NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

LEGACY SYSTEM UPGRADE FOR SOFTWARE RISK ASSESSMENT

by

Byron Vernon Terry Alexander

December 2001

Thesis Advisor: Valdis Berzins
Co-Advisor: Michael Murrah

Report Documentation Page

Report Date	Report Type	Dates Covered (from... to)
19Dec2001	N/A	-

Title and Subtitle Legacy Systems Upgrade for Software Risk Assessment	**Contract Number**
	Grant Number
	Program Element Number
Author(s) Byron Alexander	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) Naval Postgraduate School Monterey, California	**Performing Organization Report Number**
Sponsoring/Monitoring Agency Name(s) and Address(es)	**Sponsor/Monitor's Acronym(s)**
	Sponsor/Monitor's Report Number(s)
Distribution/Availability Statement Approved for public release, distribution unlimited	
Supplementary Notes	
Abstract	
Subject Terms	
Report Classification unclassified	**Classification of this page** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** UU
Number of Pages 108	

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 2001	Master's Thesis

4. TITLE AND SUBTITLE LEGACY SYSTEM UPGRADE FOR SOFTWARE RISK ASSESSMENT

5. FUNDING NUMBERS

6. AUTHOR (S) Byron Vernon Terry Alexander

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the U.S. Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT ***(maximum 200 words)***

Over the past 40 years limited progress has been made to help practitioners estimate the risk and the required effort necessary to deliver software solutions. Recent developments improve this outlook, one in particular, the research of Juan Carlos Nogueira. Dr. Nogueira developed a formal model for risk assessment that can be used to estimate a software project's risk when examined against a desired development time-line.

Dr. Nogueira developed his model based on data collected from a series of experiments conducted on the VitéProject simulation. This unique approach provides a starting point towards a proven formal model for risk assessment. Another issue with software development, especially in the Department of Defense (DoD), is dealing with aging legacy software systems. These systems perform the functionality of their design, but their interfaces are obsolete and changing requirements limit their functional usefulness.

This thesis is an exercise in upgrading a legacy system licensed to the DoD, VitéProject, for use with ongoing DoD research that seeks to discern truly quantifiable criteria that can be used to more accurately estimate the length of time needed to complete any software project. Accurately projecting software development times and accurate software development costs have eluded software developers for decades.

14. SUBJECT TERMS Risk Assessment, Software Estimation models, Legacy System, Software Extension			15. NUMBER OF PAGES
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500 **Standard Form 298 (Rev. 2-89)**
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

LEGACY SYSTEM UPGRADE FOR SOFTWARE RISK ASSESSMENT

Byron Vernon Terry Alexander
Lieutenant, United States Navy
B.S., United States Naval Academy, 1994

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December 2001

Author: _____
Byron Vernon Terry Alexander

Approved by: _____
Valdis Berzins, Thesis Advisor

_____
Michael Murrah, Co-Advisor

_____
Chris Eagle, Chairman
Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Over the past 40 years limited progress has been made to help practitioners estimate the risk and the required effort necessary to deliver software solutions. Recent developments improve this outlook, one in particular, the research of Juan Carlos Nogueira. Dr. Nogueira developed a formal model for risk assessment that can be used to estimate a software project's risk when examined against a desired development time-line.

Dr. Nogueira developed his model based on data collected from a series of experiments conducted on the VitéProject simulation. This unique approach provides a starting point towards a proven formal model for risk assessment.

Another issue with software development, especially in the Department of Defense (DoD), is dealing with aging legacy software systems. These systems perform the functionality of their design, but their interfaces are obsolete and changing requirements limit their functional usefulness.

This thesis is an exercise in upgrading a legacy system licensed to the DoD, VitéProject, for use with ongoing DoD research that seeks to discern truly quantifiable criteria that can be used to more accurately estimate the length of time needed to complete any software project. Accurately projecting software development times and accurate software development costs have eluded software developers for decades.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

ACKNOWLEDGEMENTS

First, I wish to express my sincere gratitude to my loving family for all of the love and support.

Second, I wish to express my gratitude to Dr. Valdis Berzins and Richard Riehle for allowing me to pursue this research.

Finally, I wish to express my sincere gratitude to Major Michael Murrah, my co-advisor for his constant guidance, encouragement and mentorship. His perseverance, insight, and feedback were crucial to the timely completion of this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

New technologies procured by the Department of Defense (DoD) are becoming increasingly dependent upon software to operate. New hardware platforms like the Joint Strike Fighter and the planned destroyer (DD(X)), provide undisputable evidence that successful technological advantages are dependent upon the intricate integration of both the software and the hardware. The process of procuring any system for the DoD is based on estimates provided by contractors of how long and how costly project development will be. Currently, the software process is greatly flawed. Time and cost estimates of software projects have consistently been in err, leading to very costly overruns and schedule delays.

One of the greatest challenges facing the DoD is verifying time and cost estimations of software projects. Traditional engineering project estimates are based on measurable and tested criteria, subjecting the estimates submitted by contractors to fierce scrutiny. Unfortunately, software engineering projects are not amenable to quantifiable criteria, leaving DoD with little grounds to challenge development estimates. To solve this problem researchers are attempting to discern truly quantifiable criteria that can be used to more accurately estimate the length of time needed to complete any software project.

Another major problem facing the DoD is legacy systems. The DoD procures systems that meet a need at a given time, and much time and resources are invested into these systems. As technology continues to progress at astounding rates, many of these legacy systems are either no longer supported or are not compatible with the new technologies. DoD is faced with the problem of having to spend great sums of money to acquire redundant technology or to attempt to upgrade the legacy system.

One approach researches are investigating is using VitéProject to establish quantifiable measures for software development. VitéProject is a software based modeling and simulation tool that integrates organizational and work process views of strategic, time-critical projects to systematically engineer organizational projects. The underlying model of VitéProject was developed at Stanford University and has been validated for use to model organizational projects. It is the position of Dr. Juan Carlos Nogueira that the core model of VitéProject can also be applied to software projects. In his dissertation, *A Formal Model for Risk Assessment in Software Projects*, he determined metrics that could be used to estimate a software design timeline and adapted these metrics for use in VitéProject. The VitéProject software, in its current form, does not have the ability to allow for sensitivity analysis or for the collection of statistically significant sample sizes of input scenarios.

B. PURPOSE

The ultimate objective of this thesis is to develop a research tool capable of extending the effectiveness and lifecycle usefulness of legacy software, allow customization by analysts, and maintain the integrity of the original functionality. This goal will be achieved by extending the functionality of VitéProject and adapting it for use on a current DoD project; validating its core model for use with software engineering timeline estimation.

The extended functionality allows the use of VitéProject in the creation of a statistically significant number of simulations. The simulations will enable validation of the VitéProject model for use in modeling software engineering projects. The validated core model will allow for better software project time and cost estimation, thereby potentially saving DoD millions of dollars in cost overrun.

C. SCOPE AND METHODOLOGY

This thesis to extend the functionality of VitéProject will first begin with a review of the software engineering project estimation dissertation and the necessity for the program. Second, follow with a review of the enhancement algorithms and software extensions intended to add additional functionality. Third, continue with the code and documentation, as well as a validation of the extended program. Finally, this thesis will end with conclusions and recommendations for future research.

To achieve this goal of extended functionality of VitéProject there must be some steps taken prior to the programming. First, one must gain thorough understanding of the operation of VitéProject software and its interface with Visio (see Chapter III). Second, one must gain a baseline understanding of software project parameters to be used. Finally, one must understand the mapping of the software project parameters to VitéProject-specific parameters.

The implementation of this extension will provide the user an interface that allows the manipulation of VitéProject parameters. This manipulation will automate pre-simulation parameter changes to the VitéProject scenarios, and will allow for automatic or manual simulations of VitéProject scenarios. Finally, the extended program will capture desired data from the simulations and export the data to Microsoft Excel spreadsheet software, with minor formatting.

II. SUPPORTING RESEARCH

This chapter briefly discusses the research initiative that will directly benefit from the application of this thesis. An introduction to the research initiative is provided, followed by a discussion that details a simple validation of the thesis results.

Dr. Nogueira identifies automatically collectable measures in the software process that serve as project risk indicators. The results of Dr. Nogueira's research initiative suggest that to assess software development risk, one must first determine the duration of a software project and consequently the required development effort (Nogueira, 2000). When these early measures are identified and properly correlated to project risk, a software project manager can more adequately assess software development cost. More accurate projections of the duration and cost of a software project in the requirements phase can save vast amounts of money and time, specifically in larger software projects.

In his dissertation, Dr. Nogueira surmised that the VitéProject Model could be used to accurately model a software engineering process. He implemented this modeling and simulation tool in a limited number of cases. The results, although not statistically significant, illustrated that a correlation could exist between the

VitéProject simulations and his mathematical estimates of software project duration.

The architecture of the proprietary VitéProject software imposes many limitations that must be mitigated prior to use outside of its original domain. Although the internal model is well suited and validated for business analysis, it is our belief that it can be extended to the software domain. However, the current working environment severely limits easy adaptation to the software domain. More on VitéProject limitations can be found in Chapter III.

To enable VitéProject 2.0 to conduct sensitivity analysis and multiple discrete simulations of a scenario, this research will extend the proprietary system. The software extension is accomplished by encapsulating the desired functionality of VitéProject, Visio and Microsoft Excel using Microsoft Visual Basic.

To adapt VitéProject for use in his research, Dr Nogueira first had to adapt his respective model parameters into VitéProject-specific parameters. The parameters Dr. Nogueira selected for measurement were Requirements Volatility, Efficiency and Complexity. The significance of each parameter within the software timeline estimate is not relevant for this thesis. More information on these parameters and their role in the software timeline estimate can be found in (Nogueira, 2000).

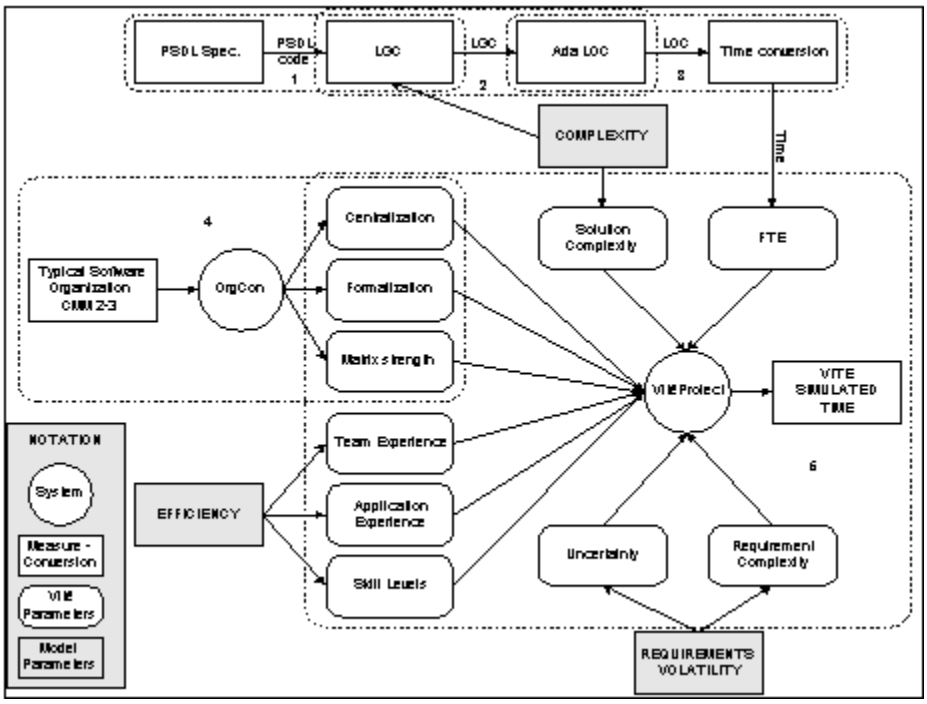


Figure 2.1 Mapping of Parameters to VitéProject

Figure 2.1 is extracted from Dr. Nogueira's dissertation (Nogueira, 2000). It demonstrates the mapping between user-defined parameters and VitéProject-specific parameters. It was incumbent upon this research to make Dr. Nogueira's model parameters into objects themselves, such that changes to the model parameters were echoed into the respective VitéProject-specific parameters.

In extending VitéProject, one has to ensure that the integrity of the internal Virtual Design Team (VDT_2) model is maintained. In his dissertation, Dr. Nogueira presented a table of simulation results that provide a logical validation baseline for this thesis. However, in this thesis, raw data simulations from the original VitéProject program were used to ensure that the extension did not

produce any deviation from the expected output of VitéProject. See Chapter IV, Section C.

III. CODING METHODOLOGY

A. THE MODEL

VitéProject is a commercial modeling and simulation tool based on the Virtual Design Tool (VDT). The VDT was based on contingency theory directed by Dr. Raymond Levitt at Stanford (Jin, 1996). The underlying model of VitéProject, which was based on VDT-2, has been validated on three levels at the Center for Integrated Facility Engineering at Stanford University (CIFE):

micro-level analysis, using toy problems

meso-level analysis, using toy problems and experiments

macro-level analysis, by testing for authenticity, reproducibility, generalizability, and prospective.

A full accounting of VitéProject validation strategy can be found in the dissertation of (Nogueira, 2000) and (Thomsen et al., 1999).

B. VITÉPROJECT

1. VitéProject 2.0

As previously mentioned VitéProject is the commercial software of the VDT-2 framework. Dr. Nogueira used the VitéProject 2.0 in an attempt to apply CPM/Pert modeling techniques to software engineering vis-à-vis VitéProject 2.0. Dr. Nogueira parameterized VitéProject to adequately simulate sixteen different software project developments

(Nogueira, 2000). His research implemented 30 simulations for each scenario development.

VitéProject has several limitations that must be addressed in order to maximize its potential. First, VitéProject limits a maximum of 100 simulated runs for a given scenario. Second, the resulting data is presented in summary format. The summary format provides very limited ability to conduct histograms or a sensitivity analysis. When validating the VDT-2 model for software projects, a larger sample size must be simulated throughout a larger range of software project scenarios.

To change software project scenarios, users must make changes to each individual activity or actors in a given scenario before a new scenario can be simulated. This process, even for the smallest organizational structures, becomes time consuming an inconvenient.

In a communication with Vité Incorporated, technicians informed me that Vité no longer supports VitéProject 2.0. The new product supported by Vité is SimVision 3.0. Although the analytical engine within VitéProject is the same as SimVision, SimVision provides a much needed user interface enhancement. However, for our research, the decomposed architecture of the original VitéProject 2.0 provides the most logical choice for any enhancements.

2. SIMVISION 3.0

SimVision 3.0, while providing a much improved user interface and the ability to increase simulation executions, still does not allow for histogram development and sensitivity analysis. Although SimVision 3.0 is the latest product of Vité and is also based on the VDT-2 research, its lack of a Windows Application Programming Interface (API) interface and a programming language interface make it unusable for the purposes of this research, and will not be discussed.

C. VISIO 5.0

VitéProject modeling and simulation software requires a graphical input of the organization's structure. Visio provides the graphics tool for VitéProject 2.0. Visio opens all VitéProject documents via the VitéProject stencil, which is akin to a template in Microsoft Word. The initial organizational structure for a given scenario, as well as all changes to a scenario or an organization's parameters occur within Visio.

Scenario simulations can be run from Visio, via a VitéProject add-in, or from VitéProject itself. If the scenario is executed from Visio, all data in the scenario drawing is first written to the VitéProject database file prior to the simulation run. If the simulation is run from Vité, the simulation relies solely on the data currently saved in the VitéProject database file.

Data is only saved to the database file from Visio during the course of a simulation. Therefore, as was discovered during this research, changes made in Visio are ignored until the simulation is executed at least once from Visio after any changes. If changes to organization parameters are made directly to the database, the changes will be reflected in the VitéProject scenario simulation, but will not be realized in Visio. Therefore changes made directly to the database will be overwritten if the scenario is run from Visio after the changes.

D. VISUAL BASIC 6.0

Neither VitéProject nor the follow-on, SimVision allow the user to run a simulation for a large number of trial runs and capture the data for each discrete run. Neither allow for parameter updates for multiple objects simultaneously. Furthermore, they do not allow the user to export data to a spreadsheet for more detailed analysis. However, only VitéProject allows for a programming language interface.

Although neither have a Windows API, VitéProject, because of its dependence upon Visio, can be manipulated in a limited way with Visual Basic. In VitéProject, organizational structure, updates to the organization and the scenario, and saving to the database are all done through Visio. VitéProject itself only accesses the data from the database and runs it through the simulator engine

to produce results. The results in turn are written to the database.

The dependence on Microsoft Visio allowed me to write a program in Visual Basic that utilizes the structure provided by Visio and the simulator engine of VitéProject to execute multiple scenarios. The use of Visual Basic also facilitates the export of all data to Microsoft Excel for analysis. Visual Basic, while not the most robust language, serves as the perfect tool for a rapid legacy system upgrade.

E. MICROSOFT EXCEL

Microsoft Excel provides a robust spreadsheet for analyzing the output data. Although other tools may prove better suited for establishing the type of curves and analyses warranted by the research, Excel has the added advantage of being widely used. The wide use of Excel in the area of data collection and statistical analysis ensures that should there be other, better-suited analysis programs, those programs will most likely interface seamlessly with Excel.

$$\left(\frac{OutputVolume(\min s)}{\frac{60\min s}{hour} * \frac{HoursWkd}{Day} * \frac{DaysWkd}{Week}} \right) * 7(days / week) \qquad \text{[Eq. 3-1]}$$

The output data from the simulations is exported to Excel with minimum loss. The areas of output are either cost or volume. Volume is a measure of time in minutes and

cost is measured in dollars. All data will be averaged in the last filled row if its column. The Volume data will be averaged and converted to days. The conversion to days takes into account the hours worked daily and the number of days worked per week. The conversion determines the number of weeks required to complete a task. Eq. 3-1 is used to calculate the number of working days and non-work days.

IV. PROGRAM CODE AND DOCUMENTATION

A. CODE

Visual Basic 6.0 (VBA) was used in the creation of the program. The program consists of three displays. The initial display allows the user to open the Visio file, the associated VitéProject database file, and indicate the storage location for the Excel file. The second display is the main program graphical user interface (GUI). The GUI allows the user to indicate the number of simulation trials, adjust the software project simulation parameters, and choose amongst possible output data to export to Excel. The third display is purely informational in that it illustrates the mapping from the software engineering parameters into the VitéProject parameters, and is called through hyperlinks in the main GUI.

The integrated development environment (IDE) of VBA, by virtue of its drag and drop nature for graphics and GUI, supplies some code as a precursor to the code entered by the programmer. This precursor code is not directly programmed, and therefore shall not be included in this chapter. Appendix A contains the code in its entirety.

1. Initial Display

```
Public dbsMyDB As DAO.Database
Dim szDbLocation As String
Dim szDir As String
Dim VisioApp As Visio.Application
Private Declare Function CloseWindow Lib "user32" (ByVal hwnd As Long)
As Long

' Load the form with the initial instructions
```

```
Private Sub Form_Load()

    szCaption$ = "To begin the program some tasks must be completed:" &
vbCrLf _
                & vbCrLf
    szCaption = szCaption & "(1) Input the location of the Visio file
that "
    szCaption = szCaption & "will be used for the organizations
structure." _
                & vbCrLf
    szCaption = szCaption & "(2) Indicate the location of the Vite
Database "
    szCaption = szCaption & "to be used for the simulation." & vbCrLf
    szCaption = szCaption & "(3) Select directory to save output files"
& vbCrLf
    szCaption = szCaption & vbCrLf
    szCaption = szCaption & "Depressing Cancel during this portion, "
    szCaption = szCaption & "or inputting invalid" & vbCrLf
    szCaption = szCaption & "data will cause the program to exit."

    lblIntro.Caption = szCaption

    Form2.Height = 4530
End Sub

' Cancel button for first frame.  If selected, quit the program

Private Sub cmdCancel1_Click()
    End
End Sub

' Cancel button for second frame.  If selected, end the visio
' application then quit the program

Private Sub cmdCancel2_Click()
    VisioApp.Quit
    End
End Sub

' Cancel button for third frame.  If selected, end the visio
' application then quit the program

Private Sub cmdCancel3_Click()
    VisioApp.Quit
    End
End Sub

' Call the procedure to open visio and allow the user
' to select a file to use with this program

Private Sub cmdContinue1_Click()
    Set VisioApp = OpenVisioDoc
    Form2.Height = 6345
    Label1.Enabled = False
```

```
    cmdContinue1.Enabled = False
    cmdCancel1.Enabled = False
    Frame1.Enabled = False
End Sub

' Call the procedure to open the VitéProject database
' file for use and manipulation

Private Sub cmdContinue2_Click()
    szDbLocation = setDataBase
    Form2.Height = 8520
    Label2.Enabled = False
    cmdContinue2.Enabled = False
    cmdCancel2.Enabled = False
    Frame2.Enabled = False
End Sub

' Select a directory to output the generated Excel files

Private Sub cmdContinue3_Click()
    Dim check As String

    szDir = Dir1.Path
    check = Right(szDir, 1)
    If check <> "\" Then
        szDir = szDir + "\"
    End If
    Form2.Hide
    Form1.Show
End Sub

' Unload the form

Private Sub Form_Terminate()
    Unload Me
End Sub

' Method of the form2 that allows the database file
' location to be retrieved by another form

Public Function returnDBfile() As String
    returnDBfile = szDbLocation
End Function

' Method of the form2 that allows the directory to save
' Excel files be retrieved by another form

Public Function returnDir() As String
    returnDir = szDir
End Function

' Opens the Visio document, with rudimentary error handling

Public Function OpenVisioDoc() As Visio.Application
    Dim errMsg As String
```

```
    Dim Vis50 As Visio.Application
    Dim work As Long

    errMsg = "File Failed to Open." & vbCrLf
    errMsg = errMsg & "Please ensure that the path and " & vbCrLf
    errMsg = errMsg & "file name are correct"
    Set Vis50 = CreateObject("visio.application")

    On Error GoTo Handler:
        Vis50.Application.DoCmd (visCmdFileOpen)
        Set OpenVisioDoc = Vis50
        work = CloseWindow(Vis50.WindowHandle32)
        Exit Function

Handler:
        MsgBox errMsg, vbExclamation + vbOKOnly, "File Error"
        End

End Function

' Method of the form2 that allows Visio Application
' object to be retrieved by another form

Public Function returnVisApp() As Visio.Application
    Set returnVisApp = VisioApp
End Function

' Allows user to open the database and set the Database object
' with rudimentary error handling

Public Function setDataBase() As String
    errMsg = "DB File Failed to Open." & vbCrLf
    errMsg = errMsg & "You must specify a file name " & vbCrLf

    dlgOpen.ShowOpen
    szFileName$ = dlgOpen.FileName

    On Error GoTo Handler
        If dlgOpen.FileTitle <> "" Then
            Set dbsMyDB = DBEngine.OpenDatabase(szFileName, False)
            dbsMyDB.Close
            setDataBase = szFileName
            Exit Function
        End If
Handler:
        MsgBox errMsg, vbExclamation + vbOKOnly, "File Error"
        End

End Function
```

2. Main Display

```
Public dbsMyDB As DAO.Database
Dim g_szDefFileName As String
Dim szCPM$, szDuration$, szSimCost$, szWorkV$, szReV$, szReC$
Dim szCordV$, szCordC$, szWaitV$, szWaitC$, szWorkD$, szWorkW$
```

```
Dim g_ReqLB%, g_EffLB%, g_CompLB%, g_ReqUB%, g_EffUB%, g_CompUB%
Dim g_MultLB%, intWorkV%, intReV%, intCordV%, intWaitV%
Dim rstMyRecordSet As DAO.Recordset
Dim rstRecordSets As DAO.Recordsets
Dim Ex As New Excel.Application
Dim Wrkbk As Excel.Workbook
Dim Vis50 As Visio.Application
Dim DocsObj As Visio.Documents
Dim DocObj As Visio.Document
Dim boolAutoSim As Boolean
Dim boolExit As Boolean
Dim boolBeginEntry As Boolean
Dim szWkbkName As String
Dim wksName As String
Dim g_intColNum As Integer
Dim strArray(50) As String
Dim indexArray(50) As Integer
Dim CompMult As Single

' Automatic Simulation of all combinations is desired.
' When clicked certain controls are no longer available to the
' user and some others are permitted

Private Sub AutoSim_Click()

    ' disable the form combo boxes that allow for updates
    ' to Requirements Volatility, Efficiency, and Complexity

    boolAutoSim = True
    ReqCtl.Enabled = True
    EffCtl.Enabled = True
    CompCtl.Enabled = True

    updateDB.Visible = True
    updateDB.FontBold = False
    updateDB.Caption = "Enter Lower Bound"
    boolBeginEntry = True

    ' disable the labels and the textboxes that allow the user
    ' to pick a Complexity multiple max and the steps

    lblMultMax.Visible = True
    lblMult.Visible = False
    MultCtl.Visible = True
    RunSimulation.Enabled = False

End Sub

' If the complexity Control gets the focus, it is assumed that the
' user will make a change to current value of the controls.  The
' ability to run the simulation will be disabled until the pre-
' run database updates are complete

Private Sub CompCtl_GotFocus()
    updateDB.Visible = True
    RunSimulation.Enabled = False
```

```
End Sub

' If the Efficiency Control gets the focus, it is assumed that the
' user will make a change to current value of the control.  The
' ability to run the simulation will be disabled until the pre-run
' database updates are complete

Private Sub EffCtl_GotFocus()
    updateDB.Visible = True
    RunSimulation.Enabled = False
End Sub

' Give the user a little help when deciding whether to enact an
' automatic simulation or a manual simulation

Private Sub Helper_Click()
    Dim szExplain As String

    szExplain = "Select which mode to run the program" & vbCrLf _
                & vbCrLf
    szExplain = szExplain + "AUTO SIM will simulate all " _
                & "scenarios between what is input" & vbCrLf
    szExplain = szExplain + "            for LOWER BOUND and " _
                & "UPPER BOUND." & vbCrLf & vbCrLf
    szExplain = szExplain + "MANUAL SIM will allow the user to " _
                & "input the parameters for" & vbCrLf
    szExplain = szExplain + "            a single scenario." _
                & vbCrLf & vbCrLf
    szExplain = szExplain + "After inputting the necessary data" _
                & " press RUN SIMULATION." & vbCrLf & vbCrLf
    szExplain = szExplain + "For AUTO SIM, after simulation this" _
                & " program will end" & vbCrLf
    szExplain = szExplain + "and the generated Excel file will " _
                & "be opened." & vbCrLf & vbCrLf
    szExplain = szExplain + "For MANUAL SIM, after simulation " _
                & "control will return" & vbCrLf
    szExplain = szExplain + "once again to the main program for " _
                & "further scenarios."

    MsgBox szExplain, vbQuestion + vbOKOnly, "Program Help"
End Sub

' Allow the user to see the mapping for its components in a
' hyperlink fashion

Private Sub lblComp_DblClick()
    templ.Show
    templ.ShowComplexity
End Sub

' Allow the user to see the mapping for its components in a
' hyperlink fashion

Private Sub lblEff_DblClick()
    templ.Show
    templ.showEfficiency
```

```
End Sub

' Allow the user to see the mapping for its components in a
' hyperlink fashion

Private Sub lblMult_DblClick()
    templ.Show
    templ.showMult
End Sub

' Allow the user to see the mapping for its components in a
' hyperlink fashion

Private Sub lblMultMax_DblClick()
    templ.Show
    templ.showMult
End Sub

' Allow the user to see the mapping for its components in a
' hyperlink fashion

Private Sub lblReqVol_DblClick()
    templ.Show
    templ.showReqVol
End Sub

Private Sub MultCtl_GotFocus()
    If Not boolAutoSim Then
        updateDB.Visible = True
        RunSimulation.Enabled = False
    End If
End Sub

' Provides for input of the high complexity multiple with
' rudimenetary error protection.  Once a valid number is input
' convert the number to minutes

Private Sub MultCtl_LostFocus()
    Dim work As Long
    If Not IsNumeric(MultCtl) Then
        MsgBox "Enter a number for work volume", vbExclamation
        MultCtl = 1
        MultCtl.SetFocus
    ElseIf MultCtl <= 0 Then
        MsgBox "Enter a number greater than 0", vbExclamation
        MultCtl = 1
    End If

End Sub

' If the Requirements Complexity Control gets the focus, it is
' assumed that the user will make a change to current value of
' the control.  The ability to run thesimulation will be
' disabled until the pre run database updates are complete

Private Sub ReqCtl_GotFocus()
```

```
    updateDB.Visible = True
    RunSimulation.Enabled = False
End Sub

' When the form is loaded and instance of Visio will be created as
' well as the Excel file for each run of the program.  When the
' form is loaded certain controls will not be available for
' interaction.  The pre-run database will be set to values
' indicated in the software project parameter comboBoxes.

Private Sub Form_Load()

    ' Variable Declarations

    Dim InitShpsObj As Visio.Shapes
    Dim activity As Visio.shape
    Dim intTick As Integer
    Dim szDrive As String
    Dim work As Long

    ' Variable Initailization
    boolExit = True
    Set Vis50 = Form2.returnVisApp
    g_szDefFileName = Form2.returnDBfile
    szDrive = Form2.returnDir
    Set DocsObj = Vis50.Documents
    Set DocObj = Vis50.ActiveDocument
    Set InitShpsObj = Vis50.ActivePage.Shapes
    setInitScenarioTrials

    'Output Excel file name to the form

    szWkbkName = "output" & "_" & Format(Now, "Medium Date") & "_" _
                & Hour(Now) & Minute(Now) & Second(Now) & ".XLS"
    Set Wrkbk = Ex.Workbooks.Add
    szDrive = szDrive & szWkbkName
    Wrkbk.SaveAs (szDrive)
    boolExit = False
    lblFileSaved.Caption = "Excel File Name: " & szWkbkName
    szWkbkName = szDrive

    ' Set up the Software project estimate parameters Requirements
    ' Volatility, Efficiency, and Complexity.

    RatingComboBox ReqCtl
    RatingComboBox EffCtl
    RatingComboBox CompCtl

    ' Set the inital values for the controls in the pre-run
    ' database, and in the Visio drawing itself

   ' For intTick = 1 To InitShpsObj.Count
   '     Set activity = InitShpsObj.Item(intTick)
        ReqCtl = ReqCtl.List(2)
        EffCtl = EffCtl.List(2)
        CompCtl = CompCtl.List(2)
```

```
   ' Next intTick

    CompMult = MultCtl
    'boolAutoSim = True

End Sub

' When the form is ended ensure that Excel is closed if desired
' and that Visio and its documents are closed

Private Sub Form_Terminate()
    closeExcel
    DocObj.Save
    DocObj.Close
    Vis50.Quit
    End
End Sub

' When Clicked will allow the user to manually change parameters
' for each run

Private Sub ManSim_Click()

    ' All Control Areas Enabled

    ReqCtl.Enabled = True
    EffCtl.Enabled = True
    CompCtl.Enabled = True
    boolAutoSim = False

    ' Disable the controls to set the Max and the step levels

    lblMultMax.Visible = False
    lblSteps.Visible = False
    stepsCtl.Visible = False

    ' Allow for a multiple to use for a high complexity

    lblMult.Visible = True
    MultCtl.Visible = True
    MultCtl.Enabled = True

    ' Set the text for the update database button for use
    ' after parameters are chosen
    updateDB.FontBold = True
    updateDB.Caption = "Update Database"
    updateDB.Visible = True
    RunSimulation.Enabled = False

End Sub

' Command button to run the simulation once the pre-run database
' is updated for a manual simulation or for all database
' combinations for autosimulation

Private Sub RunSimulation_Click()
```

```
    ' First hide the form then check to see if this is an auto or
    ' manual simulation.  After the simulation display a message
    ' box indicator.  If it is manual Prevent a follow-on autosim

    Form1.Hide
    If boolAutoSim Then

    ' If the autosimulate was selected run all of the scenarios
    ' Then end the program and display the Excel spreadsheet

        autoSimulate
        finishMsg "Simulation", True
        Wrkbk.Save
        Ex.Visible = True
        boolExit = True
        Call Form_Terminate
    Else

    'Else Run one scenario input and allow for nore scenarios

        SimRun
        AutoSim.Enabled = False
        finishMsg wksName
    End If
    Form1.Show

End Sub

' Allow for all Shapes to have a string property changed

Public Sub IterativeChange(masterName As String, visID As String, _
           rootTable As String, activity As String, _
           value As String, ID As String)

    ' Variable declarations

    Dim shpsObj As Visio.Shapes
    Dim szActID As String
    Dim shpObj As Visio.shape

    ' Variable initialization

    Set rstMyRecordSet = dbsMyDB.OpenRecordset(rootTable) 'root
    Set shpsObj = Vis50.ActivePage.Shapes

    ' Check all shape objects on the Active Page for any shape that
    ' is either an "Activity" or an "Actor" and change its value in
    ' the pre-run database

    For intTick = 1 To shpsObj.Count

        ' Set the shpObj to the shape in the collection of shapes
        ' at the current number.

        Set shpObj = shpsObj.Item(intTick)
```

```
        ' If that shape is of the type needed to change

        If shpObj.Master.name = masterName Then

            SetCellStr shpObj, visID, value
            szActID = shpObj.Text
            rstMyRecordSet.MoveFirst

            ' Then Search throught the recordset until the specific
            ' shape is found and change its value

            Do While Not rstMyRecordSet.EOF

                 If rstMyRecordSet.Fields(ID).value = szActID Then
                    rstMyRecordSet.Edit
                    rstMyRecordSet(activity) = value
                    rstMyRecordSet.Update
                    rstMyRecordSet.MoveLast
                End If
                rstMyRecordSet.MoveNext
            Loop
        End If
    Next intTick

    rstMyRecordSet.Close

End Sub

'Allow for the Vital Actor properties to be changed

Public Sub ActorPropChange()

    ' Variable Declarations

    Dim shpsObj As Visio.Shapes
    Dim actor As Visio.shape
    Dim intTick As Integer

    ' Variable Initialization

    Set shpsObj = Vis50.ActivePage.Shapes

    IterativeChange "Actor", "ViteAppExperience", "ViteActors", _
                    "AppExperience", EffCtl, "ID"
    IterativeChange "Actor", "ViteSkills", "ViteActorCrafts", _
                    "SkillLevel", EffCtl, "Actor"


End Sub

' Iterate through all activities to change the Requirements
' Volatility values, which is comprised of the Requirements
' Complexity and Uncertainty parameters

Private Sub ReqCtl_XChange()
```

```
    Set dbsMyDB = DBEngine.OpenDatabase(g_szDefFileName, False)
    IterativeChange "Activity", "ViteReqComplexity", _
                   "ViteActivities", "RequirementComplexity", _
                    ReqCtl, "ID"
    IterativeChange "Activity", "ViteUncertainty", _
                    "ViteActivities", "Uncertainty", ReqCtl, "ID"
    dbsMyDB.Close
End Sub

' Iterate through the activities and actors to change the value of
' Efficiency, which is comprised of Team Experience, Application
' Experience, and Skill Levels

Private Sub EffCtl_XChange()

    'Variable declarations and initializations

    sz_TeamTable$ = "ViteTeams"
    sz_FieldExper$ = "Experience"
    sz_VisExp$ = "ViteTeamExperience"
    Set dbsMyDB = DBEngine.OpenDatabase(g_szDefFileName, False)
    Set rstMyRecordSet = dbsMyDB.OpenRecordset(sz_TeamTable)
    Set rstRecordSets = dbsMyDB.Recordsets

    'Change team Experience
    SetCellStr Vis50.ActivePage.PageSheet, sz_VisExp, EffCtl
    rstMyRecordSet.Edit
    rstMyRecordSet(sz_FieldExper) = EffCtl
    rstMyRecordSet.Update
    rstMyRecordSet.Close

    'Change App Experience and Skill levels

    ActorPropChange
    dbsMyDB.Close

End Sub

' Allows for the selection of other output statistics to Excel

Private Sub Tester_Click()
    Dim work As Long, work2 As Integer
    Dim szMyString As String
    Dim boolEnable As Boolean
    If frameStats.Enabled Then
        boolEnable = False
        Height = 4630
    Else
        boolEnable = True
        Height = 6900
    End If

    frameStats.Enabled = boolEnable
    CkBxCPM(0).Enabled = boolEnable
    CkBxDuration(6).Enabled = boolEnable
```

```
    ' Work Cost
    CkBxWorkC(7).Enabled = boolEnable
    ' Work Volume
    CkBxWorkV(1).Enabled = boolEnable
    ' Rework Cost
    CkBxReC(8).Enabled = boolEnable
    ' Rework Volume
    CkBxReV(2).Enabled = boolEnable
    ' Coordination Volume
    CkBxCordV(3).Enabled = boolEnable
    ' Coordination Cost
    CkBxCordC(9).Enabled = boolEnable
    ' Wait Volume
    CkBxWaitV(4).Enabled = boolEnable
    ' Wait Cost
    CkBxWaitC(10).Enabled = boolEnable
    ' Minute worked in a week
    CkBxWeek(11).Enabled = boolEnable
    ' Minutes worked in a day
    CkBxDay(5).Enabled = boolEnable
End Sub

' Iterate through all activities to change the value of Complexity,
' which is comprised of Solution Complexity

Private Sub CompCtl_XChange()
    Set dbsMyDB = DBEngine.OpenDatabase(g_szDefFileName, False)
    IterativeChange "Activity", "ViteSolComplexity", _
                    "ViteActivities", "SolutionComplexity", _
                    CompCtl, "ID"
    dbsMyDB.Close
End Sub



' Used to enter the database and extract the post-run data to
' export to Excel

Private Sub extractAccess()

    'Variable initialization

    ' Table names and table headings in the Vite created database

    sz_TableName$ = "ViteScenarioStatistics"
    sz_Table2Name$ = "ViteScenarios"
    sz_FieldCPM$ = "CPMDuration"
    sz_FieldDuration$ = "Duration"
    sz_FieldWorkCost$ = "WorkCost"
    sz_FieldWorkV$ = "WorkVolume"
    sz_FieldReV$ = "ReworkVolume"
    sz_FieldReC$ = "ReworkCost"
    sz_FieldCordV$ = "CoordinationVolume"
    sz_FieldCordC$ = "CoordinationCost"
    sz_FieldWaitV$ = "WaitVolume"
    sz_FieldWaitC$ = "WaitCost"
```

```
sz_FieldWorkD$ = "WorkDay"
sz_FieldWorkW$ = "WorkWeek"


Set dbsMyDB = DBEngine.OpenDatabase(g_szDefFileName, False)
Set rstMyRecordSet = dbsMyDB.OpenRecordset(sz_TableName)
Set rstRecordSets = dbsMyDB.Recordsets

' If the specified box is checked the extract the
' corresponding data form the post-run database

If CkBxCPM(0) Then
    szCPM = rstMyRecordSet(sz_FieldCPM)
End If

If CkBxDuration(6) Then
   szDuration = rstMyRecordSet(sz_FieldDuration)
End If

If CkBxWorkC(7) Then
    szSimCost = rstMyRecordSet(sz_FieldWorkCost)
End If

If CkBxWorkV(1) Then
    szWorkV = rstMyRecordSet(sz_FieldWorkV)
End If

If CkBxReC(8) Then
    szReC = rstMyRecordSet(sz_FieldReC)
End If

If CkBxReV(2) Then
    szReV = rstMyRecordSet(sz_FieldReV)
End If

If CkBxCordV(3) Then
   szCordV = rstMyRecordSet(sz_FieldCordV)
End If

If CkBxCordC(9) Then
   szCordC = rstMyRecordSet(sz_FieldCordC)
End If

If CkBxWaitV(4) Then
    szWaitV = rstMyRecordSet(sz_FieldWaitV)
End If

If CkBxWaitC(10) Then
    szWaitC = rstMyRecordSet(sz_FieldWaitC)
End If

rstMyRecordSet.Close

' Always retrieve the values for minutes worked in a day and
' in a week.  These are used to convert Volumes to days.
```

```
    Set rstMyRecordSet = dbsMyDB.OpenRecordset(sz_Table2Name)
    Set rstRecordSets = dbsMyDB.Recordsets
    szWorkW = rstMyRecordSet(sz_FieldWorkW)
    szWorkD = rstMyRecordSet(sz_FieldWorkD)
    rstMyRecordSet.Close

    dbsMyDB.Close

End Sub

' If the check Box is checked for a certain data output, then on
' the first run the column names in Excel will be set.  On
' subsequent runs the data is input into the columns

Private Sub inputForExcel(intIndex As Integer)

    'Variable declaration

    Dim Wks As Excel.Worksheet
    Dim Rng As Excel.Range
    Dim intColNum As Integer

    'Variable initialization

    Set Wks = Wrkbk.Worksheets(wksName)
    Set Rng = Wks.UsedRange.Columns
    intColNum = 1

    If CkBxCPM(0) Then
        Wks.Cells(intIndex + 2, intColNum) = szCPM
        intColNum = intColNum + 1
    End If

    If CkBxDuration(6) Then
        Wks.Cells(intIndex + 2, intColNum) = szDuration
        intColNum = intColNum + 1
    End If

    If CkBxWorkC(7) Then
        Wks.Cells(intIndex + 2, intColNum) = szSimCost
        intColNum = intColNum + 1
    End If

    If CkBxWorkV(1) Then
        Wks.Cells(intIndex + 2, intColNum) = szWorkV
        intWorkV = intColNum
        intColNum = intColNum + 1
    End If

    If CkBxReC(8) Then
        Wks.Cells(intIndex + 2, intColNum) = szReC
        intColNum = intColNum + 1
    End If

    If CkBxReV(2) Then
        Wks.Cells(intIndex + 2, intColNum) = szReV
```

```
        intReV = intColNum
        intColNum = intColNum + 1
    End If

    If CkBxCordV(3) Then
        Wks.Cells(intIndex + 2, intColNum) = szCordV
        intCordV = intColNum
        intColNum = intColNum + 1
    End If

    If CkBxCordC(9) Then
        Wks.Cells(intIndex + 2, intColNum) = szCordC
        intColNum = intColNum + 1
    End If

    If CkBxWaitV(4) Then
        Wks.Cells(intIndex + 2, intColNum) = szWaitV
        intWaitV = intColNum
        intColNum = intColNum + 1
    End If

    If CkBxWaitC(10) Then
        Wks.Cells(intIndex + 2, intColNum) = szWaitC
        intColNum = intColNum + 1
    End If

    If CkBxWeek(11) Then
        Wks.Cells(intIndex + 2, intColNum) = szWorkW
        intColNum = intColNum + 1
    End If

    If CkBxDay(5) Then
        Wks.Cells(intIndex + 2, intColNum) = szWorkD
        intColNum = intColNum + 1
    End If

    '  Format the column for decimal places and column fit

    g_intColNum = intColNum
    Rng.NumberFormat = "#.00"
    Rng.AutoFit

End Sub

' Uses the previously created file and sets the initial column names
' and creates and names each new worksheet

Private Sub useExcel(szRunName As String)

    'Variable declaration and initialization

    Dim Wks As Excel.Worksheet
    Static intWksNum As Integer

    ' Sets the column names.  Only the chosen output stats will
    ' be shown
```

```
    szCPM = "CPM Duration"
    szDuration = "Duration"
    szSimCost = "Simulation Cost"
    szWorkV = "Work Volume"
    szReV = "Rework Volume"
    szReC = "Rework Cost"
    szCordV = "Coordination Volume"
    szCordC = "Coordination Cost"
    szWaitV = "Wait Volume"
    szWaitC = "Wait Cost"
    szWorkD = "Work Day"
    szWorkW = "Work Week"

    'boolExit = False
    'Ex.Visible = False

    If intWksNum = 0 Then
        intWksNum = 1
    Else
        intWksNum = intWksNum + 1
    End If

    On Error GoTo ErrHandler
        Set Wks = Wrkbk.Worksheets.Add
        Wks.name = szRunName
        inputForExcel (-1)
        Exit Sub

ErrHandler:
        MsgBox Err.Description, vbExclamation + vbOKOnly, "ERROR"
        Call Form_Terminate

End Sub

' Used to change the high complexity multiple, which is the WORK
' value of the Activities

Public Sub WorkPropChange(rootTable As String, activity As String, _
                          ByVal value As Long)

    ' Variable Declaration

    Dim shpsObj As Visio.Shapes
    Dim shpObj As Visio.shape
    Dim szTestString As String
    Dim intTick As Integer
    Dim szActID As String

    ' Variable initialization

    Set dbsMyDB = DBEngine.OpenDatabase(g_szDefFileName, False)
        Set rstMyRecordSet = dbsMyDB.OpenRecordset(rootTable)
    Set shpsObj = Vis50.ActivePage.Shapes

    ' Iterate through all shapes in the Visio drawing
```

```
    For intTick = 1 To shpsObj.Count
        Set shpObj = shpsObj.Item(intTick)
        If shpObj.Master.name = "Activity" Then
            SetCellInt shpObj, "ViteWorkVolume", CLng(value)
            szActID = shpObj.Text
            rstMyRecordSet.MoveFirst
            Do While Not rstMyRecordSet.EOF

                 If rstMyRecordSet.Fields("ID").value = szActID Then
                    rstMyRecordSet.Edit
                    rstMyRecordSet(activity) = value
                    rstMyRecordSet.Update
                    rstMyRecordSet.MoveLast
                Else
                    rstMyRecordSet.MoveNext
                End If
            Loop
        End If
    Next intTick

    rstMyRecordSet.Close
    dbsMyDB.Close

End Sub

' Used to run the complete automatic simulation from LLL to HHH
' with a complexity multiple.  There is an inner loop iterative
' run that is common to both cases of complexity, that is either
' with or without the multiple

Public Sub autoSimulate()

    ' Variable Declaration

    Dim intUncTick As Integer
    Dim work As Long

    For intUncTick = g_CompLB To g_CompUB Step -1

    ' Outer loop is the complexity.  The inner loop will be
    ' Efficiency and Requirements Volatility

        CompCtl = CompCtl.List(intUncTick)
        Call CompCtl_XChange  ' Set the value of Complexity pre-run
        If CompCtl = CompCtl.List(0) Then

        ' If high complexity, step through multiple from low to high
            For CompMult = g_MultLB To MultCtl Step stepsCtl
                work = Units2Mins(CompMult, "days")
                WorkPropChange "ViteActivities", "WorkVolume", _
                               work
                innerLoop
            Next CompMult

        Else
```

```
        ' Then do the non-high-complexity runs
        ' first ensure the multiple is one
            work = Units2Mins(1, "days")
            WorkPropChange "ViteActivities", "WorkVolume", _
                           work

            innerLoop
        End If

    Next intUncTick

End Sub

Public Sub innerLoop()

    Dim intReqTick As Integer, intSolTick As Integer

    For intReqTick = g_ReqLB To g_ReqUB Step -1
    ' Set Requirements Volatility Parameter for the runs
        ReqCtl = ReqCtl.List(intReqTick)
        Call ReqCtl_XChange

        For intSolTick = g_EffLB To g_EffUB Step -1
        ' Set Efficiency Parameter for the runs
            EffCtl = EffCtl.List(intSolTick)
            Call EffCtl_XChange

            ' Run the simulation
            SimRun

         Next intSolTick

    Next intReqTick

End Sub

' Opens then resident Excel file with a new worksheet then
' runs the simulation for the number of times indicated in the
' TrialsCtl.  Averages each active column of the spreadsheet
' Then closes the resident Excel file

Public Sub SimRun()

    Dim myVite As New ViteProject.Project
    Dim r As Long
    Dim intTick As Integer
    Dim szMyString As String
    wksName = WorksheetName
    useExcel (wksName)

    ' For the number of trials indicated run the sim.
    ' Always run the sim. via Vite using scenario 1
    szMyString = CellStr(Vis50.ActivePage.PageSheet, _
                         "ViteScenName", "S001")

    For intTick = 1 To TrialsCtl
```

```
        r = myVite.Simulate(g_szDefFileName, szMyString)
        extractAccess
        inputForExcel (intTick)
    Next intTick
    averageWks

End Sub

' Sets the worksheet name based on the first letter of the text
' in each of the comboboxes.  Also, for manual simulation, if this
' is a repeat scenario name add an integer identifier to
' distinguish the new worksheet from a previous run

Public Function WorksheetName() As String

    ' Variable declaration

    Dim ReqString As String, UncString As String
    Dim SolString As String, MultString As String
    Dim element As Integer
    Static intWksNum As Integer

    ' Variable initialization

    szDuplicate$ = ""
    szWksName$ = ""
    SolString = Left$(EffCtl, 1)
    ReqString = Left$(ReqCtl, 1)
    UncString = Left$(CompCtl, 1)

    ' If there is a complexity multiple higher than one then put
    ' it in the worksheet name

    If CompMult > 1 Then
        MultString = Str(CompMult)
        szWksName = SolString & ReqString & UncString & MultString
    Else
        szWksName = SolString & ReqString & UncString
    End If

    ' In manual sim mode prevent duplicate worksheet names which
    ' will crash Excel

    If Not boolAutoSim Then
        element = LinearSearch(strArray(), szWksName)
        If element <> -1 Then
            indexArray(intWksNum) = indexArray(element) + 1
            szDuplicate = "(" & Str(indexArray(intWksNum)) & ")"
        Else
            indexArray(intWksNum) = 1
        End If
        strArray(intWksNum) = szWksName
    End If

    WorksheetName = szWksName & szDuplicate
    intWksNum = intWksNum + 1
```

```
End Function

' Function from Access 2000 Programming for Dummies
' This function is used to set the column letter for use in formulas
' created in Excel format

Public Function ColumnLetter(ColNumber)

    Dim Letter1, Letter2
    If ColNumber < 27 Then
        ColumnLetter = Chr(64 + ColNumber)
    Else
        Letter1 = Chr(Int(ColNumber / 26) + 64)
        Letter2 = Chr((ColNumber Mod 26) + 64)
        ColumnLetter = Letter1 + Letter2
    End If

End Function

' Places a space between the bulk data and a line used to average
' the individual columns that were selected for display

Public Sub averageWks()

    ' Varible declarations

    Dim Wks As Excel.Worksheet
    Dim Rng As Excel.Range
    Dim intColNum As Integer

    ' Variable initialization

    RowMax% = TrialsCtl + 2
    intUnitRow% = 2
    intColStart% = 1
    szUnitLbl$ = "(Minutes)"
    szAvgLbl$ = "(Days)"
    Set Wks = Wrkbk.Worksheets(wksName)
    Set Rng = Wks.UsedRange.Columns
    Rng.NumberFormat = "#.00"

    ' All columns have some formatting involved

    If CkBxCPM(0) Then
        Set Rng = Wks.Cells(intUnitRow, intColStart)
        Rng.Formula = szUnitLbl
        Rng.HorizontalAlignment = xlCenterAcrossSelection
        Set Rng = Wks.Cells(RowMax + 1, intColStart)
        Rng.Formula = szAvgLbl
        Rng.HorizontalAlignment = xlCenterAcrossSelection
        Set Rng = Wks.Cells(RowMax + 2, intColStart)
        Rng.Formula = "=(Average(" & ColumnLetter(intColStart) _
                    & "3:" & ColumnLetter(intColStart) & RowMax _
                    & ") * 7)/" & szWorkW
            Rng.NumberFormat = "#.00"
```

```
    intColStart = intColStart + 1
End If

If CkBxDuration(6) Then
    Set Rng = Wks.Cells(intUnitRow, intColStart)
    Rng.Formula = szUnitLbl
    Rng.HorizontalAlignment = xlCenterAcrossSelection
    Set Rng = Wks.Cells(RowMax + 1, intColStart)
    Rng.Formula = szAvgLbl
    Rng.HorizontalAlignment = xlCenterAcrossSelection
    Set Rng = Wks.Cells(RowMax + 2, intColStart)
    Rng.Formula = "=(Average(" & ColumnLetter(intColStart) _
                & "3:" & ColumnLetter(intColStart) & RowMax _
                & ") * 7)/" & szWorkW
        Rng.NumberFormat = "#.00"
    intColStart = intColStart + 1
End If

For intColNum = intColStart To g_intColNum - 1
    Set Rng = Wks.Cells(RowMax + 2, intColNum)
    Rng.Formula = "=Average(" & ColumnLetter(intColNum) _
                & "3:" & ColumnLetter(intColNum) & RowMax & ")"
        Rng.NumberFormat = "#.00"
Next intColNum

If CkBxWorkV(1) Then
    Set Rng = Wks.Cells(intUnitRow, intWorkV)
    Rng.Formula = szUnitLbl
    Rng.HorizontalAlignment = xlCenterAcrossSelection
    Set Rng = Wks.Cells(RowMax + 1, intWorkV)
    Rng.Formula = szAvgLbl
    Rng.HorizontalAlignment = xlCenterAcrossSelection
    Set Rng = Wks.Cells(RowMax + 2, intWorkV)
    Rng.Formula = "=(Average(" & ColumnLetter(intWorkV) _
                & "3:" & ColumnLetter(intWorkV) & RowMax _
                & ") * 7)/" & szWorkW
        Rng.NumberFormat = "#.00"
End If

If CkBxReV(2) Then
    Set Rng = Wks.Cells(intUnitRow, intReV)
    Rng.Formula = szUnitLbl
    Rng.HorizontalAlignment = xlCenterAcrossSelection
    Set Rng = Wks.Cells(RowMax + 1, intReV)
    Rng.Formula = szAvgLbl
    Rng.HorizontalAlignment = xlCenterAcrossSelection
    Set Rng = Wks.Cells(RowMax + 2, intReV)
    Rng.Formula = "=(Average(" & ColumnLetter(intReV) _
                & "3:" & ColumnLetter(intReV) & RowMax _
                & ") * 7)/" & szWorkW
        Rng.NumberFormat = "#.00"
End If

If CkBxCordV(3) Then
    Set Rng = Wks.Cells(intUnitRow, intCordV)
    Rng.Formula = szUnitLbl
```

```
        Rng.HorizontalAlignment = xlCenterAcrossSelection
        Set Rng = Wks.Cells(RowMax + 1, intCordV)
        Rng.Formula = szAvgLbl
        Rng.HorizontalAlignment = xlCenterAcrossSelection
        Set Rng = Wks.Cells(RowMax + 2, intCordV)
        Rng.Formula = "=(Average(" & ColumnLetter(intCordV) _
                    & "3:" & ColumnLetter(intCordV) & RowMax _
                    & ") * 7)/" & szWorkW
            Rng.NumberFormat = "#.00"
    End If

    If CkBxWaitV(4) Then
        Set Rng = Wks.Cells(intUnitRow, intWaitV)
        Rng.Formula = szUnitLbl
        Rng.HorizontalAlignment = xlCenterAcrossSelection
        Set Rng = Wks.Cells(RowMax + 1, intWaitV)
        Rng.Formula = szAvgLbl
        Rng.HorizontalAlignment = xlCenterAcrossSelection
        Set Rng = Wks.Cells(RowMax + 2, intWaitV)
        Rng.Formula = "=(Average(" & ColumnLetter(intWaitV) _
                    & "3:" & ColumnLetter(intWaitV) & RowMax _
                    & ") * 7)/" & szWorkW
            Rng.NumberFormat = "#.00"
    End If

End Sub

' If there is a resident Excel file to close save it and  its
' corresponding worksheets then Exit Excel and indicate that
' there is no resident Excel file

Private Sub closeExcel()
    If Not boolExit Then
        Wrkbk.Save
        Ex.Quit
        Set Ex = Nothing
        boolExit = True
    End If

End Sub

' When the Exit button is clicked ensure terminate the form

Private Sub ExitCmd_Click()
    Form_Terminate
    End
End Sub

' Display appropriate message for the end of the sim run

Public Sub finishMsg(typeRun As String, Optional ifMore As Boolean)

    Dim szText As String
    szText = "DONE!" & vbCrLf
    szText = szText & typeRun & " completed."
    If MoreBool Then
```

```
        szText = szText & "Press OK, the program will Exit " _
                 & vbCrLf
        szText = szText & "and the Excel Spreadsheet will Open"
    End If
    MsgBox szText, vbInformation + vbOKOnly, "Program Status"

End Sub

' An array that maintains the list of worksheets that have been
' already used will be searched with this function

Public Function LinearSearch(inArray() As String, inKey As String) _
               As Integer

    Dim ArrIndex As Integer

    For ArrIndex = UBound(inArray) To LBound(inArray) Step -1

        If inArray(ArrIndex) = inKey Then
            LinearSearch = ArrIndex
            Exit Function
        End If

    Next ArrIndex

    LinearSearch = -1

End Function

' Convert time in specified units to minutes

Public Function Units2Mins(value As Single, unit As String) _
               As Single

    If InStr(unit, "mins") Then
        Units2Mins = value
    ElseIf InStr(unit, "hours") Then
        Units2Mins = value * 60#
    ElseIf InStr(unit, "days") Then
        Units2Mins = value * Vis50.ActivePage.PageSheet.Cells _
                    ("User.ViteScenWorkDay").Result(visNumber)
    ElseIf InStr(unit, "weeks") Then
        Units2Mins = value * Vis50.ActivePage.PageSheet.Cells _
                    ("User.ViteScenWorkWeek").Result(visNumber)
    Else
        Unit2Mins = -1
    End If

End Function

' Initialize combo box with ratings
' From ViteProject template

Public Sub RatingComboBox(ctl As ComboBox)
    ctl.Clear
    ctl.AddItem ("High")
```

```
    ctl.AddItem ("Medium")
    ctl.AddItem ("Low")
End Sub

' Assign integer value to user-defined cell
' This will be used if later users desire to see the changes
' made reflected in Visio
' LT Alexander Added

Public Sub SetCellInt(shape As Visio.shape, name As String, _
                      value As Long)
    If Not shape.CellExists("User." & name, 0) Then
        shape.AddNamedRow visSectionUser, name, 0
    End If
    shape.Cells("User." & name).Formula = value
End Sub

' Return value of specified user-defined cell as string
' This will be used if later users desire to see the changes
' made reflected in Visio
' From ViteProject template

Public Function CellStr(shape As Visio.shape, name As String, _
                        default As String) As String
    If shape.CellExists("User." & name, 0) Then
        CellStr = shape.Cells("User." & name).ResultStr("")
    Else
        CellStr = default
    End If
End Function

'Set specified user-defined cell to string
' This will be used if later users desire to see the changes
' made reflected in Visio
'LT Alexander Added

Public Sub SetCellStr(shape As Visio.shape, name As String, _
                      ByVal value As String)
    skills$ = "ViteSkills"
    craft$ = "Generic.."
    If Not shape.CellExists("User." & name, 0) Then
        shape.AddNamedRow visSectionUser, name, 0
    End If
    If name = skills Then
        value = craft + value
    End If
    shape.Cells("User." & name).Formula = Chr$(34) & value _
                & Chr$(34)
End Sub

' Number of trials to run in a given simulation with rudimentary
' error check

Private Sub TrialsCtl_LostFocus()

    If Not IsNumeric(TrialsCtl) Then
```

```
        MsgBox "Enter a Number greater than 0 ", vbExclamation
        TrialsCtl = 5
        TrialsCtl.SetFocus
    ElseIf TrialsCtl <= 0 Then
        MsgBox "Enter a number greater than 0 ", vbExclamation
        TrialsCtl = 5
        TrialsCtl.SetFocus
    End If

End Sub

' For MANUAL SIMULATION when the comboboxes have been set
' this button is pressed to update the pre-run database, but only
' update if there has been an actual change
' For AUTO SIMULATION this button is used to enter the low end
' and high end scenarios that bracket the simulations


Private Sub updateDB_Click()

    If Not boolAutoSim Then
    ' Update DB if Manual Sim

        Static szReqCtl$, szCompCtl$, szEffCtl$

        'If Requirements Volatility selection has changed
        If Not szReqCtl = ReqCtl Then
            szReqCtl = ReqCtl
            Call ReqCtl_XChange
        End If

        ' If Complexity selection has changed
        If Not szCompCtl = CompCtl Then
            szCompCtl = CompCtl
            Call CompCtl_XChange
        End If

        ' If Efficiency selection has changed
        If Not szEffCtl = EffCtl Then
            szEffCtl = EffCtl
            Call EffCtl_XChange
        End If

        If CompCtl = CompCtl.List(0) Then   'MultCtl.DataChanged

            'Update the Work parameter of Activity
            CompMult = MultCtl

        Else
            ' Ensure that the database multiple is 1
            CompMult = 1
        End If

        work = Units2Mins(CompMult, "days")
        WorkPropChange "ViteActivities", "WorkVolume", work
        updateDB.Visible = False
```

```
        RunSimulation.Enabled = True

    ElseIf boolBeginEntry Then
    ' If Auto sim chosen and the lower bound has not been
    ' entered, then enter the lower bound data

        RunSimulation.Enabled = False
        ManSim.Enabled = False
        g_ReqLB = ReqCtl.ListIndex
        g_EffLB = EffCtl.ListIndex
        g_CompLB = CompCtl.ListIndex
        ' If the complexity is High then allow for a multipe
        If CompCtl = CompCtl.List(0) Then
        ' Its High, therefore set the lowerbound Multiple
            g_MultLB = MultCtl
        Else
            g_MultLB = 1
        End If
        boolBeginEntry = False
        ' change the caption of the control for the next part
        updateDB.FontBold = False
        updateDB.Caption = "Enter Upper Bound"

        lblMultMax.Visible = True
        MultCtl.Visible = True
        MultCtl.Enabled = True
        lblSteps.Visible = True
        stepsCtl.Visible = True
        lblMult.Visible = False

    Else
    ' Auto sim has been chosen and lower bound data has
    ' been entered.  Enter upper bound data.

        g_ReqUB = ReqCtl.ListIndex
        g_EffUB = EffCtl.ListIndex
        g_CompUB = CompCtl.ListIndex
        RunSimulation.Enabled = True
        updateDB.Visible = False
        ReqCtl.Enabled = False
        EffCtl.Enabled = False
        CompCtl.Enabled = False
        stepsCtl.Enabled = False
        MultCtl.Enabled = False
        If Not g_MultLB <= MultCtl Then
            szMsg$ = "The higher bound Multiple is less than the"
            szMsg = szMsg + vbCrLf
            szMsg = szMsg + " lower bound, they will be switched"
            MsgBox szMsg, vbInformation + vbOKOnly, "Invalid Order"
            CompMult = g_MultLB
            g_MultLB = MultCtl
            MultCtl = CompMult
        End If
    End If

End Sub
```

```
'Used to ensure that the number of runs in the Vite program is
'maintained at 1 so that this program can make all the runs

Public Sub setInitScenarioTrials()

    'Variable declarations and initializations

    sz_TeamTable$ = "ViteScenarios"
    sz_FieldExper$ = "Trials"
    Set dbsMyDB = DBEngine.OpenDatabase(g_szDefFileName, False)
    Set rstMyRecordSet = dbsMyDB.OpenRecordset(sz_TeamTable)
    Set rstRecordSets = dbsMyDB.Recordsets

    'Change team Experience

    rstMyRecordSet.Edit
    rstMyRecordSet(sz_FieldExper) = 1
    rstMyRecordSet.Update
    rstMyRecordSet.Close
    dbsMyDB.Close

End Sub
```

3. Information Mapping Display

```
Const TRANSPARENT_BACK = 0
Const TRANSPARENT_FILL = 1
Const BACK_COLOR = 1
Const FILL_COLOR = 0

' Close this form and return to the calling form

Private Sub Command1_Click()
    templ.Hide
    Form1.Show

    shpEff(12).FillStyle = TRANSPARENT_FILL
        shpTeam(5).BackStyle = TRANSPARENT_BACK
        shpApp(2).BackStyle = TRANSPARENT_BACK
        shpSkills(1).BackStyle = TRANSPARENT_BACK

    shpComp(20).FillStyle = TRANSPARENT_FILL
        shpSolComp(3).BackStyle = TRANSPARENT_BACK
        shpFTE(0).BackStyle = TRANSPARENT_BACK

    shpReqV(19).FillStyle = TRANSPARENT_FILL
        shpUnc(8).BackStyle = TRANSPARENT_BACK
        shpReqComp(7).BackStyle = TRANSPARENT_BACK

End Sub

' When the Requirements Volatility link is selected
' illustrate the vite parameters that will be affected
```

```
Public Sub showReqVol()

    shpReqV(19).FillStyle = FILL_COLOR
        shpUnc(8).BackStyle = BACK_COLOR
        shpReqComp(7).BackStyle = BACK_COLOR

End Sub

' When the Efficiency link is selected
' illustrate the vite parameters that will be affected

Public Sub showEfficiency()
    shpEff(12).FillStyle = FILL_COLOR
        shpTeam(5).BackStyle = BACK_COLOR
        shpApp(2).BackStyle = BACK_COLOR
        shpSkills(1).BackStyle = BACK_COLOR
End Sub

' When the Complexity link is selected
' illustrate the vite parameter that will be affected

Public Sub ShowComplexity()
    shpComp(20).FillStyle = FILL_COLOR
        shpSolComp(3).BackStyle = BACK_COLOR
End Sub

' When the "high" complexity Multiple link is selected
' illustrate the vite parameter that will be affected

Public Sub showMult()
    shpFTE(0).BackStyle = BACK_COLOR
End Sub

' Unload the form

Private Sub Form_Terminate()
    Call Command1_Click
    Unload Me
End Sub
```

B. DOCUMENTATION

There are some prerequisites that must be set prior to using this research for analysis. Visio 5.0 or better, VitéProject 2.0, and Excel must be loaded onto the computer designated to run the simulation. Preferably a new VitéProject project with only one scenario should be used to run the analysis. If an older project is to be used, only

Scenario 1 will be manipulated and simulated via this program. It is imperative to ensure Scenario 1 has the correct organization structure and parameter settings (not affected by this program) for the analysis.

Ensure also that if a large sample size is to be used that there is enough storage space for the generated Excel files. All Excel file size limitations still apply. For comparison, the Excel file generated for an execution from LLL to HHH25 (1000 trials each) is in excess of 12Mb. It is also recommended that when simulating large samples the designated computer be used exclusively for the simulation and subsequent storage.

As previously discussed, the first display, Figure 4.1, is designed to set up the program. The analyst, through this display, locates the Visio file containing the organization's structure, the corresponding VitéProject database file, and where to output the analysis data's Excel file.

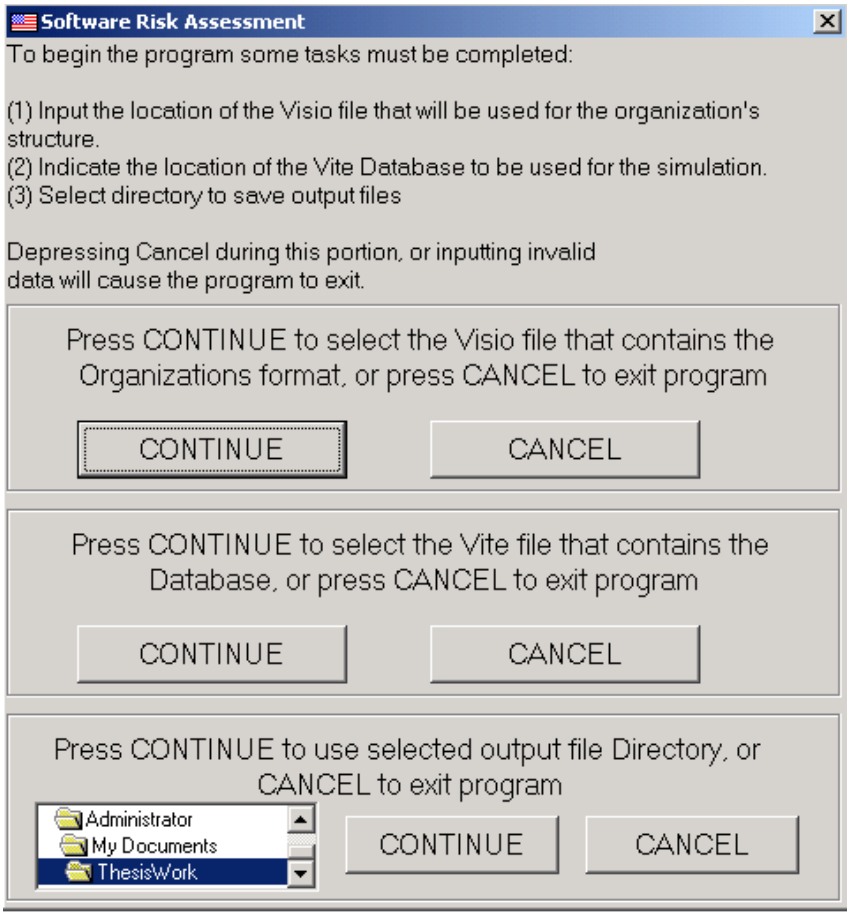


Figure 4.1 Introductory Display

The main form, Figure 4.2 has various user (analyst) interfaces that allow the analyst to set the previously discussed VitéProject parameters for simulation. The illustration is labeled with five areas for ease of discussion. The large underlined numbers do no appear on the actual main display of the program.

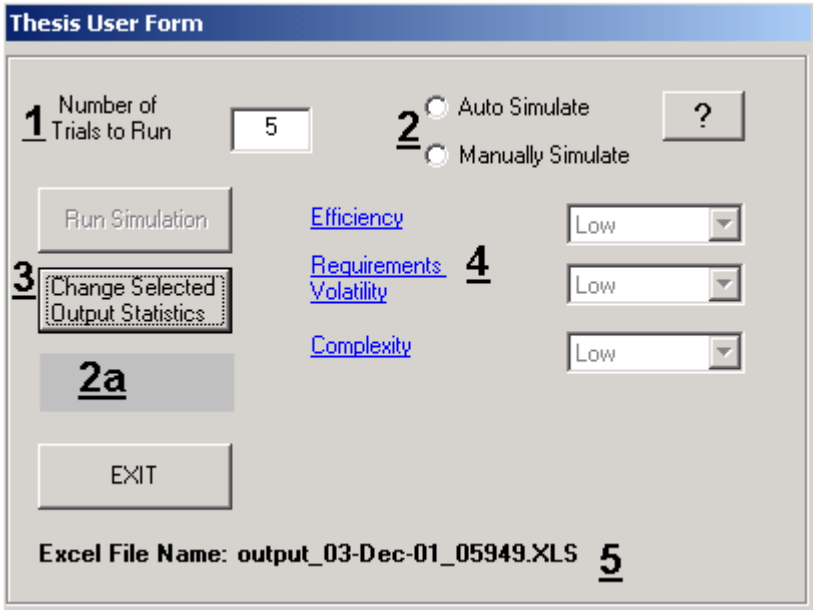


Figure 4.2 Main Display

In Area 1, the user indicates the number of simulation trials for a chosen set of parameters. It is incumbent upon the analyst to remain cognizant of computer storage limitations. Only integer values are accepted.

Area 2 allows the user to determine whether to manually simulate individually defined scenarios, or to simulate scenarios between a designated lower and upper bound. When a manual simulation is executed, the analyst loses the option of automatic simulation for the current instance of the program. When an automatic simulation is initiated, the program terminates upon completion of the simulation. Area 2a is a control that either updates the database prior to a scenario simulation or is used to set the lower and upper bounds for an auto simulation.

The controls in Area 3 are the RUN SIMULATION and CHANGE SELECTED OUTPUT STATISTICS control. The analyst will not be allowed to run a simulation until the pre-run database is updated for a manual simulation or the lower and upper bound are set for the automatic simulation. The analyst may choose to change the statistical information collected in the output Excel file. Three default statistics are initially selected: CPM Duration, Duration, and Work Cost. See Figure 4.3.

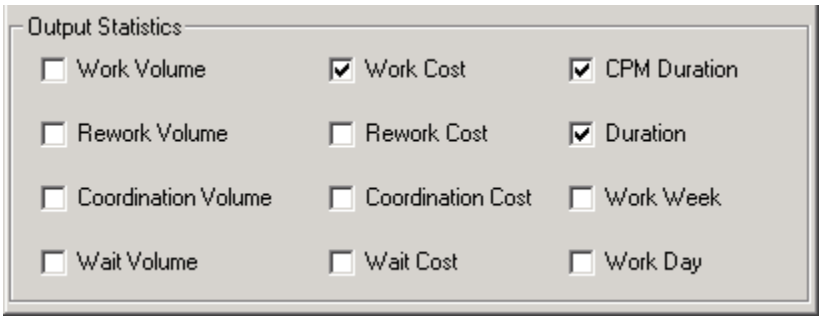


Figure 4.3 Output Statistics

The VitéProject parameters are adjusted in Area 4. The three areas, as previously discussed, are Efficiency, Requirements Volatility, and Complexity. The labels for the combo boxes are hyperlinks to the third display. The analyst can manipulate the three parameters from low, medium, and high. When an Automatic simulation is selected, other selection controls are enabled below the Complexity control; this allows the analyst to pick the "high" complexity multiple. This multiple can influence either bound that stipulates a "high" complexity. Entering a value for the multiple when a "high" complexity is not selected has no effect.

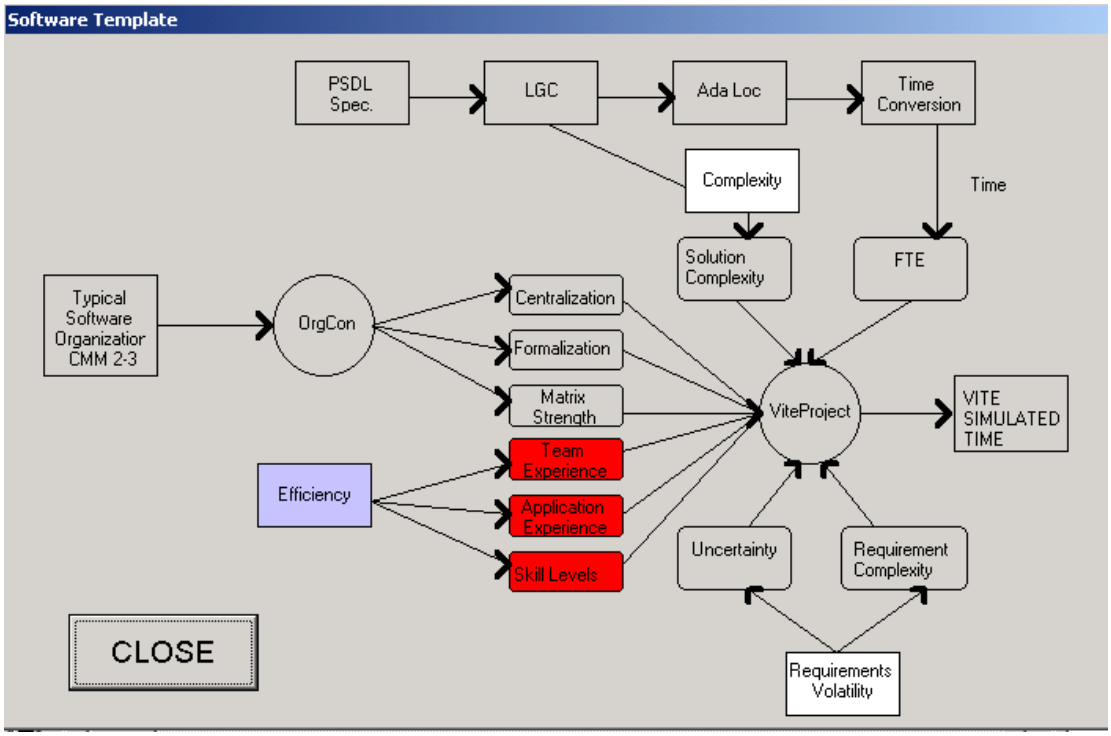


Figure 4.4 Informational Display

The name of the output Excel file is displayed in Area 5, minus the path. The output file is named for the day and time of its creation, thus making each new file name unique and distinguishable.

Figure 4.4 is the third display in the research program and is titled Informational Display. As shown, the display shows the mapping of the research parameter EFFICENCY to its actual VitéProject parameters. This display is solely used to illustrate the mapping from the research parameters to the actual VitéProject parameters. The only access to this display is via hyperlinks in the

main display; see Area 4 of Figure 4.2. The graphics of the informational display, other than the CLOSE control, are non-interactive.

C. EXTENSION VALIDATION

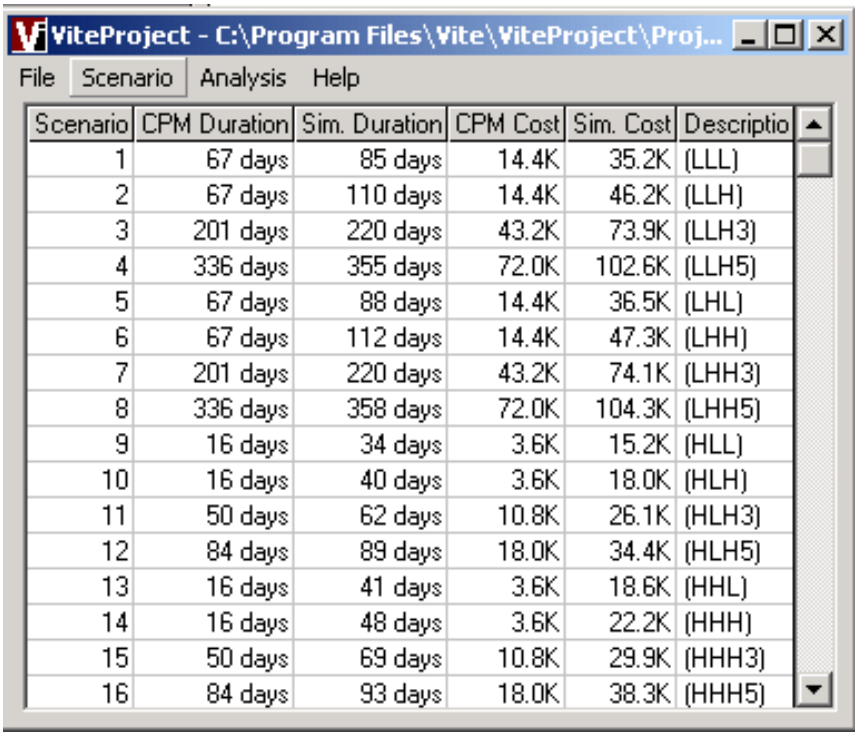
To validate the thesis, 100 trials were simulated for 16 scenarios. The scenarios were chosen to somewhat mirror the scenarios of (Nogueira, 2000). Given that the VitéProject model is a stochastic process whereby each particular simulation has some randomness involved, the values per simulation will vary by a small amount. Table 2.1 lists the 16 scenarios and the corresponding output values using the extension produced by this thesis. Figure 2.1 displays the output from VitéProject. Three individual simulation batches using each method were implemented. The following displays are each a representative average.

	CPM Duration	Duration	Simulation Cost
LLL	67	85	14400
LLH	67	111	14400
LLH3	202	218	43200
LLH5	336	356	72000
LHL	67	89	14400
LHH	67	113	14400
LHH3	202	222	43200
LHH5	336	358	72000
HLL	17	34	3600
HLH	17	40	3600
HLH 3	50	63	10800
HLH 5	84	89	18000
HHL	17	42	3600
HHH	17	48	3600
HHH3	50	68	10800
HHH5	84	93	18000

Table 4.1 Research 16 scenario output

As previously mentioned, some VitéProject parameters were set to mirror the work done in (Nogueira, 2000) to ensure research integrity for particular scenarios. Particularly, scenario properties that were customized for the validation were: Work Day (hrs) – 6, Centralization – "Low", Formalization – "Low", Matrix Strength – "High", Functional Error Rate - 0.01, and Information Exchange – 0.8. Additionally, each Dependency is initialized with strength of 10. The significance of the aforementioned scenario properties is negligible for this thesis. For a more detailed understanding of these parameters and how they pertain to modeling and simulating the software process refer to (Nogueira, 2000). Of note, the scenario property Project Error Rate was set at 0.1 for the thesis validation simulations whereas in (Nogueira, 2000) it is set to 0.01.[1]

[1] This change was made after consultation with Major Michael Murrah, who's Dissertation shall use this research to further the work of Dr. Nogueira de León



Scenario	CPM Duration	Sim. Duration	CPM Cost	Sim. Cost	Descriptio
1	67 days	85 days	14.4K	35.2K	(LLL)
2	67 days	110 days	14.4K	46.2K	(LLH)
3	201 days	220 days	43.2K	73.9K	(LLH3)
4	336 days	355 days	72.0K	102.6K	(LLH5)
5	67 days	88 days	14.4K	36.5K	(LHL)
6	67 days	112 days	14.4K	47.3K	(LHH)
7	201 days	220 days	43.2K	74.1K	(LHH3)
8	336 days	358 days	72.0K	104.3K	(LHH5)
9	16 days	34 days	3.6K	15.2K	(HLL)
10	16 days	40 days	3.6K	18.0K	(HLH)
11	50 days	62 days	10.8K	26.1K	(HLH3)
12	84 days	89 days	18.0K	34.4K	(HLH5)
13	16 days	41 days	3.6K	18.6K	(HHL)
14	16 days	48 days	3.6K	22.2K	(HHH)
15	50 days	69 days	10.8K	29.9K	(HHH3)
16	84 days	93 days	18.0K	38.3K	(HHH5)

Figure 4.5 VitéProject 16 Scenario output

D. ADDITIONAL REQUIREMENTS

Due to time constraints and a high requirements volatility, some aspects of the research lend themselves to improvement. There are also some legacy system improvements that might prove useful as the research progresses. The most needed improvement, for future research, is better access to the VitéProject database file. In the current version of the research program, the VitéProject database is accessed via data access objects (DAO). The DAO method *recordset.seek* will search the recordset (table) for a value based on a key. This method of search is theoretically faster than searching the entire table, however repeated attempts at its implementation have failed. Understanding and successfully implementing the

seek method will speed up database searches and reduce some of the program's overhead.

Another overhead reduction can occur if the graphic generated by VitéProject, for each simulation run, could be suppressed or inactivated. Also, an initial data file where the working files and directories can be hard coded could allow for suppression of the introductory display.

This research program makes updates to the database on all similar objects. For instance, each task in the software development lifecycle is represented as an *Activity* shape. Changes in the parameters of any *Activity* will make the same change to all activities on the Visio drawing. Workers are represented as *Actor* shapes, and changes to any one *Actor,* as with the *Activity,* will occur for all actors implemented via this research.

Follow-on upgrades to this research should look at providing the ability to group many Activities or many Actors such that changes can be made for groups of shapes without necessitating an all-encompassing parameter change. This will allow the analyst to better refine assumptions throughout the organizational structure.

V. CONCLUSION

This thesis developed an extension to VitéProject, a proprietary software product, to expand its functionality. The expanded functionality enables use of VitéProject for sensitivity analysis and allows validation of the VDT-2 model, the core model of VitéProject. Research analysts can now implement this thesis when estimating software engineering timelines.

The automation of parameter manipulation and multiple scenario simulations has been tailored to the work done by Dr. Nogueira. However, this thesis is easily adaptable to additional organizational structures.

Future work on the software risk assessment model that uses this research might find that the extensions created for VitéProject are not sufficient. Then it will be time for the product of this thesis to be extended. To aid in the development of additional extensions, the code and the interfaces are outlined within this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A

1. INITIAL DISPLAY

```
VERSION 5.00
Object = "{F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.2#0";
"COMDLG32.OCX"
Begin VB.Form Form2
   BorderStyle     =   1  'Fixed Single
   Caption         =   "Software Risk Assessment "
   ClientHeight    =   7365
   ClientLeft      =   780
   ClientTop       =   330
   ClientWidth     =   7125
   Icon            =   "Form2.frx":0000
   LinkTopic       =   "Form2"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   7365
   ScaleWidth      =   7125
   Begin VB.Frame Frame2
      Height          =   1695
      Left            =   0
      TabIndex        =   10
      Top             =   3900
      Width           =   7095
      Begin VB.CommandButton cmdContinue2
         Caption         =   "CONTINUE"
         BeginProperty Font
            Name            =   "MS Sans Serif"
            Size            =   12
            Charset         =   0
            Weight          =   400
            Underline       =   0   'False
            Italic          =   0   'False
            Strikethrough   =   0   'False
         EndProperty
         Height          =   495
         Left            =   600
         TabIndex        =   12
         Top             =   1080
         Width           =   2295
      End
      Begin VB.CommandButton cmdCancel2
         Caption         =   "CANCEL"
```

```
         BeginProperty Font
            Name            =   "MS Sans Serif"
            Size            =   12
            Charset         =   0
            Weight          =   400
            Underline       =   0   'False
            Italic          =   0   'False
            Strikethrough   =   0   'False
         EndProperty
         Height          =   495
         Left            =   3600
         TabIndex        =   11
         Top             =   1080
         Width           =   2295
      End
      Begin VB.Label Label2
         Alignment       =   2  'Center
         Caption         =   "Press CONTINUE to select the
Vite file that contains the Database, or press CANCEL to
exit program"
         BeginProperty Font
            Name            =   "MS Sans Serif"
            Size            =   12
            Charset         =   0
            Weight          =   400
            Underline       =   0   'False
            Italic          =   0   'False
            Strikethrough   =   0   'False
         EndProperty
         Height          =   735
         Left            =   120
         TabIndex        =   13
         Top             =   240
         Width           =   6855
      End
   End
   Begin VB.Frame Frame3
      Height          =   1695
      Left            =   0
      TabIndex        =   5
      Top             =   5640
      Width           =   7095
      Begin VB.CommandButton cmdContinue3
         Caption         =   "CONTINUE"
         BeginProperty Font
            Name            =   "MS Sans Serif"
```

```
         Size            =   12
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   495
      Left            =   2880
      TabIndex        =   8
      Top             =   960
      Width           =   1815
   End
   Begin VB.CommandButton cmdCancel3
      Cancel          =   -1  'True
      Caption         =   "CANCEL"
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   12
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   495
      Left            =   4920
      TabIndex        =   7
      Top             =   960
      Width           =   1815
   End
   Begin VB.DirListBox Dir1
      Height          =   765
      Left            =   240
      TabIndex        =   6
      Top             =   840
      Width           =   2415
   End
   Begin VB.Label Label3
      Alignment       =   2  'Center
      Caption         =   "Press CONTINUE to use
selected output file Directory, or CANCEL to exit program"
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   12
         Charset         =   0
```

```
            Weight          =   400
            Underline       =   0   'False
            Italic          =   0   'False
            Strikethrough   =   0   'False
         EndProperty
         Height          =   600
         Left            =   120
         TabIndex        =   9
         Top             =   240
         Width           =   6615
      End
   End
   Begin MSComDlg.CommonDialog dlgOpen
      Left            =   6480
      Top             =   1680
      _ExtentX        =   847
      _ExtentY        =   847
      _Version        =   393216
      DialogTitle     =   "Specify Vite Source/ Destination
File"
      Filter          =   "Vite Database file (*.vnb) |
*.vnb"
      InitDir         =   "c:\"
   End
   Begin VB.Frame Frame1
      Height          =   1695
      Left            =   0
      TabIndex        =   1
      Top             =   2160
      Width           =   7095
      Begin VB.CommandButton cmdCancel1
         Caption         =   "CANCEL"
         BeginProperty Font
            Name            =   "MS Sans Serif"
            Size            =   12
            Charset         =   0
            Weight          =   400
            Underline       =   0   'False
            Italic          =   0   'False
            Strikethrough   =   0   'False
         EndProperty
         Height          =   495
         Left            =   3600
         TabIndex        =   4
         Top             =   1080
         Width           =   2295
```

```
      End
      Begin VB.CommandButton cmdContinue1
         Caption         =   "CONTINUE"
         BeginProperty Font
            Name            =   "MS Sans Serif"
            Size            =   12
            Charset         =   0
            Weight          =   400
            Underline       =   0   'False
            Italic          =   0   'False
            Strikethrough   =   0   'False
         EndProperty
         Height          =   495
         Left            =   600
         TabIndex        =   3
         Top             =   1080
         Width           =   2295
      End
      Begin VB.Label Label1
         Alignment       =   2  'Center
         Caption         =   "Press CONTINUE to select the
Visio file that contains the Organizations format, or press
CANCEL to exit program"
         BeginProperty Font
            Name            =   "MS Sans Serif"
            Size            =   12
            Charset         =   0
            Weight          =   400
            Underline       =   0   'False
            Italic          =   0   'False
            Strikethrough   =   0   'False
         EndProperty
         Height          =   855
         Left            =   120
         TabIndex        =   2
         Top             =   240
         Width           =   6855
      End
   End
   Begin VB.Label lblIntro
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   9.75
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
```

```
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   2235
      Left            =   0
      TabIndex        =   0
      Top             =   0
      Width           =   7125
   End
End
Attribute VB_Name = "Form2"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
```

2. MAIN DISPLAY

```
VERSION 5.00
Object = "{F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.2#0";
"COMDLG32.OCX"
Begin VB.Form Form1
   BorderStyle     =   1  'Fixed Single
   Caption         =   "Thesis User Form"
   ClientHeight    =   4260
   ClientLeft      =   780
   ClientTop       =   330
   ClientWidth     =   6030
   ControlBox      =   0   'False
   Icon            =   "ThesisWk.frx":0000
   LinkTopic       =   "Form1"
   MinButton       =   0   'False
   ScaleHeight     =   4260
   ScaleWidth      =   6030
   Begin VB.Frame frameStats
      Caption         =   "Output Statistics"
      Enabled         =   0   'False
      Height          =   2175
      Left            =   0
      TabIndex        =   12
      Top             =   4300
      Width           =   6015
      Begin VB.CheckBox CkBxWeek
         Caption         =   "Work Week"
         Enabled         =   0   'False
         Height          =   375
         Index           =   11
```

```
   Left            =   4200
   TabIndex        =   24
   Top             =   1200
   Width           =   1215
End
Begin VB.CheckBox CkBxWaitC
   Caption         =   "Wait Cost"
   Enabled         =   0   'False
   Height          =   375
   Index           =   10
   Left            =   2400
   TabIndex        =   23
   Top             =   1680
   Width           =   1095
End
Begin VB.CheckBox CkBxCordC
   Caption         =   "Coordination Cost"
   Enabled         =   0   'False
   Height          =   375
   Index           =   9
   Left            =   2400
   TabIndex        =   22
   Top             =   1200
   Width           =   1815
End
Begin VB.CheckBox CkBxReC
   Caption         =   "Rework Cost"
   Enabled         =   0   'False
   Height          =   375
   Index           =   8
   Left            =   2400
   TabIndex        =   21
   Top             =   720
   Width           =   1335
End
Begin VB.CheckBox CkBxWorkC
   Caption         =   "Work Cost"
   Enabled         =   0   'False
   Height          =   375
   Index           =   7
   Left            =   2400
   TabIndex        =   20
   Top             =   240
   Value           =   1  'Checked
   Width           =   1215
End
```

```
Begin VB.CheckBox CkBxDuration
   Caption         =   "Duration"
   Enabled         =   0   'False
   Height          =   375
   Index           =   6
   Left            =   4200
   TabIndex        =   19
   Top             =   720
   Value           =   1  'Checked
   Width           =   975
End
Begin VB.CheckBox CkBxDay
   Caption         =   "Work Day"
   Enabled         =   0   'False
   Height          =   375
   Index           =   5
   Left            =   4200
   TabIndex        =   18
   Top             =   1680
   Width           =   1095
End
Begin VB.CheckBox CkBxWaitV
   Caption         =   "Wait Volume"
   Enabled         =   0   'False
   Height          =   375
   Index           =   4
   Left            =   240
   TabIndex        =   17
   Top             =   1680
   Width           =   1815
End
Begin VB.CheckBox CkBxCordV
   Caption         =   "Coordination Volume"
   Enabled         =   0   'False
   Height          =   375
   Index           =   3
   Left            =   240
   TabIndex        =   16
   Top             =   1200
   Width           =   1815
End
Begin VB.CheckBox CkBxReV
   Caption         =   "Rework Volume"
   Enabled         =   0   'False
   Height          =   375
   Index           =   2
```

```
      Left            =   240
      TabIndex        =   15
      Top             =   720
      Width           =   1815
   End
   Begin VB.CheckBox CkBxWorkV
      Caption         =   "Work Volume"
      Enabled         =   0   'False
      Height          =   375
      Index           =   1
      Left            =   240
      TabIndex        =   14
      Top             =   240
      Width           =   1815
   End
   Begin VB.CheckBox CkBxCPM
      Caption         =   "CPM Duration"
      Enabled         =   0   'False
      Height          =   375
      Index           =   0
      Left            =   4200
      TabIndex        =   13
      Top             =   240
      Value           =   1  'Checked
      Width           =   1335
   End
End
Begin VB.Frame Frame1
   Height          =   4215
   Left            =   0
   TabIndex        =   1
   Top             =   0
   Width           =   6015
   Begin VB.ComboBox EffCtl
      Enabled         =   0   'False
      Height          =   315
      ItemData        =   "ThesisWk.frx":0442
      Left            =   4200
      List            =   "ThesisWk.frx":0444
      Style           =   2  'Dropdown List
      TabIndex        =   33
      Top             =   1200
      Width           =   1335
   End
   Begin VB.CommandButton Helper
      Caption         =   "?"
```

```
         BeginProperty Font
            Name            =   "MS Sans Serif"
            Size            =   12
            Charset         =   0
            Weight          =   700
            Underline       =   0   'False
            Italic          =   0   'False
            Strikethrough   =   0   'False
         EndProperty
         Height          =   375
         Left            =   4920
         TabIndex        =   28
         Top             =   360
         Width           =   615
      End
      Begin VB.CommandButton RunSimulation
         Caption         =   "Run Simulation"
         Enabled         =   0   'False
         Height          =   495
         Left            =   240
         TabIndex        =   27
         Top             =   1080
         Width           =   1455
      End
      Begin MSComDlg.CommonDialog dlgOpen
         Left            =   5160
         Top             =   2640
         _ExtentX        =   847
         _ExtentY        =   847
         _Version        =   393216
         DialogTitle     =   "Specify Vite Source/
Destination File"
         Filter          =   "Vite Database file (*.vnb) |
*.vnb"
         InitDir         =   "c:\"
      End
      Begin VB.CommandButton updateDB
         Caption         =   "Update Database"
         BeginProperty Font
            Name            =   "Arial"
            Size            =   12
            Charset         =   0
            Weight          =   700
            Underline       =   0   'False
            Italic          =   0   'False
            Strikethrough   =   0   'False
```

```
   EndProperty
   Height          =   615
   Left            =   240
   TabIndex        =   26
   Top             =   2280
   Visible         =   0   'False
   Width           =   1455
End
Begin VB.ComboBox ReqCtl
   Enabled         =   0   'False
   Height          =   315
   ItemData        =   "ThesisWk.frx":0446
   Left            =   4200
   List            =   "ThesisWk.frx":0448
   Style           =   2  'Dropdown List
   TabIndex        =   11
   Top             =   1650
   Width           =   1335
End
Begin VB.TextBox TrialsCtl
   Alignment       =   2  'Center
   Height          =   345
   Left            =   1680
   TabIndex        =   10
   Text            =   "5"
   Top             =   480
   Width           =   615
End
Begin VB.TextBox stepsCtl
   Alignment       =   2  'Center
   Height          =   345
   Left            =   4200
   TabIndex        =   9
   Text            =   "1"
   Top             =   3195
   Visible         =   0   'False
   Width           =   615
End
Begin VB.TextBox MultCtl
   Alignment       =   2  'Center
   Height          =   345
   Left            =   4200
   TabIndex        =   8
   Text            =   "1"
   Top             =   2595
   Visible         =   0   'False
```

```
         Width           =   615
      End
      Begin VB.ComboBox CompCtl
         Enabled         =   0   'False
         Height          =   315
         Left            =   4200
         Style           =   2  'Dropdown List
         TabIndex        =   7
         Top             =   2160
         Width           =   1335
      End
      Begin VB.CommandButton ExitCmd
         Caption         =   "EXIT"
         Height          =   495
         Left            =   240
         TabIndex        =   0
         Top             =   3000
         Width           =   1455
      End
      Begin VB.CommandButton Tester
         Caption         =   "Change Selected Output
Statistics"
         Height          =   495
         Left            =   240
         TabIndex        =   5
         Top             =   1680
         Width           =   1455
      End
      Begin VB.OptionButton ManSim
         Caption         =   "Manually Simulate"
         Height          =   255
         Left            =   3120
         TabIndex        =   3
         Top             =   720
         Width           =   1575
      End
      Begin VB.OptionButton AutoSim
         Caption         =   "Auto Simulate"
         Height          =   255
         Left            =   3120
         TabIndex        =   2
         Top             =   360
         Width           =   1335
      End
      Begin VB.Label lblEff
         Caption         =   "Efficiency"
```

```
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   400
      Underline       =   -1  'True
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H00FF0000&
   Height          =   315
   Left            =   2280
   TabIndex        =   34
   Top             =   1200
   Width           =   855
End
Begin VB.Label lblComp
   Caption         =   "Complexity"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   400
      Underline       =   -1  'True
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H00FF0000&
   Height          =   315
   Left            =   2280
   TabIndex        =   31
   Top             =   2160
   Width           =   975
End
Begin VB.Label lblSteps
   Caption         =   "Steps (integer)"
   Height          =   255
   Left            =   2280
   TabIndex        =   30
   Top             =   3240
   Visible         =   0   'False
   Width           =   1095
End
Begin VB.Label lblReqVol
   Caption         =   "Requirements Volatility"
   BeginProperty Font
```

```
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   400
      Underline       =   -1  'True
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H00FF0000&
   Height          =   495
   Left            =   2280
   TabIndex        =   29
   Top             =   1560
   Width           =   1095
End
Begin VB.Label lblFileSaved
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   285
   Left            =   240
   TabIndex        =   25
   Top             =   3720
   Width           =   5295
End
Begin VB.Label Label1
   Alignment       =   2  'Center
   Caption         =   "Number of Trials to Run"
   Height          =   495
   Left            =   240
   TabIndex        =   4
   Top             =   360
   Width           =   1095
   WordWrap        =   -1  'True
End
Begin VB.Label lblMult
   Caption         =   "High Complexity Multiple"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
```

```
            Charset         =   0
            Weight          =   400
            Underline       =   -1  'True
            Italic          =   0   'False
            Strikethrough   =   0   'False
         EndProperty
         ForeColor       =   &H00FF0000&
         Height          =   495
         Left            =   2280
         TabIndex        =   32
         Top             =   2520
         Visible         =   0   'False
         Width           =   1215
      End
      Begin VB.Label lblMultMax
         Caption         =   "High Complexity Multiple:
Max"
         BeginProperty Font
            Name            =   "MS Sans Serif"
            Size            =   8.25
            Charset         =   0
            Weight          =   400
            Underline       =   -1  'True
            Italic          =   0   'False
            Strikethrough   =   0   'False
         EndProperty
         ForeColor       =   &H00FF0000&
         Height          =   405
         Left            =   2280
         TabIndex        =   6
         Top             =   2520
         Visible         =   0   'False
         Width           =   1335
      End
   End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
```

3. INFORMATIONAL MAPPING DISPLAY

```
VERSION 5.00
Begin VB.Form templ
   BorderStyle     =   1  'Fixed Single
   Caption         =   "Software Template"
   ClientHeight    =   6585
   ClientLeft      =   45
   ClientTop       =   330
   ClientWidth     =   10515
   ControlBox      =   0   'False
   Icon            =   "Templ.frx":0000
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   6585
   ScaleWidth      =   10515
   Begin VB.CommandButton Command1
      Caption         =   "CLOSE"
      BeginProperty Font
         Name            =   "Arial"
         Size            =   14.25
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   735
      Left            =   600
      TabIndex        =   22
      Top             =   5520
      Width           =   1815
   End
   Begin VB.Label Label5
      Alignment       =   2  'Center
      BackStyle       =   0  'Transparent
      Caption         =   "Team Experience"
      Height          =   375
      Left            =   4845
      TabIndex        =   16
      Top             =   3855
      Width           =   975
   End
   Begin VB.Label Label7
      BackStyle       =   0  'Transparent
      Caption         =   "Skill Levels"
```

```
   Height          =   255
   Left            =   4845
   TabIndex        =   14
   Top             =   5040
   Width           =   975
End
Begin VB.Label Label6
   Alignment       =   2  'Center
   BackStyle       =   0  'Transparent
   Caption         =   "Application Experience"
   Height          =   375
   Left            =   4845
   TabIndex        =   15
   Top             =   4395
   Width           =   975
End
Begin VB.Label Label8
   Alignment       =   2  'Center
   BackStyle       =   0  'Transparent
   Caption         =   "Uncertainty"
   Height          =   255
   Left            =   6525
   TabIndex        =   13
   Top             =   4800
   Width           =   855
End
Begin VB.Label Label9
   BackStyle       =   0  'Transparent
   Caption         =   "Requirement Complexity"
   Height          =   390
   Left            =   8085
   TabIndex        =   12
   Top             =   4800
   Width           =   975
End
Begin VB.Label Label11
   Alignment       =   2  'Center
   BackStyle       =   0  'Transparent
   Caption         =   "FTE"
   Height          =   375
   Left            =   8205
   TabIndex        =   10
   Top             =   2040
   Width           =   855
End
Begin VB.Label Label10
```

```
   BackStyle       =   0  'Transparent
   Caption         =   "Solution Complexity"
   Height          =   390
   Left            =   6480
   TabIndex        =   11
   Top             =   2010
   Width           =   855
End
Begin VB.Line Line30
   BorderWidth     =   4
   Index           =   17
   X1              =   8880
   X2              =   8760
   Y1              =   1920
   Y2              =   1800
End
Begin VB.Line Line29
   BorderWidth     =   4
   Index           =   17
   X1              =   8880
   X2              =   9000
   Y1              =   1920
   Y2              =   1800
End
Begin VB.Line Line30
   BorderWidth     =   4
   Index           =   16
   X1              =   7680
   X2              =   7680
   Y1              =   3120
   Y2              =   3000
End
Begin VB.Line Line29
   BorderWidth     =   4
   Index           =   16
   X1              =   7680
   X2              =   7800
   Y1              =   3120
   Y2              =   3120
End
Begin VB.Line Line30
   BorderWidth     =   4
   Index           =   15
   X1              =   7800
   X2              =   7800
   Y1              =   4200
```

```
   Y2              =   4080
End
Begin VB.Line Line29
   BorderWidth     =   4
   Index           =   15
   X1              =   7440
   X2              =   7560
   Y1              =   4080
   Y2              =   4080
End
Begin VB.Line Line30
   BorderWidth     =   4
   Index           =   14
   X1              =   7560
   X2              =   7560
   Y1              =   4200
   Y2              =   4080
End
Begin VB.Line Line29
   BorderWidth     =   4
   Index           =   14
   X1              =   7800
   X2              =   7920
   Y1              =   4080
   Y2              =   4080
End
Begin VB.Line Line30
   BorderWidth     =   4
   Index           =   13
   X1              =   8760
   X2              =   8760
   Y1              =   5400
   Y2              =   5280
End
Begin VB.Line Line29
   BorderWidth     =   4
   Index           =   13
   X1              =   8640
   X2              =   8760
   Y1              =   5280
   Y2              =   5280
End
Begin VB.Line Line30
   BorderWidth     =   4
   Index           =   12
   X1              =   8880
```

```
   X2              =   9000
   Y1              =   3720
   Y2              =   3600
End
Begin VB.Line Line29
   BorderWidth     =   4
   Index           =   12
   X1              =   8880
   X2              =   9000
   Y1              =   3480
   Y2              =   3600
End
Begin VB.Line Line30
   BorderWidth     =   4
   Index           =   11
   X1              =   7080
   X2              =   7080
   Y1              =   5400
   Y2              =   5280
End
Begin VB.Line Line29
   BorderWidth     =   4
   Index           =   11
   X1              =   7080
   X2              =   7200
   Y1              =   5280
   Y2              =   5280
End
Begin VB.Line Line30
   BorderWidth     =   4
   Index           =   10
   X1              =   7560
   X2              =   7560
   Y1              =   3120
   Y2              =   3000
End
Begin VB.Line Line29
   BorderWidth     =   4
   Index           =   10
   X1              =   7440
   X2              =   7560
   Y1              =   3120
   Y2              =   3120
End
Begin VB.Line Line30
   BorderWidth     =   4
```

```
   Index           =   9
   X1              =   7080
   X2              =   7200
   Y1              =   1920
   Y2              =   1800
End
Begin VB.Line Line29
   BorderWidth     =   4
   Index           =   9
   X1              =   6960
   X2              =   7080
   Y1              =   1800
   Y2              =   1920
End
Begin VB.Line Line30
   BorderWidth     =   4
   Index           =   8
   X1              =   8040
   X2              =   8160
   Y1              =   720
   Y2              =   600
End
Begin VB.Line Line29
   BorderWidth     =   4
   Index           =   8
   X1              =   8040
   X2              =   8160
   Y1              =   480
   Y2              =   600
End
Begin VB.Line Line30
   BorderWidth     =   4
   Index           =   7
   X1              =   4680
   X2              =   4800
   Y1              =   5160
   Y2              =   5040
End
Begin VB.Line Line29
   BorderWidth     =   4
   Index           =   7
   X1              =   4680
   X2              =   4800
   Y1              =   4920
   Y2              =   5040
End
```

```
Begin VB.Line Line30
   BorderWidth     =   4
   Index           =   6
   X1              =   4680
   X2              =   4800
   Y1              =   4680
   Y2              =   4560
End
Begin VB.Line Line29
   BorderWidth     =   4
   Index           =   6
   X1              =   4680
   X2              =   4800
   Y1              =   4440
   Y2              =   4560
End
Begin VB.Line Line30
   BorderWidth     =   4
   Index           =   5
   X1              =   4680
   X2              =   4800
   Y1              =   4200
   Y2              =   4080
End
Begin VB.Line Line29
   BorderWidth     =   4
   Index           =   5
   X1              =   4680
   X2              =   4800
   Y1              =   3960
   Y2              =   4080
End
Begin VB.Line Line30
   BorderWidth     =   4
   Index           =   4
   X1              =   4680
   X2              =   4800
   Y1              =   3600
   Y2              =   3480
End
Begin VB.Line Line29
   BorderWidth     =   4
   Index           =   4
   X1              =   4680
   X2              =   4800
   Y1              =   3360
```

```
   Y2              =   3480
End
Begin VB.Line Line30
   BorderWidth     =   4
   Index           =   3
   X1              =   4680
   X2              =   4800
   Y1              =   3120
   Y2              =   3000
End
Begin VB.Line Line29
   BorderWidth     =   4
   Index           =   3
   X1              =   4680
   X2              =   4800
   Y1              =   2880
   Y2              =   3000
End
Begin VB.Line Line30
   BorderWidth     =   4
   Index           =   2
   X1              =   4680
   X2              =   4800
   Y1              =   2520
   Y2              =   2400
End
Begin VB.Line Line29
   BorderWidth     =   4
   Index           =   2
   X1              =   4680
   X2              =   4800
   Y1              =   2280
   Y2              =   2400
End
Begin VB.Line Line30
   BorderWidth     =   4
   Index           =   1
   X1              =   2400
   X2              =   2520
   Y1              =   2880
   Y2              =   2760
End
Begin VB.Line Line29
   BorderWidth     =   4
   Index           =   1
   X1              =   2400
```

```
   X2              =   2520
   Y1              =   2640
   Y2              =   2760
End
Begin VB.Line Line32
   BorderWidth     =   4
   X1              =   6240
   X2              =   6360
   Y1              =   705
   Y2              =   585
End
Begin VB.Line Line31
   BorderWidth     =   4
   X1              =   6240
   X2              =   6360
   Y1              =   465
   Y2              =   585
End
Begin VB.Line Line28
   BorderWidth     =   4
   X1              =   4440
   X2              =   4560
   Y1              =   720
   Y2              =   600
End
Begin VB.Line Line27
   BorderWidth     =   4
   X1              =   4440
   X2              =   4560
   Y1              =   480
   Y2              =   600
End
Begin VB.Line Line30
   BorderWidth     =   4
   Index           =   0
   X1              =   7080
   X2              =   7200
   Y1              =   3720
   Y2              =   3600
End
Begin VB.Line Line29
   BorderWidth     =   4
   Index           =   0
   X1              =   7080
   X2              =   7200
   Y1              =   3480
```

```
   Y2              =   3600
End
Begin VB.Label Label14
   Alignment       =   2  'Center
   BackStyle       =   0  'Transparent
   Caption         =   "Time"
   Height          =   255
   Left            =   9000
   TabIndex        =   21
   Top             =   1320
   Width           =   735
End
Begin VB.Line Line26
   X1              =   8880
   X2              =   8880
   Y1              =   840
   Y2              =   1920
End
Begin VB.Line Line25
   X1              =   7080
   X2              =   7080
   Y1              =   1680
   Y2              =   1920
End
Begin VB.Line Line24
   X1              =   5160
   X2              =   6480
   Y1              =   840
   Y2              =   1440
End
Begin VB.Label Label1
   Alignment       =   2  'Center
   BackStyle       =   0  'Transparent
   Caption         =   "Complexity"
   Height          =   255
   Index           =   8
   Left            =   6600
   TabIndex        =   20
   Top             =   1280
   Width           =   855
End
Begin VB.Shape shpComp
   BackStyle       =   1  'Opaque
   FillColor       =   &H00FFC0C0&
   Height          =   615
   Index           =   20
```

```
   Left            =   6480
   Top             =   1080
   Width           =   1095
End
Begin VB.Line Line23
   X1              =   7440
   X2              =   8160
   Y1              =   600
   Y2              =   600
End
Begin VB.Line Line21
   X1              =   5640
   X2              =   6360
   Y1              =   585
   Y2              =   585
End
Begin VB.Line Line20
   X1              =   3840
   X2              =   4560
   Y1              =   585
   Y2              =   585
End
Begin VB.Line Line22
   X1              =   1440
   X2              =   2520
   Y1              =   2760
   Y2              =   2760
End
Begin VB.Line Line19
   X1              =   8760
   X2              =   7920
   Y1              =   5280
   Y2              =   5880
End
Begin VB.Line Line18
   X1              =   7920
   X2              =   7080
   Y1              =   5880
   Y2              =   5280
End
Begin VB.Line Line17
   X1              =   8280
   X2              =   7800
   Y1              =   4680
   Y2              =   4080
End
```

```
Begin VB.Line Line16
   X1              =   7080
   X2              =   7560
   Y1              =   4680
   Y2              =   4080
End
Begin VB.Line Line15
   X1              =   8160
   X2              =   9000
   Y1              =   3600
   Y2              =   3600
End
Begin VB.Line Line14
   X1              =   8640
   X2              =   7680
   Y1              =   2520
   Y2              =   3120
End
Begin VB.Line Line13
   X1              =   6960
   X2              =   7560
   Y1              =   2520
   Y2              =   3120
End
Begin VB.Line Line12
   X1              =   5880
   X2              =   7200
   Y1              =   5040
   Y2              =   3600
End
Begin VB.Line Line11
   X1              =   5880
   X2              =   7200
   Y1              =   4560
   Y2              =   3600
End
Begin VB.Line Line10
   X1              =   5880
   X2              =   7200
   Y1              =   3960
   Y2              =   3600
End
Begin VB.Line Line9
   X1              =   5880
   X2              =   7200
   Y1              =   3600
```

```
   Y2              =   3600
End
Begin VB.Line Line8
   X1              =   5880
   X2              =   7200
   Y1              =   3000
   Y2              =   3600
End
Begin VB.Line Line7
   X1              =   5880
   X2              =   7200
   Y1              =   2520
   Y2              =   3600
End
Begin VB.Line Line6
   X1              =   4800
   X2              =   3480
   Y1              =   3000
   Y2              =   2760
End
Begin VB.Line Line5
   X1              =   4800
   X2              =   3480
   Y1              =   3480
   Y2              =   2760
End
Begin VB.Line Line4
   X1              =   4800
   X2              =   3480
   Y1              =   2400
   Y2              =   2760
End
Begin VB.Line Line3
   X1              =   4800
   X2              =   3480
   Y1              =   4080
   Y2              =   4440
End
Begin VB.Line Line2
   X1              =   3480
   X2              =   4800
   Y1              =   4440
   Y2              =   4560
End
Begin VB.Line Line1
   X1              =   3480
```

```
   X2              =   4800
   Y1              =   4440
   Y2              =   5040
End
Begin VB.Shape shpFTE
   BackColor       =   &H000000FF&
   FillColor       =   &H0080FF80&
   Height          =   615
   Index           =   0
   Left            =   8085
   Shape           =   4  'Rounded Rectangle
   Top             =   1920
   Width           =   1095
End
Begin VB.Shape shpSolComp
   BackColor       =   &H000000FF&
   FillColor       =   &H0080FF80&
   Height          =   615
   Index           =   3
   Left            =   6405
   Shape           =   4  'Rounded Rectangle
   Top             =   1920
   Width           =   1095
End
Begin VB.Shape Shape1
   FillColor       =   &H0080FF80&
   Height          =   735
   Index           =   6
   Left            =   9045
   Top             =   3240
   Width           =   1095
End
Begin VB.Shape shpReqComp
   BackColor       =   &H000000FF&
   FillColor       =   &H0080FF80&
   Height          =   615
   Index           =   7
   Left            =   7965
   Shape           =   4  'Rounded Rectangle
   Top             =   4680
   Width           =   1215
End
Begin VB.Shape shpUnc
   BackColor       =   &H000000FF&
   FillColor       =   &H0080FF80&
   Height          =   615
```

```
   Index           =   8
   Left            =   6405
   Shape           =   4  'Rounded Rectangle
   Top             =   4680
   Width           =   1095
End
Begin VB.Shape Shape1
   FillColor       =   &H0080FF80&
   Height          =   975
   Index           =   9
   Left            =   6885
   Shape           =   3  'Circle
   Top             =   3120
   Width           =   1575
End
Begin VB.Shape shpSkills
   BackColor       =   &H000000FF&
   FillColor       =   &H0080FF80&
   Height          =   390
   Index           =   1
   Left            =   4800
   Shape           =   4  'Rounded Rectangle
   Top             =   4920
   Width           =   1095
End
Begin VB.Shape shpApp
   BackColor       =   &H000000FF&
   FillColor       =   &H0080FF80&
   Height          =   405
   Index           =   2
   Left            =   4800
   Shape           =   4  'Rounded Rectangle
   Top             =   4380
   Width           =   1095
End
Begin VB.Shape Shape1
   FillColor       =   &H0080FF80&
   Height          =   405
   Index           =   4
   Left            =   4800
   Shape           =   4  'Rounded Rectangle
   Top             =   3330
   Width           =   1095
End
Begin VB.Shape shpTeam
   BackColor       =   &H000000FF&
```

```
   FillColor       =   &H0080FF80&
   Height          =   405
   Index           =   5
   Left            =   4800
   Shape           =   4  'Rounded Rectangle
   Top             =   3840
   Width           =   1095
End
Begin VB.Shape Shape1
   FillColor       =   &H0080FF80&
   Height          =   390
   Index           =   10
   Left            =   4800
   Shape           =   4  'Rounded Rectangle
   Top             =   2805
   Width           =   1095
End
Begin VB.Shape Shape1
   FillColor       =   &H0080FF80&
   Height          =   390
   Index           =   11
   Left            =   4800
   Shape           =   4  'Rounded Rectangle
   Top             =   2280
   Width           =   1095
End
Begin VB.Label Label2
   Alignment       =   2  'Center
   BackStyle       =   0  'Transparent
   Caption         =   "Centralization"
   Height          =   255
   Index           =   0
   Left            =   4845
   TabIndex        =   19
   Top             =   2400
   Width           =   975
End
Begin VB.Label Label3
   BackStyle       =   0  'Transparent
   Caption         =   "Formalization"
   Height          =   255
   Left            =   4845
   TabIndex        =   18
   Top             =   2880
   Width           =   975
End
```

```
Begin VB.Label Label4
   Alignment       =   2  'Center
   BackStyle       =   0  'Transparent
   Caption         =   "Matrix Strength"
   Height          =   375
   Left            =   4845
   TabIndex        =   17
   Top             =   3345
   Width           =   975
End
Begin VB.Label Label12
   BackStyle       =   0  'Transparent
   Caption         =   "VITE SIMULATED TIME"
   Height          =   600
   Left            =   9135
   TabIndex        =   9
   Top             =   3360
   Width           =   975
End
Begin VB.Label Label13
   Alignment       =   2  'Center
   BackStyle       =   0  'Transparent
   Caption         =   "ViteProject"
   Height          =   375
   Left            =   7245
   TabIndex        =   8
   Top             =   3480
   Width           =   855
End
Begin VB.Label Label1
   Alignment       =   2  'Center
   BackStyle       =   0  'Transparent
   Caption         =   "Requirements Volatility"
   Height          =   375
   Index           =   2
   Left            =   7465
   TabIndex        =   7
   Top             =   5955
   Width           =   975
End
Begin VB.Shape shpReqV
   BackStyle       =   1  'Opaque
   FillColor       =   &H00FFC0C0&
   Height          =   615
   Index           =   19
   Left            =   7440
```

```
      Top             =   5880
      Width           =   1095
   End
   Begin VB.Label Label1
      Alignment       =   2  'Center
      BackStyle       =   0  'Transparent
      Caption         =   "Ada Loc"
      Height          =   255
      Index           =   7
      Left            =   6480
      TabIndex        =   6
      Top             =   420
      Width           =   855
   End
   Begin VB.Label Label1
      Alignment       =   2  'Center
      BackStyle       =   0  'Transparent
      Caption         =   "Time Conversion"
      Height          =   375
      Index           =   6
      Left            =   8280
      TabIndex        =   5
      Top             =   360
      Width           =   855
   End
   Begin VB.Label Label1
      Alignment       =   2  'Center
      BackStyle       =   0  'Transparent
      Caption         =   "Typical Software Organization
CMM 2-3"
      Height          =   840
      Index           =   5
      Left            =   480
      TabIndex        =   4
      Top             =   2400
      Width           =   855
   End
   Begin VB.Label Label1
      Alignment       =   2  'Center
      BackStyle       =   0  'Transparent
      Caption         =   "Efficiency"
      Height          =   255
      Index           =   4
      Left            =   2520
      TabIndex        =   3
      Top             =   4280
```

```
   Width           =   855
End
Begin VB.Label Label1
   Alignment       =   2  'Center
   BackStyle       =   0  'Transparent
   Caption         =   "OrgCon"
   Height          =   255
   Index           =   3
   Left            =   2640
   TabIndex        =   2
   Top             =   2640
   Width           =   855
End
Begin VB.Label Label1
   Alignment       =   2  'Center
   BackStyle       =   0  'Transparent
   Caption         =   "LGC"
   Height          =   255
   Index           =   1
   Left            =   4680
   TabIndex        =   1
   Top             =   420
   Width           =   855
End
Begin VB.Label Label1
   Alignment       =   2  'Center
   BackStyle       =   0  'Transparent
   Caption         =   "PSDL Spec."
   Height          =   375
   Index           =   0
   Left            =   2880
   TabIndex        =   0
   Top             =   360
   Width           =   855
End
Begin VB.Shape Shape1
   FillColor       =   &H0080FF80&
   Height          =   615
   Index           =   18
   Left            =   4560
   Top             =   240
   Width           =   1095
End
Begin VB.Shape Shape1
   FillColor       =   &H0080FF80&
   Height          =   615
```

```
   Index           =   17
   Left            =   6360
   Top             =   240
   Width           =   1095
End
Begin VB.Shape Shape1
   FillColor       =   &H0080FF80&
   Height          =   615
   Index           =   16
   Left            =   8160
   Top             =   240
   Width           =   1095
End
Begin VB.Shape Shape1
   FillColor       =   &H0080FF80&
   Height          =   975
   Index           =   15
   Left            =   360
   Top             =   2280
   Width           =   1095
End
Begin VB.Shape Shape1
   FillColor       =   &H0080FF80&
   Height          =   615
   Index           =   14
   Left            =   2760
   Top             =   240
   Width           =   1095
End
Begin VB.Shape Shape1
   FillColor       =   &H0080FF80&
   Height          =   975
   Index           =   13
   Left            =   2250
   Shape           =   3  'Circle
   Top             =   2280
   Width           =   1575
End
Begin VB.Shape shpEff
   BackStyle       =   1  'Opaque
   FillColor       =   &H00FFC0C0&
   Height          =   615
   Index           =   12
   Left            =   2400
   Top             =   4080
   Width           =   1095
```

```
   End
End
Attribute VB_Name = "templ"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
```

LIST OF REFERENCES

Deitel, H. M., Deitel, P.J., Nieto, T. R., *Visual Basic 6: How to Program*, Prentice Hall, Inc., 1999

Edson, David, *Professional Development with Visio 2000: The Authoritative Solution*, SAMS Publishing, 2000

Jin, Y., Levitt, R., "The Virtual Design Team: A Computational Model of Project Organizations," Paper to appear in *Computational and Mathematical Organization Theory*. 1996

Krumm, Rob, *Access 2000 Programming for Dummies*, IDG Books Worldwide, Inc., 1999

Nogueira, J.C., *A Formal Model for Risk Assessment in Software Projects,* Ph.D. Dissertation, Naval Postgraduate School, Monterey, California, September 2000.

Thomsen, J., and others, "A Trajectory for Validating Computational Emulation Models of Organizations," *Journal of Computational & Mathematical Organization Theory*, 5, (4), pp. 385-401, December 1999

Vité Incorporated, *ViteProject 2.0 Users Guide*, 1996-1998

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. LT Byron Alexander
   Naval Submarine School
   Groton, CT. 06349-5700

4. Ms. Reba Alexander
   712 E. 51st Street
   Chicago, Il. 60615