL(SMOOTH((100-(MDRPTN/JBSZMD)*100),RPTDLY),0)
~ PERCENT
~ PERCENT OF TASKS REPORTED COMPLETE
   |
179. PUTDPD=TPUTDD(PJPWYK)
~ 1/DAY
~ PERCENT OF UNDISCOVERED TASKS DISCOVERED PER DAY
   |
180. QAMPNE=QAMPE*(1/MPDMCL)*MDEFED
~ MAN-DAYS/ERROR
~ QA MANPOWER NEEDED TO DETECT AVERAGE ERROR
   |
181. QART=DELAY3(SDVRT,AQADLY)
~ TASKS/DAY
~ FOR QA RATE
   |
182. QO=0
~ QUALITY OBJECTIVE ... NORMAL QO = 0
   |
183. QUITRT=WFEXP/AVEMPT
~ PEOPLE/DAY
~ EXPERIENCED EMPLOYEES QUIT RATE
   |
184. RDEXHL=IF THEN ELSE(0 >= RIXH,EXHLEV/EXHDDY,0)
~ EXHAUST UNITS/DAY
~ RATE OF DEPLETION IN EXHAUSTION LEVEL
   |
185. RIXH=TRIXH1((1-AMDPJ)/(1-NFMDP))
~ EXHAUST UNITS/DAY
~ RATE OF INCREASE IN EXHAUSTION LEVEL
   |
186. RJBDSI=64000
~ REAL JOB SIZE IN DSI
   |
187. RJBDSZ=INITIAL(RJBDSI/DSIPTK)
~ TASKS
~ REAL JOB SIZE IN TASKS
   |
188. RLXTMC=INTEG((IF THEN ELSE(EXHLEV/MXEXHT >= 0.1,1,-RLXTMC/ TIME STEP)-((1/TIME STEP)*RLXTMC*IF THEN ELSE(OVWIDTH = 0,1,0))),0)
   ~ VARIABLE THAT CONTROLS TIME TO DE-EXHAUST

189. RPPROD=PRDPRD*DSIPTK
   ~ DSI/MAN-DAY
   ~ REPORTED PRODUCTIVITY

190. RPTDLY=10
   ~ DAYS
   ~ REPORTING DELAY

191. RSZDCT=PSZDCT/(MDPRNT+0.0001)
   ~ DIMENSIONLESS
   ~ RELATIVE SIZE OF DISCOVERED TASKS

192. RTDSTK=UNDJTK*PUTDPD/100
   ~ TASKS/DAY
   ~ RATE OF DISCOVERING TASKS

193. RTINCT=DELAY3(RTDSTK,DLINCT)
   ~ TASKS/DAY
   ~ RATE OF INCORPORATING DISCOVERED TASKS INTO PROJECT

194. RWMPPPE=NRWMPE/MPDMCL
   ~ MAN-DAYS/ERROR
   ~ REWORK MANPOWER NEEDED PER ERROR

195. RWRATE=DMPRW/RWMPPPE
   ~ ERRORS/DAY
   ~ REWORK RATE

196. SAVEPER=40
   ~ DAYS
   ~ CHANGED VALUE TO 40 DAYS VICE 10.
197. SCHADT=TSHADT(TIMERM)
    ~ DAYS
    ~ SCHEDULE ADJUSTMENT TIME

198. SCHCDT=PROJDR
    ~ SCHEDULE COMPLETION DATE

199. SCHCOM=1
    ~ DIMENSIONLESS
    ~ SCHEDULE COMPRESSION FACTOR

200. SCHPR=(TMDPSN-MDRM)/MDRM
    ~ DIMENSIONLESS
    ~ SCHEDULE PRESSURE

201. SCSWCH=0
    ~ SWITCH 0 OR 1

202. SDVPRD=POTPRD*MPDMCL
    ~ TASKS/MAN-DAY
    ~ SOFTWARE DEVELOPMENT PRODUCTIVITY

203. SDVRT=ACTIVE INITIAL(MIN((DMPSDV*SDVPRD),TSKPRM/ TIME STEP),0)
    ~ TASKS/DAY
    ~ SOFTWARE DEVELOPMENT RATE

204. SHRRPT=PMDSHR-MDHL
    ~ MAN-DAYS
    ~ SHORTAGE REPORTED

205. TADJQA(0,0.1,0.2,0.3,0.4,0.5,0,-0.025,-0.15,-0.35,-0.475,-0.5)
    ~ ~

206. TARMPE=10
    ~ DAYS
    ~ TIME TO ADJUST PRWMPE
207. TCOMOH(0,5,10,15,20,25,30,0,0.015,0.06,0.135,0.24,0.375,0.54) 
    ~ ~ | 

208. TDAJMD(0,20,0.5,3) 
    ~ ~ | 

209. TDEV=INITIAL(SCSWCH*(19*2.5*EXP(0.38*LN(TOTMD/19))*SCHCOM) +  
     (1-SCSWCH)*TDEV1)  
    ~ DAYS  
    ~ TOTAL DEVELOPMENT TIME  
    | 

210. TDEV1=296.79 
    ~ DAYS 
    ~ TIME TO DEVELOP 
    | 

211. TDEVINI=INTEGER(TDEV1) 
    ~ DAYS 
    ~ INITIAL TIME TO DEVELOP 
    | 

212. TEAMSZ=INITIAL((TOTMD/TDEV)/ADMPPS) 
    ~ ~ | 

213. TERMFR(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,0,0,0,0,0.01,0.02,0.03,0.04,0.1,0.3,1) 
    ~ ~ | 

214. TEXABS(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,0,2,0.4,0.55,0.7,0.8,0.9,0.95,1,1,1) 
    ~ ~ | 

215. TFAHWO(0,0.2,0.4,0.6,0.8,1,1.2,1.4,1.6,1.8,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,7,0,9,0.975,1,1) 
    ~ ~ | 

216. TFEFTS(0,0.04,0.08,0.12,0.16,0.2,1,0.5,0.28,0.15,0,0.05,0) ~ ~ | 

217. TFRERA(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,1,1,1,1,1,1,0.95,0.85,0.5,0.2,0.075,0,0) 
    ~ ~ | 

218. TIME STEP=0.5 
    ~ DAYS 
    ~ DT 
    | 

219. TIMEPR=MDRM/(WFS*ADMPPS) 
    ~ DAYS 
    ~ TIME PERCEIVED STILL REQUIRED 
    | 

169
220.  TIMERM=MAX(SCHCDT-Time,0)
    ~ DAYS
    ~   TIME REMAINING
    |  

221.  TKDSCV=INTEG((TIME STEP*TIME STEP)((-1)*TKDSCV+ 
 MAX((TKDSCV+TIME STEP*(RTDSTK-RTINCT)),0))/(TIME STEP,0)
    ~ TASKS
    ~   TASKS DISCOVERED
    |  

222.  TM=Time
    ~  

223.  TMDFED(0,1,2,3,4,5,6,7,8,9,10,50,36,26,17.5,10,4,1.75,1.2,1,1,1)
    ~  

224.  TMDPSN=MDPNNT+MDPNTS+MDPNRW
    ~ MAN-DAYS
    ~   TOTAL MAN DAYS PERCEIVED STILL NEEDED
    |  

225.  TMEGSP(-0.4,-0.2,0,0,2,0.4,0.6,0.8,1,0.9,0.94,1,1.05,1.14,1.24,1.36,1.5)
    ~  

226.  TMEGWM(0,0.2,0.4,0.6,0.8,1,2,1.8,1.6,1.4,1.2,1)
    ~  

227.  TIMERED(0,10,20,30,40,50,60,70,80,90,100,1,1.1,1.2,1.325,1.45,1.6,2,2.5,3,2.5,4.35,6)
    ~  

228.  TMODEX(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,1,0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.1,0)
    ~  

229.  TMPDEV(0.01,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,1,1,1,1,0.975,0.9,0.75,0.5,0)
    ~  

230.  TMPNPE=0.15
    ~ MAN-DAY/ERROR
    ~   TESTING MANPOWER NEEDED PER ERROR
    |  

231.  TMPNPT=(TSTOVH*(DSIPTK/1000)+TMPNPE*(PERRDS+AERRDS))/MPDMCL
    ~ MAN-DAYS/TASK
    ~   TESTING MANPOWER NEEDED PER TASK
    |  

232.  TMPREX(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,1,1,1,1,0.975,0.9,0.75,0.5,0)
    ~  

170
233. TMPMRM=0
   ~
   ~ TIME PARAMETER (= HIREDY+ASIMDY) DAYS
   |

234. MPTPD(0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,1.1,1.1,0.125,1.0325,1.055,1.09,1.15,1.2,1.22,
           1.245,1.25,1.25)
   ~ ~ |

235. TMSTOP=IF THEN ELSE(PJBAW*PULSE(TIME,1,TIME STEP,TIME STEP,40) >= 0.995,TIME,MAXLEN)
   ~
   ~ FINAL TIME
   |

236. TNERPK(0.0,2.0,4.0,6.0,8.1,12.5,20.23.86,21.59,15.9,13.6,12.5)
   ~ ~ |

237. TNOWDT(0,10,20,30,40,50,0,10,20,30,40,50)
   ~ ~ |

238. TNPAPE(0.0,1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.1,0.4,0.4,0.39,0.375,0.35,0.3,
           0.25,0.225,0.21,0.2,0.2)
   ~ ~ |

239. TNRWME(0.0,2.0,4.0,6.0,8.1,10.6,0.575,0.5,0.4,0.325,0.3)
   ~ ~ |

240. TNWRAD(0.5,10,15,20,25,30,2,3.5,5,6.5,8.9,5,10)
   ~ ~ |

241. TOTDMP=TOTWF*ADMPPS
   ~ MAN-DAYS/DAY
   ~ TOTAL DAILY MANPOWER
   |

242. TOTMD=INITIAL(MDSWCH*(((2.4*EXP(1.05*LN(PJBDSI/1000)))*19)*1-UNDESM)+(1-MDSWCH)*TOTMD1)
   ~
   ~ TOTAL MAN DAYS
   |

243. TOTMD1=GAME(TOTMD1INI)
   ~ MAN DAY
   ~ TOTAL MAN DAYS
   |

244. TOTMD1INI=2360
   ~ MAN DAY
   ~ INITIAL ESTIMATED TOTAL MAN DAYS
245. TOTWF=WFNEW+WFEXP
~ PEOPLE
~ TOTAL WF LEVEL

246. TPFMQA(0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,1.5,0.15,0.15,0.15,0.15,0.15,0.15,0.15,0.15,0.15)
~ ~ ~

247. TPUTDD(0.2,0.4,0.6,0.8,1.0,0.0,0.4,2.5,5,10,100)
~ ~ ~

248. TRIHXL(-0.5,-0.4,-0.3,-0.2,-0.1,0.7,1.45058e-009,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.2,5,2,2,1,9,1.6,1.3,1.15,0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.0)
~ ~ ~

249. TRNFRN=MAX(0,-WFGAP/TRNSDY)
~ PEOPLE/DAY
~ TRANSFER RATE OF PEOPLE OUT OF PROJECT

250. TRNSDY=10
~ DAYS
~ TIME DELAY TO TRANSFER PEOPLE OUT

251. TRPNHR=0.2
~ DIMENSIONLESS
~ NUMBER OF TRAINERS PER NEW EMPLOYEE

252. TSAEDS=40
~ DAYS
~ TIME TO SMOOTH ACTIVE ERROR DENSITY (AERRDS)

253. TSHADT(0.5,0.5,5)
~ ~ ~

254. TSKPRM=PBJSZ-CMTKDV
~ TASKS
~ NEW TASKS PERCEIVED REMAINING

255. TSKWK=INTEG((SDVGT-QART),0)
~ TASKS
~ TASKS WORKED

256. TSRATE=MIN(CUMTQA/TIME STEP,DMPTST/TMPNPT)
~ TASKS/DAY
~ TESTING RATE
257.  TSSZMD=INTEG((IRTSDT+(1/TIME STEP)*ARTJBM*IF THEN ELSE
          (FREFTS >= 0.9,1,0)),TSTMD)
     ~
     ~  PLANNED TESTING SIZE IN MAN-DAYS ... BEFORE WE START TESTING
          ~
258.  TSTMD=INITIAL((1-DEVPR)*TOTMD)
     ~
     ~  TESTING MAN DAYS
          ~
259.  TSTOVH=1
     ~ MAN-DAYS/KDSI
     ~  TESTING EFFORT OVERHEAD
          ~
260.  TSTPRM=PJSZ-CUMTKT
     ~ TASKS
     ~  TASKS REMAINING TO BE TESTED
          ~
261.  TSTSPD=50
     ~ DAYS
     ~  TIME TO SMOOTH TESTING PRODUCTIVITY
          ~
262.  Twcwf1(0,0.3,0.6,0.9,1.2,1.5,1.8,2.1,2.4,2.7,3.0,0,0,0.1,0.4,0.85,1.1,1.1,1,1,1)
     ~ ~
263.  Twcwf2(0.86,0.88,0.9,0.92,0.94,0.96,0.98,1.0,0.1,0.2,0.35,0.6,0.7,0.77,0.8)
     ~ ~
264.  UDAVER=INTEG((AEGRT+AERGRT-AERRRT-DCRRTAE),0)
     ~ ERRORS
     ~  UNDETECTED ACTIVE ERRORS
          ~
265.  UDPVER=INTEG((PEGRT+AERRRT-DCRTPE),0)
     ~ ERRORS
     ~  UNDETECTED PASSIVE ERRORS
          ~
266.  UNDESM=0
     ~ FRACTION
     ~  MAN-DAYS UNDERESTIMATION FRACTION
          ~

173
267. UNDEST=0.33
    ~ FRACTION
    ~ TASKS UNDERESTIMATION FRACTION

268. UNDJTK=INTEG(-RTDSTK,RJBSZ-PJBSZ)
    ~ TASKS
    ~ UNDISCOVERED JOB TASKS

269. WCWF=MAX(WCWF1,WCWF2)
    ~ DIMENSIONLESS
    ~ WILLINGNESS TO CHANGE WORKFORCE LEVEL

270. WCWF1=TWCFWF1(TIMERM/WCWFTP)
    ~ DIMENSIONLESS
    ~ WILLINGNESS TO CHANGE WORKFORCE (1)

271. WCWF2=TWCFWF2(SCHCDT/MXTLCD)
    ~ DIMENSIONLESS
    ~ WILLINGNESS TO CHANGE WF (2)

272. WCWFTP=IF THEN ELSE(TMPRMR = 0,HIREDY+ASIMDY,TMPRMR)
    ~ TIME PARAMETER DAYS

273. WFEXP=INTEG((ASIMRT-EXPTRR-QUITRT),WFSTRT)
    ~ PEOPLE
    ~ EXPERIENCED WORKFORCE INITIAL VALUE OF EXPERIENCED
    ~ WORKFORCE LEVEL

274. WFGAP=WFS-TOTWF
    ~ PEOPLE
    ~ WORKFORCE GAP

275. WFINDC=(MDRM/(TIMERM+0.001))/ADMPPS
    ~ PEOPLE
    ~ INDICATED WORKFORCE

276. WFNEED=MIN((WCWF*WFINDC+(1-WCWF)*TOTWF),WFINDC)
    ~ PEOPLE
    ~ WORKFORCE LEVEL NEEDED
277. WFNEW = INTEG((HIRERT-ASIMRT-NEWTRR),0)  
~ PEOPLE  
  ~ NEW WORKFORCE

278. WFS = MIN(CELTWF, WFS1/ADMPPS)  
~ PEOPLE  
  ~ WF SOUGHT

279. WFS1 = GAME(WFSINI)  
~ PEOPLE  
  ~ TOTAL REQUESTED STAFFING LEVEL

280. WFS2 = WFS1  
~ PEOPLE  
  ~ WORK FORCE SOUGHT

281. WFSINI = 2  
~ PEOPLE  
  ~ INITIAL VALUE OF STAFFING LEVEL

282. WFSSTR = WFS1  
~ MEN  
  ~ TEAM SIZE AT BEGINNING OF DESIGN

283. WKRADY = NWRADY*EWKRTS  
~ DAYS  
  ~ WORK RATE ADJUSTMENT DELAY

284. WKRTS = (1+PBWKRS)*NFMDPJ  
~ DIMENSIONLESS  
  ~ WORK RATE SOUGHT

285. WRADJR = (WKRTS-ADFMDPJ)/WKRADY  
~ 1/DAY  
  ~ WORK RATE ADJUSTMENT RATE
286.  WTOVWK=IF THEN ELSE(Time >= BRKDTM+RLXTMC,1,0)
    ~ ZERO OR ONE
    ~ WILLINGNESS TO OVERWORK

287.  WTPJDP=MPWDEV*MPWREX
    ~ DIMENSIONLESS
    ~ WEIGHT TO PROJECTED DEVELOPMENT PRODUCTIVITY
LIST OF REFERENCES


177


INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center......................................................2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, Virginia 22060-6218

2. Library, Code 52..................................................................................2
   Naval Postgraduate School
   Monterey, California 93943-5101

3. Dr. Tarek Abdel-Hamid, Code SM/AH..................................................5
   Department of Systems Management
   Naval Postgraduate School
   Monterey, California 93943

4. Dr. Balasubramaniam Ramesh, Code SM/RA.......................................2
   Department of Systems Management
   Naval Postgraduate School
   Monterey, California 93943

5. Maj Luis Hernandez..............................................................................1
   P.O. Box 1010
   Ft. Meade, Maryland 20755-3010