



2015-09

Planning to execution earned value risk management tool

Baldwin, Matthew

Monterey, California: Naval Postgraduate School

<http://hdl.handle.net/10945/47326>



Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

**Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943**

<http://www.nps.edu/library>



**NAVAL
POSTGRADUATE
SCHOOL**

MONTEREY, CALIFORNIA

**SYSTEMS ENGINEERING
CAPSTONE PROJECT REPORT**

**PLANNING TO EXECUTION EARNED VALUE RISK
MANAGEMENT TOOL**

by

Team MERICA
Cohort 311-141A

September 2015

Project Advisors:

Mark M. Rhoades
Gregory Miller

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE September 2015	3. REPORT TYPE AND DATES COVERED Capstone Project Report	
4. TITLE AND SUBTITLE PLANNING TO EXECUTION EARNED VALUE RISK MANAGEMENT TOOL			5. FUNDING NUMBERS	
6. AUTHOR(S) Cohort 311-141A/Team MERICA				
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 Department of Defense or the U.S. Government. IRB Protocol number NPS.2015.0034-IR-EM2-A.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) <p>This study presents a proof-of-concept software application capable of producing risk-based estimates for project costs and schedules, tracking project performance against baseline plans, and providing real-time updates for cost and schedule at completion. Current budget and schedule estimation practices within the Department of Defense (DOD) are often performed by different personnel across several software tools. Market research of existing commercial off-the-shelf (COTS) project planning and tracking software shows that tools commonly used by many industries require additional software packages and user training to perform uncertainty analyses. A modeling and simulation experiment demonstrates how using minimum and maximum duration estimates can result in baseline project plans that greatly underestimate and overestimate the project schedule, respectively.</p> <p>The prototype software application integrates the planning and execution tracking of projects. It accounts for uncertainty within the various activities in a project, thus creating more realistic baseline plans. During project execution, the prototype software application shortens the project status feedback cycle, identifies any activities that go over cost or schedule, and guides management action by determining which activities have the greatest impact on the outcome of the project. It is easier to use when compared to current planning and execution tracking software applications.</p>				
14. SUBJECT TERMS budget, schedule, cost, duration, uncertainty, estimates, earned-value management, risk-based scheduling, risk-based budgeting			15. NUMBER OF PAGES 143	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

PLANNING TO EXECUTION EARNED VALUE RISK MANAGEMENT TOOL

Cohort 311-141A/Team MERICA

LCDR Matthew Baldwin

Lisa Aschenbrenner

Devilin Moye

LT David Shultz

Brian Hirth

Emmanuel Navedo

Jason Ancheta

Joseph Martin

Andrew Twigg

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

AND

MASTER OF SCIENCE IN ENGINEERING SYSTEMS

from the

**NAVAL POSTGRADUATE SCHOOL
September 2015**

Lead editor: LCDR Matthew Baldwin

Reviewed by:

Mark M. Rhoades
Project Advisor

Gregory Miller
Project Advisor

Accepted by:

Ronald Giachetti, Ph.D.
Systems Engineering Department

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This study presents a proof-of-concept software application capable of producing risk-based estimates for project costs and schedules, tracking project performance against baseline plans, and providing real-time updates for cost and schedule at completion. Current budget and schedule estimation practices within the Department of Defense (DOD) are often performed by different personnel across several software tools. Market research of existing commercial off-the-shelf (COTS) project planning and tracking software shows that tools commonly used by many industries require additional software packages and user training to perform uncertainty analyses. A modeling and simulation experiment demonstrates how using minimum and maximum duration estimates can result in baseline project plans that greatly underestimate and overestimate the project schedule, respectively.

The prototype software application integrates the planning and execution tracking of projects. It accounts for uncertainty within the various activities in a project, thus creating more realistic baseline plans. During project execution, the prototype software application shortens the project status feedback cycle, identifies any activities that go over cost or schedule, and guides management action by determining which activities have the greatest impact on the outcome of the project. It is easier to use when compared to current planning and execution tracking software applications.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	PROBLEM STATEMENT	1
B.	GOALS AND OBJECTIVES.....	2
C.	SCOPE	3
D.	RESEARCH QUESTIONS.....	3
E.	ASSUMPTIONS.....	5
F.	SYSTEMS ENGINEERING PROCESS.....	5
	1. Concept Development.....	6
	2. Requirements Engineering.....	6
	3. System Architecture.....	7
	4. System Design and Development.....	7
	5. Verification Test.....	7
	6. Validation Test	8
	7. System Acceptance.....	8
G.	REPORT ROADMAP	8
II.	PROBLEM SPACE EXPLORATION	11
A.	BACKGROUND	11
	1. Cost Estimation Point Estimates	11
	2. Cost Estimation Uncertainty.....	12
	3. Existing Cost Estimation Tools.....	14
	4. Schedule Estimation Point Estimates and Critical Path Method..	15
	5. Schedule Estimation Uncertainty	16
	6. Existing Schedule Estimation Tools	17
	7. Integrated Cost and Schedule Estimation Practices.....	17
	<i>a. Concerns.....</i>	<i>18</i>
	<i>b. Existing Tools.....</i>	<i>18</i>
	8. Earned Value Management	19
	9. Earned Value Management Uncertainty Considerations	21
	10. Accuracy of Estimation Values.....	23
	<i>a. Design of Experiments.....</i>	<i>24</i>
	<i>b. Result Analysis</i>	<i>31</i>
B.	STAKEHOLDER ANALYSIS	33
C.	OPERATIONAL-BASED SCENARIO	37
	1. Project Planning Phase.....	39
	<i>a. Project Initialization Activity.....</i>	<i>40</i>
	<i>b. Project Network Design Activity.....</i>	<i>42</i>
	<i>c. Simulation and Analysis Activity.....</i>	<i>43</i>
	2. Project Management Phase.....	45
	<i>a. Performance Tracking Activity</i>	<i>48</i>
	<i>b. Project Management and Corrective Action Activity</i>	<i>50</i>
D.	CHAPTER SUMMARY.....	53

III.	REQUIREMENT ANALYSIS.....	55
	A. STAKEHOLDER INITIAL REQUIREMENTS.....	55
	B. STAKEHOLDER REQUIREMENT PRIORITIZATION.....	58
	C. REQUIREMENT ANALYSIS PROCESS	59
	D. CHAPTER SUMMARY.....	61
IV.	ARCHITECTURAL DEVELOPMENT.....	63
	A. ARCHITECTURAL MODEL DEVELOPMENT.....	63
	B. FUNCTIONAL ARCHITECTURE	64
	C. CHAPTER SUMMARY.....	69
V.	SOFTWARE DEVELOPMENT	71
	A. SOFTWARE DEVELOPMENT PROCESS.....	71
	B. OBJECT-ORIENTED PARADIGM	72
	C. DOCUMENTATION AND SOURCE CONTROL	73
	D. SYSTEM SPECIFICATION DEVELOPMENT	74
	E. PROTOTYPE DEVELOPMENT	75
	1. Initial Prototype	76
	2. Ideation Prototype	80
	3. Final Prototype.....	84
	F. PERFORMANCE AGAINST EXISTING TOOLS AND PROCESSES	89
	G. CHAPTER SUMMARY.....	90
VI.	SUMMARY AND CONCLUSION	93
	A. SUMMARY	93
	B. FUTURE WORK	98
	1. Final Prototype Development	98
	2. Future Vision.....	100
	C. CONCLUSION	100
	APPENDIX A. STAKEHOLDER RESEARCH QUESTIONS.....	103
	APPENDIX B. STAKEHOLDER REQUIREMENTS	109
	APPENDIX C. SOFTWARE SOURCE CODE.....	119
	LIST OF REFERENCES	121
	INITIAL DISTRIBUTION LIST	123

LIST OF FIGURES

Figure 1.	Capstone System Engineering Process (after MITRE 2014, 2).....	6
Figure 2.	Three Cost Estimating Methods Compared (from GAO 2009, 108).....	12
Figure 3.	Potential Sources of Program Cost Estimate Uncertainty (from GAO 2009, 160)	13
Figure 4.	Sample Cumulative Probability distribution or S-Curve for Cost Estimates (from GAO 2009, 157)	14
Figure 5.	Point Estimate Probability Distribution Driven by WBS Distributions (from GAO 2009, 168)	14
Figure 6.	An Activity Network (from GAO 2009, 219)	16
Figure 7.	Sample cumulative probability distribution or S-curve for schedule (from Pritchard 2015, 47).....	17
Figure 8.	DAU Earned Value Management Gold Card (from DAU 2015)	20
Figure 9.	Summary of steps to integrate EVM and RM (from Hillson 2004, 5)	22
Figure 10.	Risk-based cumulative spend profile (from Hillson 2004, 6).....	23
Figure 11.	Five Task Elements, Two Critical Paths and Low Interdependence	24
Figure 12.	Five Task Elements, Two Critical Paths and Medium Interdependence	25
Figure 13.	Five Task Elements, Four Critical Paths, and High Interdependence	25
Figure 14.	Boxplot of Normalized Results for each Test Case	32
Figure 15.	Prototype Software Application Operational View	38
Figure 16.	Operational-Based Scenario Decomposition	38
Figure 17.	Operational Sequence Diagram for Planning Phase	40
Figure 18.	Operational Sequence Diagram for Project Execution Phase.....	47
Figure 19.	Stakeholder Interview Notes on RealtimeBoard.....	56
Figure 20.	Portion of Project Execution Phase on RealtimeBoard	57
Figure 21.	IDEF0 Functional Building Block (from DAU 2001, 51).....	63
Figure 22.	PEEVRMT Functional Hierarchy.....	65
Figure 23.	PEEVRMT IDEF0 Diagram	66
Figure 24.	PEEVRMT IDEF0 Diagram Functional Upper Section Close Up.....	67
Figure 25.	PEEVRMT IDEF0 Diagram Functional Lower Section Close Up	68
Figure 26.	Prototype Software Application Model-View-Controller Architectural Pattern	73
Figure 27.	Initial Cost-Schedule Analysis Demonstration Prototype in MS Excel	77
Figure 28.	Cost-Schedule Forecast and Trajectory Demonstration Prototype.....	79
Figure 29.	Prototype Software Application – Project Description Tab	80
Figure 30.	Prototype Software Application – Project Network Tab	81
Figure 31.	Prototype Software Application – Project Planning Tab.....	82
Figure 32.	Prototype Software Application – Project Cost Trajectory Tab	83
Figure 33.	Prototype Software Application – Simulation Output Demonstration	84
Figure 34.	Software Application – Project Description	85
Figure 35.	Software Application – Project Network Design.....	86
Figure 36.	Software Application – Project Network Design Alternate View	86
Figure 37.	Software Application – Simulation Parameters.....	87

Figure 38.	Software Application – Project Planning.....	88
Figure 39.	Software Application – Project Cost Trajectory	89

LIST OF TABLES

Table 1.	DAU EVM Terms Definition (from DAU 2015)	20
Table 2.	Default Bounds for Subjective Distributions (from <i>Air Force Cost Risk and Uncertainty</i> 2007, 16)	26
Table 3.	JMP Pro Design of Experiment Run Definitions.....	26
Table 4.	Monte Carlo Simulation Results (Time-Schedule).....	28
Table 5.	Monte Carlo Simulation Results (Probability of Completion)	29
Table 6.	Normalized Time-Schedule Monte Carlo Simulation Results.....	30
Table 7.	Probability of Completion Compared to Monte Carlo Median Results	31
Table 8.	NAVAIR Project Management Tool Functions	36
Table 9.	Generic Project Planning Process	39
Table 10.	Decomposition of the Project Initialization Activity	41
Table 11.	Decomposition of the Project Network Design Activity	43
Table 12.	Decomposition of the Simulation and Analysis Activity	44
Table 13.	Generic Project Management Process	46
Table 14.	Decomposition of the Project Performance Tracking Activity.....	48
Table 15.	Decomposition of the Project Management and Corrective Action Activity ..	51
Table 16.	Stakeholder Requirements Prioritization	59
Table 17.	Prototype Software Application – Team Merica Functional Implementation	90

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ACWP	actual cost of work performed
BAC	budget at completion
BCWP	budgeted cost of work performed
BCWS	budgeted cost of work scheduled
BFA	business financial analyst
COTS	commercial off-the-shelf
CPI	cost performance index
CPM	critical path method
CV	cost variance
DAU	Defense Acquisition University
DOD	Department of Defense
DoE	design of experiments
EAC	estimate at completion
EV	earned value
EVM	earned value management
GAO	Government Accountability Office
HCD	human centered design
IBR	integrated baseline review
IPT	integrated project team
IT	information technology
LE	lead engineer
M&S	modeling and simulation
MC	Monte Carlo
ML	most-likely
MS	Microsoft
NAVAIR	U.S. Navy Naval Air Systems Command
NMCI	Navy Marine Corps Internet
NPS	Naval Postgraduate School
OOP	object oriented programming
OPP	Open Plan Professional

PE	project executor
PM	project manager
PMA	project management authority
PV	planned value
RM	risk management
ROM	rough order of magnitude
SME	subject matter expert
SPI	schedule performance index
SRA	schedule risk assessment
SV	schedule variance
V&V	verification and validation
VAC	variance at completion
WBS	work breakdown structure
WMS	Workforce Management System

EXECUTIVE SUMMARY

Current methods utilized within the Department of Defense (DOD) for estimating project budget and schedule frequently results in overruns of the initial estimates. Many of the prevailing methods utilized by the DOD rely on point estimates, subject matter expert (SME) opinions, extrapolation from historical data, and/or the application of learning curves. These methods are inadequate in considering uncertainty within specific project tasks and accounting for normal performance variance during project execution. This shortfall in current estimation practices can cause significant deviations from the initial baseline estimates. Additionally, current methods for generating estimates and tracking project performance are often difficult to use and update. This study designed a proof of concept software application that accounts for cost and schedule uncertainties within project tasks and provides realistic and achievable estimates for the duration of the project. The proof of concept also provides stakeholder management staff a method to track actual project performance in terms of cost and schedule in relation to simulated estimates.

Research was conducted into current methods of cost and schedule estimating as well as project tracking. Two of the primary tools used, Microsoft (MS) Project and Primavera P6 Professional Project Management (P6) by Oracle, require additional software tools such as Open Plan Professional (OPP) by Deltek or Full Monte by Barbecana to perform uncertainty analysis. MS Project and P6 utilize Critical Path Methods (CPM) based on point estimates generated from other programs to determine the path of tasks that results in the longest schedule. For project tracking, earned value management (EVM) is frequently utilized to determine if a project is on track to meet the initial baseline estimate. However, the baselines from which earned value is compared do not consider uncertainty. Based on the research conducted, no single tool could be found that performs all of the tasks associated with estimating and tracking while also accounting for uncertainty.

A modeling experiment was performed to explore how different factors affected the probability that the project would finish on schedule. The experiment pursued how

the number of potential critical paths, length of the critical path, task interdependence, and risk uncertainty impacted the project schedule. A design of experiment (DoE) was created in which the developed software application was used to perform a Monte Carlo (MC) analysis. Results indicated that using minimum and maximum duration estimates can result in schedule estimates that under-estimate by as much as 45% or over-estimate by as much as 35% depending on the assumption for the probability distributions that characterize the uncertainties for each specific task. Using most-likely, mean, and median estimates resulted in durations within 10% of the median project duration determined by the Monte Carlo (MC) simulation.

To develop the software application prototype, a Human Centered Design (HCD) systems engineering approach was utilized. The HCD approach focuses on developing a product that actively considers the end-user throughout the design process. To facilitate stakeholder discussion, critical for the success of HCD, a list of specifically tailored research questions was generated and several Naval Air Systems Command (NAVAIR) stakeholders were interviewed. A thorough stakeholder analysis was then completed outlining their current methods for estimating and tracking budgets and schedules. The analysis also included the challenges associated with various project sizes and the tools being utilized. Using the information gathered from the stakeholder interviews an operational-based scenario was generated to represent the desired operational intent of the stakeholders for the prototype software application.

Requirements were also generated from the stakeholder interviews. Notes from the stakeholder interviews were organized into specific categories based on software development and project phases. Similar notes were stacked together and the frequency of occurrence indicated a higher preference for stakeholders. Several stakeholder themes emerged including a desire for import functionality, database utilization, collaborative operations, geographically distributed access, usability, modifiability, uncertainty consideration, real-time project execution status, and quick project status determination. Once the notes were organized, they were converted into derived testable requirements and prioritized.

Architectural development was conducted by looking at the stakeholder interview notes and the established requirements. A high level functional hierarchy was created to show the top level functional breakdown and then an IDEF0 was developed to make sure all the requirements were addressed. Data flow, controls, and mechanisms were then identified for each function to determine the functional flow of the prototype software application and to identify where feedback was appropriate. A large emphasis was placed on system feedback within the architectural design to provide the system with continuous updates.

For software development, the Scrum process was determined to be the most appropriate method of development due to time constraints and because it complements HCD philosophies. Scrum incorporated requisite stakeholder interaction of HCD and allowed for requirements reprioritization and refinement. Several ancillary products were produced including high level graphical story boards, MS Excel interactive prototypes, architectural views, and functional flows. Although not software specifically, these products facilitated HCD and removed layers of separation between the developers and the stakeholders. The C# programming language was selected for the development environment and GitLab was used for configuration management. The open-source tool Doxygen, which is a documentation generator used with programming languages such as: C++, C and C# among others, was used to properly document the source code of the prototype software application and to generate a user's manual. The final prototype demonstrated the design concept and achieved limited functionality.

Current methods of estimating a project's budget and schedule and then tracking the progress through completion are cumbersome and fail to consider uncertainty. Stakeholder participation through HCD facilitated the development of a prototype software application that has demonstrated the potential to address these shortfalls. The concept of incorporating the functionality of several existing software applications into one streamlined software application was successful and satisfied the goals of the study. With further refinement, a full scale version of this prototype could be utilized by DOD project managers to dramatically improve the reliability and accuracy of project budget and schedule estimates and provide an easy way to track project progress.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Current practices of estimating budget and schedule for many projects within the DOD often result in programs that exceed planned estimates. In some cases, uncertainties in task duration and cost were not fully considered during the project planning phases. Prevailing procedures rely on fixed value or point estimates for budget and schedule that do not account for uncertainties related to the task(s) under consideration. The study presented within this document examines the effects of considering cost and duration uncertainties in the project planning and execution processes. The systems engineering process, detailed in this chapter, was used to create a proof of concept in the form of a prototype software application that facilitates incorporating uncertainty in the planning and management of complex engineering projects.

This chapter discusses the purpose of this study and the overall systems engineering process that was used to develop the software application. Section A presents the challenges experienced by stakeholders related to their current project estimation and tracking practices. Section B establishes the goals of this study to examine the benefits of considering uncertainties in project plans. This section also examines the functional benefits of the software tool that has been designed. Section C depicts the boundaries that were established by the design team when developing the proof of concept software application. Section D articulates the research questions that guided the design team's interaction with the stakeholders of this project and the outcome of this study. Given the vast volume of information about the topic of uncertainty and project management, the design team used the assumptions listed in Section E to further constrain the scope of this study. Section F depicts the System Engineering process that the design team followed for this study in order to develop the proof of concept software application prototype. Finally, Section G provides a roadmap of the remaining chapters of this document.

A. PROBLEM STATEMENT

Current methods of estimation do not adequately account for schedule performance variance. Often, estimates for the task durations and costs are made by the

person generating the project plan, usually with one check from an engineer or supervisor who has an understanding of the tasks involved. As the plans begin execution, the duration of tasks are likely to change as real world conditions present themselves. The product plan becomes an evolving web of variations that require shifts of resources to correct the tasking and attempt to keep the program on track. While complicated to use, MS Project has become the primary tool to establish a production schedule. A limitation of MS Project is that it does not perform statistical analysis to determine critical and near critical paths of the project under development. Updating previously generated schedules in MS Project was found to be cumbersome and a time consuming task. Also, many organizations track project execution with tools that are not integrated with their planning tools. Using separate tools can lead to confusion while tracking progress and may result in losing track of the actual progress of the project. It may also cause delays in project performance feedback, which can hamper efforts to maintain the project on the planned budget and schedule. No software tool was found that can create a project plan schedule, examine likely areas of delay and cost overrun, and is relatively easy to use. Determination of likely areas of delay and cost overrun could be used to create timely interventions and prevent further schedule delays.

B. GOALS AND OBJECTIVES

The goal of this study was to develop a proof of concept system that 1) accounts for cost and schedule uncertainties within the activities of a project and provides realistic, achievable estimates for project cost and schedule over the duration of the project, and 2) provides the ability for stakeholder management staff to track actual project performance in terms of cost and schedule against the previously generated estimates.

The result at the completion of this study was a proof of concept prototype software application that demonstrated basic functionality of the following management activities: schedule importing from MS Project, schedule planning, cost-schedule risk analysis, provides management insight similar to EVM, and provides the desired user functionality requested by the stakeholder. Further development is warranted to turn the basic functionality of the prototype software application into a fully integrated software

application. Stakeholder collaborations and feedback on prototype development were essential to the success of this system.

C. SCOPE

The DOD recognizes the importance of capturing cost and duration uncertainties on budget and schedule estimates that guide the development of new projects. According to the *DOD Risk Management Guide for Defense Acquisition Programs* (2014), the term uncertainty is defined as “the confidence level associated with each alternative’s schedule estimate, proposed performance and associated technical risk” (64). There is substantial documentation regarding which statistical models should be used when considering cost and duration uncertainties. This study focused on the implementation of such models in order to consider the intrinsic uncertainties of the tasks under analysis. The following list defines the scope of the study.

1. This study focused on the development of a software application prototype that considers uncertainties in the stakeholders’ project planning and execution processes.
2. This study did not focus on the industrial processes of the planned tasks.
3. The software application prototype addressed project networks with activities that have a “Finish to Start” relationship.
4. This study used existing research and guidance to adopt uncertainty consideration in the existing budget and schedule planning processes of the stakeholder.
5. This study did not develop new risk factor multipliers in order to establish the cost and duration uncertainty of a task.
6. The developed software application prototype is not intended to be deployed on a Navy Marine Corps Intranet (NMCI) asset.
7. This study did not consider the funding required for developing a software application that can be deployed on an NMCI asset.

D. RESEARCH QUESTIONS

Prevalent estimation practices consist of producing a discrete dollar or duration value that becomes the budget and/or schedule of a project. Identifying and accounting for the uncertainty of different tasks in a project should result in more realistic budget and schedule estimates that support management processes. Without conducting a formal study of the existing stakeholder’s project planning and execution processes, it would have been impractical to assume that they are not considering uncertainties in their

estimates. It would have also been presumptuous to assume that an automated approach of developing uncertainty products would result in a better project management performance. Two research questions were developed to understand current budget and schedule estimation practices of the stakeholder. Research Question #1: “What are the influencing factors and constraints of the current planning processes?” Research Question #2: “What are the required interfaces of the current cost and schedule planning system with focus on hardware, software, and human systems?”

Once the stakeholders’ estimation processes were analyzed, the next part of the study determined how the stakeholders were using the products of the project planning phases to track the progress of the project and this led to Research Question #3: “How are the stakeholders tracking and analyzing the progress of the planned activities in a project?” To further understand the stakeholders’ project execution processes and how uncertainty is applied, two subordinate-questions were developed. Research Question #3a: “In what way is uncertainty considered during this process?” Research Question #3b: “What are the consequences that come from not adequately considering these uncertainties during the project execution phases?”

The answers of these three research questions guided the stakeholders’ collaboration to determine what outputs, and in what format, proved to be the most beneficial for depicting the budget and schedule estimates on a project that considers tasks’ uncertainties. This set up Research Question #4: “What information would be considered valuable for depicting an overall picture of the cost and schedule estimates of the project under analysis?”

With complete understanding of the stakeholders’ project planning and execution processes the following research question was developed to address the problem statement in Research Question #5: “How can the current process of planning and project management be modified to account for the uncertainty of cost and duration?” The answer to Research Question #5 was decomposed to scope human factors requirements for the architecture under development. Research Question #5a: “How could a tool be designed to facilitate or automate this new process?” Research Question #5b: “How much user involvement in terms of training is desirable?”

E. ASSUMPTIONS

Section C of this chapter narrowed the scope of this project. Additionally, assumptions were made in order to guide the development efforts. In an effort to address the needs of different users of the developed proof of concept software application, stakeholders' interview feedback was collected and synthesized into a generic operational scenario that was used to verify and validate the outputs of the System Design and Development Process of this study. The generated operational scenario was used as a guide for the modeling and simulations that were ultimately used to answer research questions 4 and 5. It is assumed that the operational scenario is fairly representative of project planning efforts within some DOD organizations. The developed proof of concept identifies the uncertainties of the current processes without the intention of directly affecting changes on these processes. Team Merica, authors of this report, does not make any claim that the generated products of this study will be implemented in the stakeholders' organization. It is also assumed that realistic estimates and accurate execution data will be entered into the software prototype during use. Additional assumptions are identified and explained within the applicable sections of this report.

F. SYSTEMS ENGINEERING PROCESS

The systems engineering process that was followed by the design team of this study, is the V-Model of systems engineering life cycle depicted in Figure 1.

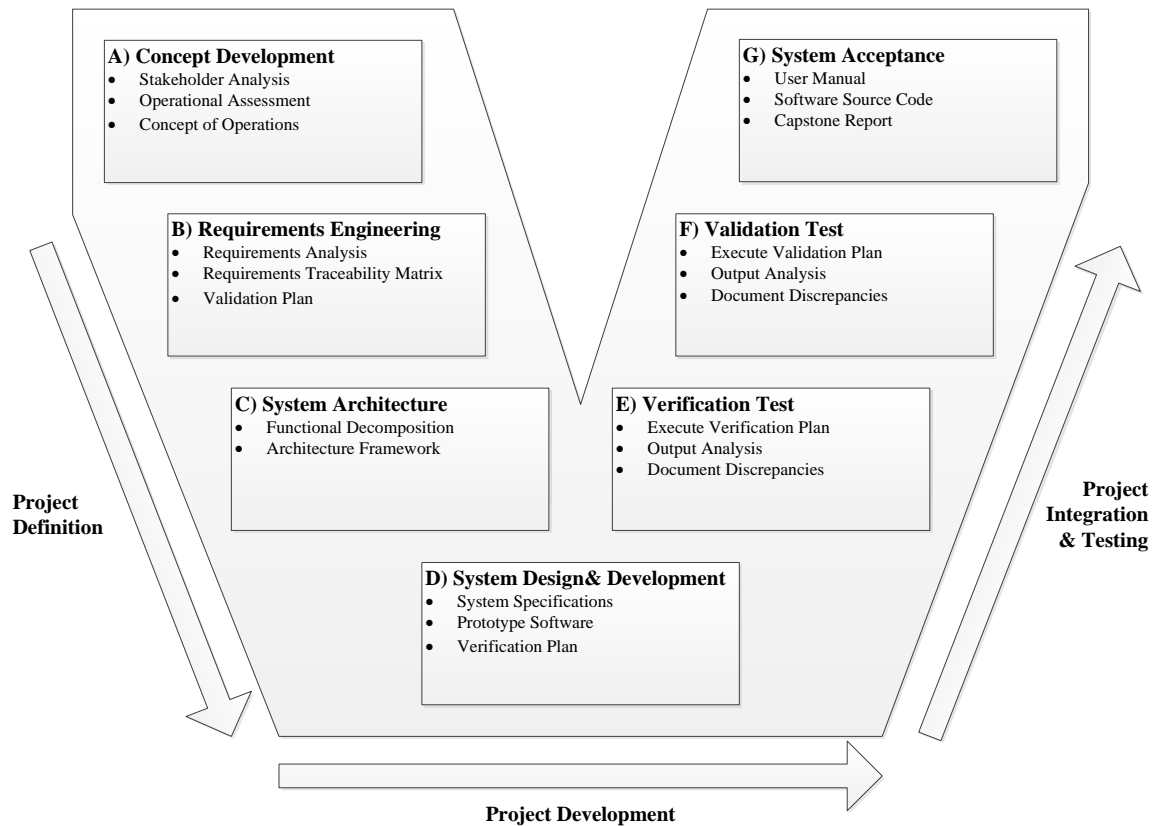


Figure 1. Capstone System Engineering Process (after MITRE 2014, 2)

1. Concept Development

During the concept development phase, the development team utilized the problem statement, goals, and objectives contained within this document to aid in a “needs” discussions with applicable stakeholders. These discussions helped to develop a concept of operation that allowed the team to capture the users’ expectations of the project. The final products of these discussions were an analysis of needs and the requested functionality of the system. The team applied human-centered-design methodology by identifying usage practices for the current systems, developing an operational-based scenario, and exploring features of the systems currently in use.

2. Requirements Engineering

The development team translated the identified needs into overall system requirements. The requirements were managed through a requirements matrix found in

Appendix A that stated every requirement, explicit and derived. Having identified the overall system requirements, the development team generated a system validation plan in order to make sure that every requirement was testable and complete.

3. System Architecture

In this phase of the systems engineering process, the team performed a functional decomposition of the requirements which provided the information required to generate a system's architecture. The outputs of this phase included: a functional description in IDEF-0 format to demonstrate the desired functionality of the system, sequence diagrams of the operational-based scenario used to ensure the IDEF-0 captured all desired stakeholder functionality, required timing of events, and aided in the decomposition of the functional requirements which produced a system architecture that addressed the user's needs.

4. System Design and Development

After an initial design concept was defined, the development team used the developed architecture to generate a list of derived systems specifications that guided the development of the software prototype. The prototype was constructed in incremental builds, starting with a conceptual model and then expanding the capability from that point outwards to ultimately encapsulate all of the system modules necessary to fulfill the requirements. The development team openly welcomed stakeholders to participate in providing feedback into the design after the completion of each incremental build. Due to the limited schedule of the project, the reviews of these builds were performed internally by the design team with limited participation of the stakeholders.

5. Verification Test

Verification of each module requirement successfully integrated into the prototype was performed following each build release, with the intention of catching and addressing software bugs at each of these gates. Once the integration was complete the prototype was tested and verified against the system requirements, architecture, and

stakeholder feedback. Once the prototype passed the verification process it then moved forward into validation test.

6. Validation Test

The validation test was executed through the use of the established operational-based scenario. The basic functionality of the prototype software application was demonstrated to stakeholders with emphasis on software interfaces, activity generation and data feedback. Data was collected from these interactions in order to document other desirable functionalities that could be implemented in the future.

7. System Acceptance

During this phase the development team delivered the developed source code of the prototype software tool and a copy of this reports that documents the features of the prototype software application. Additionally, this capstone report includes various artifacts generated for this project and answers to the research questions.

G. REPORT ROADMAP

The purpose of this study was to address the question: “How does considering cost and duration uncertainties in the project planning and execution process help to generate realistic plans and efficiently manage complex engineering projects?” To answer this question, the study group investigated the existing barriers for the implementation of project management processes that consider cost and duration uncertainty. A proof of concept in the form of a software application was developed following sound system engineering processes in order to meet the stakeholder needs.

To examine and answer the question of how to improve project planning by incorporating cost and duration uncertainties in the project planning and execution phases, the study team looked at tools and methods for creating cost and schedule estimates and the challenges of examining schedules with many possible network paths. During this study an analysis was conducted into how current users are performing cost and schedule estimates. This analysis facilitated the development of a generic operational-based scenario that delineates the intended use of the prototype software

application and is fully described in Chapter II. The artifacts generated from the stakeholders' interviews and the developed operational-based scenario, refined the requirements that were used to develop the prototype software application. These requirements were prioritized using stakeholders inputs as the driving force and are presented in Chapter III. After establishing the user requirements for a system capable of incorporating budget and schedule uncertainties into the project planning and execution phases, the design team developed a functional architecture that captures the needed capabilities of the prototype software application. This functional architecture is presented in Chapter IV. A storyboard, in the form of a wireframe prototype, was put together using the established functional architecture. This storyboard was used to continuously capture the feedback of the stakeholders and to make sure that the developed software application meets their needs. In Chapter V of this document the design team presents the initial prototype of the software application that was created as part of this study. Chapter VI of this document presents the conclusions of this study and recommendations for future work to further expand the capabilities of the prototype software application.

THIS PAGE INTENTIONALLY LEFT BLANK

II. PROBLEM SPACE EXPLORATION

A. BACKGROUND

For the purposes of this study, the project planning phases include all the management and engineering actions involved in the development of a baseline project plan. The execution phase of the project starts once the contract is awarded and continues until completion or cancelation of the development effort.

1. Cost Estimation Point Estimates

There are several types of cost estimating methods used, alone or in combination, during project planning phases to develop an initial point estimate or baseline. These include analogous, parametric, and engineering build-up. Analogous and parametric methods are used very early in the program planning phases. The analogous compares cost data from previously executed similar programs and makes cost adjustments based on project differences. This method is used when little detailed information about the project exists. The parametric method uses statistical cost data relationships with key system parameters to calculate a duration or cost. The engineering build-up method estimates task cost and duration by breaking them into smaller pieces, typically using the project work breakdown structure (WBS), and adds them together. Strengths and weaknesses of the three estimating methods can be found in Figure 2.

Other cost estimating methods include SME opinions, extrapolating from historical data, and applying learning curves. SME opinion is considered extremely subjective but is useful when no historical data is available. Learning curve application in cost estimation is useful for determining the estimates for a first production unit, the average unit, or every unit produced because unit cost varies by the number of units produced. The above methods' resulting individual WBS estimates are then added for the entire project and the total is considered the total project point estimate (GAO 2009).

Method	Strength	Weakness	Application
Analogy	<ul style="list-style-type: none"> ▪ Requires few data ▪ Based on actual data ▪ Reasonably quick ▪ Good audit trail 	<ul style="list-style-type: none"> ▪ Subjective adjustments ▪ Accuracy depends on similarity of items ▪ Difficult to assess effect of design change ▪ Blind to cost drivers 	<ul style="list-style-type: none"> ▪ When few data are available ▪ Rough-order-of-magnitude estimate ▪ Cross-check
Engineering build-up	<ul style="list-style-type: none"> ▪ Easily audited ▪ Sensitive to labor rates ▪ Tracks vendor quotes ▪ Time honored 	<ul style="list-style-type: none"> ▪ Requires detailed design ▪ Slow and laborious ▪ Cumbersome 	<ul style="list-style-type: none"> ▪ Production estimating ▪ Software development ▪ Negotiations
Parametric	<ul style="list-style-type: none"> ▪ Reasonably quick ▪ Encourages discipline ▪ Good audit trail ▪ Objective, little bias ▪ Cost driver visibility ▪ Incorporates real-world effects (funding, technical, risk) 	<ul style="list-style-type: none"> ▪ Lacks detail ▪ Model investment ▪ Cultural barriers ▪ Need to understand model's behavior 	<ul style="list-style-type: none"> ▪ Budgetary estimates ▪ Design-to-cost trade studies ▪ Cross-check ▪ Baseline estimate ▪ Cost goal allocations

Source: © 2003, MCR, LLC, "Cost Estimating: The Starting Point of EVM."

Figure 2. Three Cost Estimating Methods Compared (from GAO 2009, 108)

2. Cost Estimation Uncertainty

Uncertainty is always present within initial project point estimates. Not accounting for uncertainty frequently results in low budget and schedule estimates and ultimately, project overruns. Figure 3 shows several common sources of project cost uncertainty. Uncertainty analysis using S-Curves or cumulative probability distributions is one way to determine if original project point estimates are realistic. The S-curve shown in Figure 4 can be generated from a MC simulation and can show uncertainty implications of point estimates. The S-curve shows the cumulative probability (vertical axis) of a particular cost outcome (horizontal axis) with respect to all the different task distributions. S-curve data identifies overall project outcome uncertainty and may be used initially to justify the project management reserve, or funds retained outside of the initially estimated project cost to mitigate possible cost overruns (GAO 2009). The S-curve is also useful in making more informed project go or no-go decisions.

The accuracy of the S-curve is dependent on the quality of risk data which is obtained by any combination of the following: SME opinion or using cost growth factors, technology readiness levels, mathematical approaches using optimistic, most-likely and pessimistic ranges (i.e., three-point estimates), risk cubes, and risk scoring (GAO 2009).

Choosing an appropriate probability distribution for the overall point cost estimate is difficult because each WBS task item may be best modeled differently. A more realistic approach is to specify the distribution shape using three point estimates for each element or task. A MC analysis may then be used to statistically sum all the differing probability tasks to the point estimate probability distribution as depicted in Figure 5.

Uncertainty	Definition	Example
Business or economic	Variations from change in business or economic assumptions	Changes in labor rate assumptions—e.g., wages, overhead, general and administrative cost—supplier viability, inflation indexes, multiyear savings assumptions, market conditions, and competitive environment for future procurements
Cost estimating	Variations in the cost estimate despite a fixed configuration baseline	Errors in historical data and cost estimating relationships, variation associated with input parameters, errors with analogies and data limitations, data extrapolation errors, optimistic learning and rate curve assumptions, using the wrong estimating technique, omission or lack of data, misinterpretation of data, incorrect escalation factors, overoptimism in contractor capabilities, optimistic savings associated with new ways of doing business, inadequate time to develop a cost estimate
Program	Risks outside the program office control	Program decisions made at higher levels of authority, indirect events that adversely affect a program, directed funding cuts, multiple contractor teams, conflicting schedules and workload, lack of resources, organizational interface issues, lack of user input when developing requirements, personnel management issues, organization's ability to accept change, other program dependencies
Requirements	Variations in the cost estimate caused by change in the configuration baseline from unforeseen design shifts	Changes in system architecture (especially for system of systems programs), specifications, hardware and software requirements, deployment strategy, critical assumptions, program threat levels, procurement quantities, network security, data confidentiality
Schedule	Any event that changes the schedule: stretching it out may increase funding requirements, delay delivery, and reduce mission benefits	Amount of concurrent development, changes in configuration, delayed milestone approval, testing failures requiring rework, infeasible schedule with no margin, overly optimistic task durations, unnecessary activities, omission of critical reviews
Software	Cost growth from overly optimistic assumptions about software development	Underestimated software sizing, overly optimistic software productivity, optimistic savings associated with using commercial off-the-shelf software, underestimated integration effort, lack of commercial software documentation, underestimating the amount of glue code needed, configuration changes required to support commercial software upgrades, changes in licensing fees, lack of support for older software versions, lack of interface specification, lack of software metrics, low staff capability with development language and platform, underestimating software defects
Technology	Variations from problems associated with technology maturity or availability	Uncertainty associated with unproven technology, obsolete parts, optimistic hardware or software heritage assumptions, feasibility of producing large technology leaps, relying on lower reliability components, design errors or omissions

Source: DHS, DOD, DOE, NASA, OMB, SCEA, and industry.

Figure 3. Potential Sources of Program Cost Estimate Uncertainty (from GAO 2009, 160)

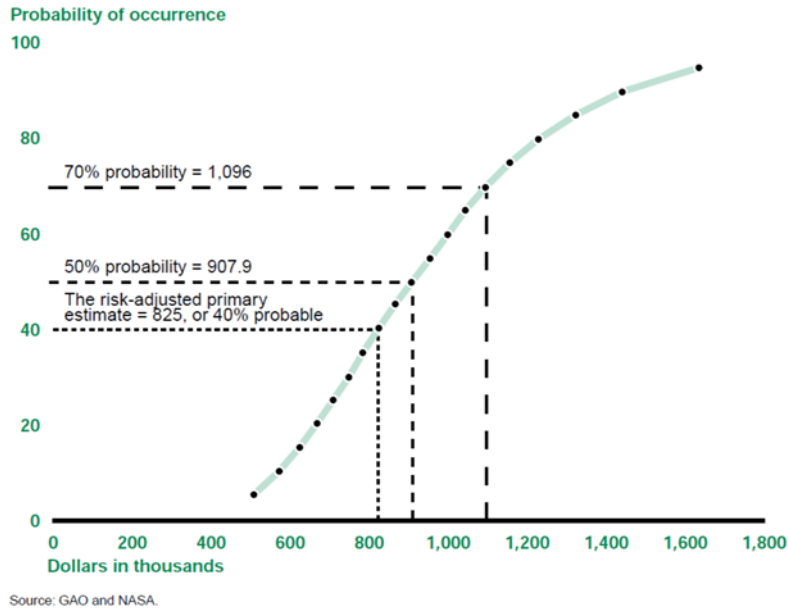


Figure 4. Sample Cumulative Probability distribution or S-Curve for Cost Estimates (from GAO 2009, 157)

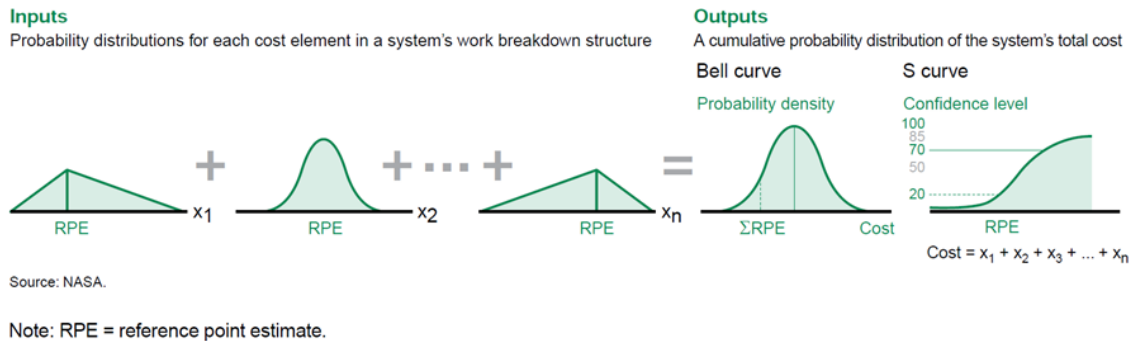


Figure 5. Point Estimate Probability Distribution Driven by WBS Distributions (from GAO 2009, 168)

3. Existing Cost Estimation Tools

There are several COTS tools available to compute project cost uncertainty using Monte Carlo analysis. Two examples of cost estimation software include Crystal Ball by Decisioneering and @Risk by Palisade. Both researched tools require cost data input into spreadsheet models using a program such as MS Excel. Once the data has been inserted, both researched tools compute the MC and generate uncertainty results with histograms

and S-curves. Both tools have a sensitivity analysis feature which shows the tasks with highest project impact.

4. Schedule Estimation Point Estimates and Critical Path Method

Schedule estimation is used to develop a timeline for project tasks and identifies key milestone dates. Similar to cost point estimates, schedule point estimates facilitate many of the same estimating methods such as analogous, parametric, and engineering build-up which are discussed above. SME opinions are also commonly used to develop initial task duration estimates.

The aforementioned methods establish individual WBS task time estimates which may be organized sequentially in the order they must be accomplished. Tasks requiring completion prior to other tasks are considered predecessors, and tasks that must begin after another task's accomplishment are successors. Preceding and succeeding task dependencies must be identified and linked to create an activity network such as the one shown in Figure 6. Once an activity network is generated a critical path may be identified. The critical path is the network path that has the longest duration in accomplishment time and dictates the project completion date. If a critical task is delayed then the entire project will be delayed unless productivity is increased elsewhere on the critical path. Float or slack in a schedule is the time any task in the network can slip before other tasks are affected. A project critical path generally has tasks with the least slack (GAO 2009). The CPM is a project management style that uses a determined critical path to ensure the project stays on the planned schedule. With schedule delays, it is possible that during project execution the critical path may change.

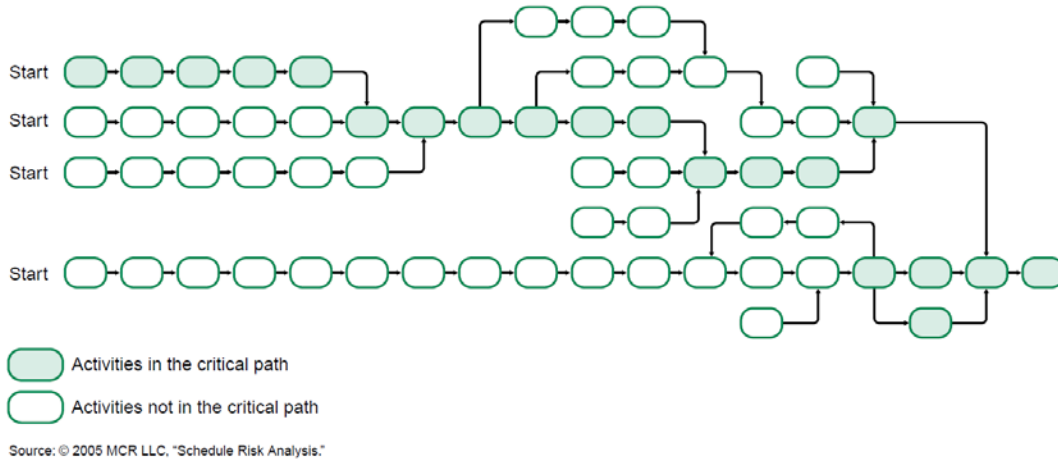


Figure 6. An Activity Network (from GAO 2009, 219)

5. Schedule Estimation Uncertainty

Cumulative probability distributions or S-curves may also be developed for schedule, similar to cost as shown in Figure 7. There are three steps to create a schedule risk analysis and develop project S-curves (Hulett 1996):

1. Create the activity network and critical path or CPM schedule for the project.
2. Use the schedule estimation techniques discussed above to estimate uncertainty. This usually entails identifying the best case, worst case, and most likely duration for each task.
3. Conduct the risk analysis using a MC simulation method similar to the theory shown in Figure 5.

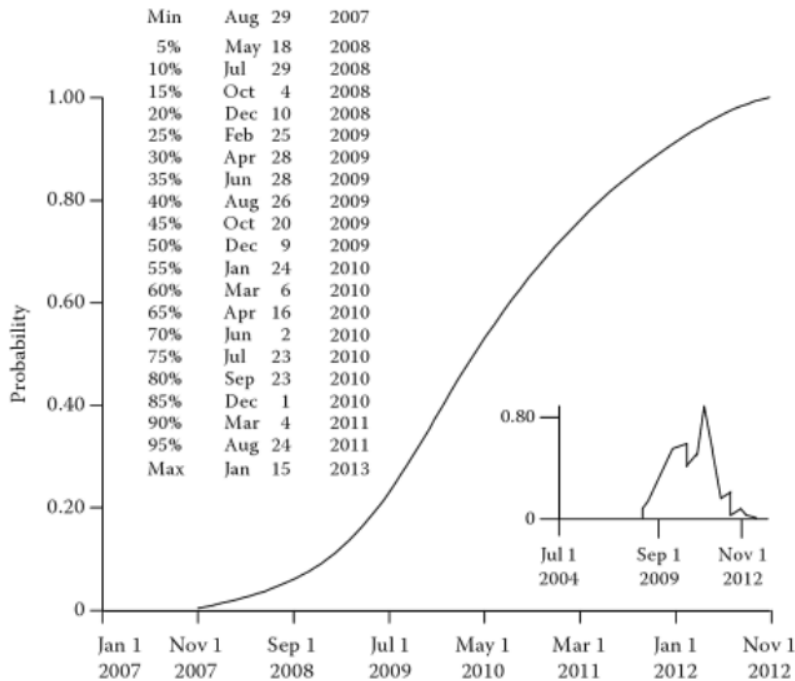


Figure 3.6 Sample cumulative probability distribution.

Figure 7. Sample cumulative probability distribution or S-curve for schedule (from Pritchard 2015, 47)

6. Existing Schedule Estimation Tools

Examples of COTS tools that facilitate building CPM schedules include MS Project and Primavera P6 Professional Project Management (P6) by Oracle. These tools alone do not compute MC simulations. The tools researched use or import the MS Project or P6 schedule data to then do an uncertainty or risk analysis. Examples of schedule uncertainty tools include Open Plan Professional (OPP) by Deltek which works with MS Project schedule data and Full Monte by Barbecana which works with MS Project or P6 schedule data. These risk tools provide histograms and S-curves for project schedule and Full Monte has capability to also conduct sensitivity analysis.

7. Integrated Cost and Schedule Estimation Practices

It is possible to include uncertainty when estimating both cost and schedule which results in a more accurate, collaborative, and comprehensive project estimate. Benefits of

integrating cost and schedule, given proper tools, include better reporting, task to schedule linking which simplifies data updating, and task to risk linking for building in appropriate task slack time as required (Williams 2013).

a. Concerns

For several reasons, it is more common for the project cost and schedule estimates to be computed separately. Issues related to integrating cost uncertainty estimates with schedule uncertainty estimates are as follows (Hulett 2004):

1. Schedule and cost estimates differ in detail. Project schedules deviate from WBS structure more often than cost elements.
2. Calculating schedule uncertainty requires duration of tasks or activities and most schedules are date oriented. According Hulett, “dates and durations are not equivalent, e.g., in this schedule, we cannot tell when the integration and test phase begins because of the uncertainty in its predecessor activities” (Hulett 2004).
3. Generally different software tools are used to compute schedule risk and cost risk which are not compatible with one another.
4. Computing cost uncertainty at a high level requires average labor resources and average compensation per day which project managers often do not think of cost in averages.

Additionally, integrating cost and schedule uncertainty may not suit all projects for the following reasons:

1. Integration tools are difficult to use without training.
2. Developing three-point estimates and applicable distribution types may be time consuming and projects are best managed provided data as close to real-time as possible.

b. Existing Tools

Tools that integrate cost and CPM schedule are available. Many require the use of base cost or schedule data software to build initial estimates. The tools then import the data from the base sources. Examples include Primavera Risk Analysis by Oracle, @Risk for Project Management by Palisade, and Winsight by Deltek. Primavera Risk Analysis requires P6 schedule data and is fairly comprehensive though, the most costly of the integrated cost and schedule risk software. Winsight must be used with other Deltek software including OPP and Cobra and imports data from MS Excel and MS

Project. Winsight is unique in that it additionally captures EVM statistics, which is a required functionality for large NAVAIR projects. Winsight contains cumulative cost and schedule distributions but does not allow the risk detail Primavera Risk Analysis provides; specifically, Primavera Risk Analysis allows each task to identify with a choice of one of several probability distributions. @Risk also uses MS Excel and MS Project base data and provides integrated project risk including Monte Carlo analyses, sensitivity analyses including reports such as histograms and S-curves. Existing tools researched and their functions are listed in Table 8.

8. Earned Value Management

One commonly accepted cost and schedule execution tracking practice is EVM. For many projects, “earned value is a means to objectively measure how much work a contractor has accomplished” (Undersecretary of Defense [AT&L] 1999, 1). The use of EVM has not been limited to government contractor oversight, but has also been commonly used in industry. According to Sutter, “EVM enables managers to anticipate problems and to take pre-emptive action” during project execution (Suter 2006, 407). During the planning phase the EVM method uses an initial cost and schedule baseline which is a “basis for determining whether and when enough information is available to satisfy specific confidence levels for estimates” (Suter 2006, 407). During program execution real cost and schedule data are compared to the baseline to assess project status. Figure 8 shows the graphical representation of the EVM theory. The blue line of the graph represents the established time-phased budget plan that is enacted at the beginning of the project execution phase. The green line in the figure shows the actual project expenditures since the project start date and forecasted all the way through project completion. With this information the management personnel can calculate the current performance of the project and determine if management actions are required. The Defense Acquisition University (DAU) definitions of the EVM terms used in Figure 8 are presented in Table 1.

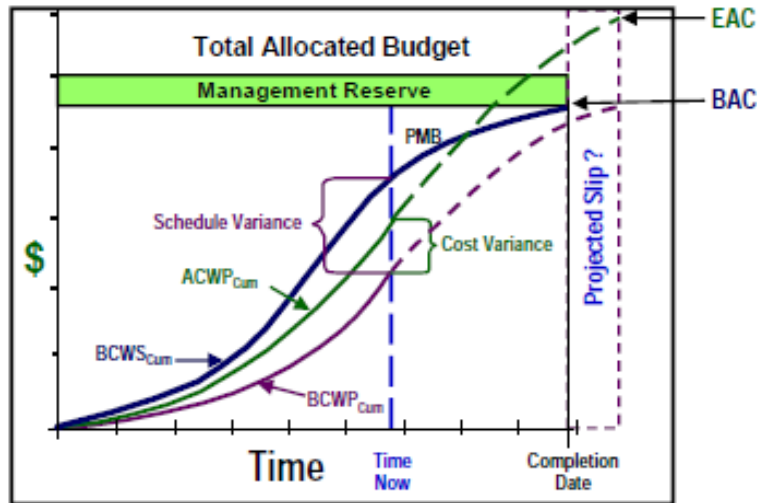


Figure 8. DAU Earned Value Management Gold Card (from DAU 2015)

Table 1. DAU EVM Terms Definition (from DAU 2015)

EVM Term	Description
BCWS	Budgeted cost for work scheduled or planned value (PV)
BCWP	Budgeted cost for work performed or earned value (EV)
ACWP	Actual cost of work performed
BAC	Budget at completion
EAC	Estimate at completion which is a sum of ACWP to date in addition to estimated remaining work costs
CV	Cost variance which is $BCWP - ACWP$
SV	Schedule variance which is $BCWP - BCWS$
VAC	Variance at completion which is $BAC - EAC$
CPI	Cost performance indicator which is $BCWP / ACWP$
SPI	Schedule performance indicator which is $BCWP / BCWS$
PMB	Performance Measurement Baseline

EVM uses CV, SV, and VAC to determine whether a project is on track to meet the initial baseline estimate. If CV is ever less than 0, the cost is overrunning the baseline. If SV is ever less than 0, the project is behind schedule. Similarly, if VAC is less than 0 the project will likely be overrun at completion (Humphreys 2015). A CPI or SPI below 1.0 indicates project performance is falling short of the plan (Hillson 2004).

One drawback of EVM is that the method relies on uncertain initial baselines. Initial project planning baselines are inherently inaccurate, regardless of the project technology readiness level and organizational capability level, due to variables in project execution such as material availability, integration issues, and other unknown development issues (Suter 2006). Generally, as the project progresses toward completion cost estimating becomes more accurate because the cost estimates incorporate actual data, and there is less and less to estimate. Suter blames EVM project schedule slips on data delay and on schedule changes that may not be recognized while they are occurring but later when data is compiled. The result is inaccurate EVM calculations and inability to control the program due to a lack of realistic progress measures.

9. Earned Value Management Uncertainty Considerations

Conceptually, EVM uses a single project baseline and does not consider uncertainty on a task by task level aside from the management reserve which is estimated when the initial baseline is developed. However, integrating risk management (RM) with EVM may be beneficial. Hillson agrees that there is “possible synergy” between integrating the forward looking risk management and the past assessment EVM approaches (Hillson 2004, 2). Hillson defines risk “as any uncertainty that, if it occurs, would have a positive or negative effect on achievement of one or more project objectives” (Hillson 2004, 4). RM is intended to proactively address project uncertainty to monitor and ensure projects are completed on time and within budget. Hillson uses EVM CPI and SPI measures to determine project plan deviation. He incorporates RM and EVM using the approach outlined in Figure 9. Hillson uses a full risk assessment using three point estimates and probability distributions followed by MC simulations for both schedule and cost similar to that discussed above and shown in Figure 10. Hillson recommends “from a combined approach to EVM and RM is to use the expected value cumulative profile from a quantitative time-cost risk analysis as the baseline for BCWS. In other words, the central S-curve in Figure 10 would be used as the baseline instead of the original S-curve” (Hillson 2004, 3).

<p>1. Creating the baseline spend plan (BCWS/PV)</p> <ul style="list-style-type: none"> a. Develop costed WBS to describe scope of work, without hidden contingency b. Produce fully costed and resourced project schedule c. Assess estimating uncertainty associated with initial time/cost estimates d. Perform risk identification, risk assessment and response development e. Quantify time and cost risk exposure for each risk, taking account of the effect of agreed responses f. Create integrated time/cost risk model from project schedule, reflecting both estimating uncertainty (via 3-point estimates) and discrete risks (via stochastic branches) g. Perform Monte Carlo simulation on integrated risk model to generate “eyeball plot” h. Select risk-based profile as baseline spend profile (BCWS/PV); it is most common to use the “expected values”, although some other confidence level may be selected (say 80%)
<p>2. Predicting future outcomes (EAC)</p> <ul style="list-style-type: none"> a. Record project progress and actual cost spent to date (ACWP), and calculate earned value (BCWP) b. Review initial time/cost estimates for activities not completed, to identify changes, including revised estimating uncertainty c. Update risk identification, assessment and quantification, to identify new risks and reassess existing risks d. Update integrated time/cost risk model with revised values for estimating uncertainty and discrete risks, taking account of progress to date and agreed risk responses e. Repeat Monte Carlo simulation for remaining portion of project to generate updated “eyeball plot” f. Select risk-based calculation as estimate of final project duration and cost (EAC), using either “expected values”, or some other confidence level (say 80%) g. Use risk-based profile as updated expected spend from time-now to project completion
<p>3. Evaluating risk management process effectiveness</p> <ul style="list-style-type: none"> a. Determine threshold values for CPI and SPI to trigger corrective action in risk process (or use default values of 0.75, 0.90 and 1.25) b. Calculate earned value performance indices (CPI and SPI), plot trends and compare with thresholds c. Consider modifications to risk process if CPI and/or SPI cross thresholds, enhancing the process to tackle opportunities more effectively if CPI and/or SPI are high, or refocusing the process on threat reduction if they are low d. Take appropriate action either to exploit opportunities (high CPI/SPI), address threats (low CPI/SPI), spend contingency to recover time (high CPI/low SPI), or spend time to reduce cost drivers (high SPI/low CPI) e. Consider need to review initial baseline, project plan or scope if CPI and/or SPI persistently have unusually high or low values

Figure 9. Summary of steps to integrate EVM and RM (from Hillson 2004, 5)

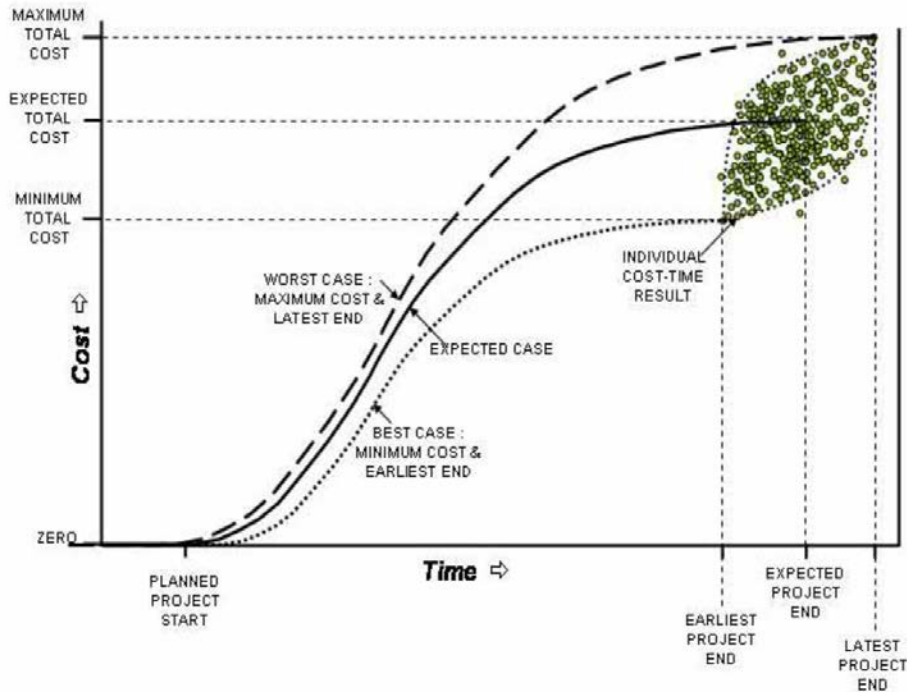


Figure 10. Risk-based cumulative spend profile (from Hillson 2004, 6)

10. Accuracy of Estimation Values

Early program estimates are made to establish a timeline for the program. These estimates are then utilized to establish costs for the program as well as having impact on other factors such as manning. While many programs build in some extra time into a program's timeline to account for delays, it is often not enough and results in the project being behind schedule and over budget.

A modeling experiment was conducted to explore the impact different factors of a program had on the probability that the program would be able to be completed on schedule. The factors that were explored are:

1. **Number of Potential Critical Paths:** Each project eventually has only one true critical path that encompasses all the tasks that determine the length of the project's timeline, but many projects at their onset have more than one path that might become the critical path if the task durations are modeled with probability distributions. The more complex a project network is, the more likely it is to have multiple potential critical paths.
2. **Length of the Critical Path:** Critical paths may be as short as a single task in a very simple system but may contain hundreds of individual tasks.

3. **Task Interdependence:** A task may be performed in a single critical path or may need to be completed for many potential critical paths to be completed. A project was considered to have a high interdependence if tasks were shared by multiple critical paths and low if the critical paths separate from each other with no shared tasks.
4. **Risk Uncertainty:** Depending on the characteristics of a project, a risk distribution must be selected to reflect this risk to aid in estimating the length of a project's schedule. A project with many tasks or elements utilizing new technology will more than likely have a higher level of uncertainty and be reflected by a corresponding risk distribution.

a. Design of Experiments

To analyze the impacts that the four factors had on the probability that a project would be completed on-time the following methodology was utilized:

1. Three different numbers of near critical paths were examined: two, four, and eight.
2. Three different lengths of critical paths were examined: 5, 10, and 20.
3. An interdependence of low, medium, and high was examined. A system with a low interdependence had no shared tasks between potential critical paths as shown in Figure 11. A system with a medium interdependence had some shared tasks between the potential critical paths as shown in Figure 12. A system with high interdependence had numerous shared tasks between the critical paths as shown in Figure 13.

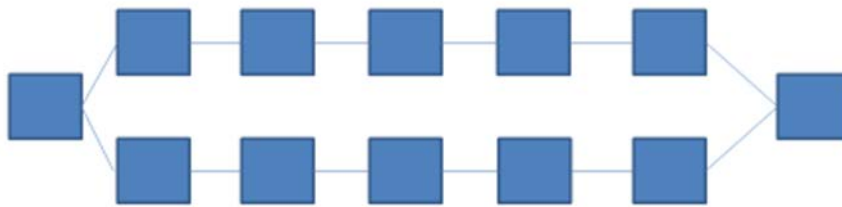


Figure 11. Five Task Elements, Two Critical Paths and Low Interdependence

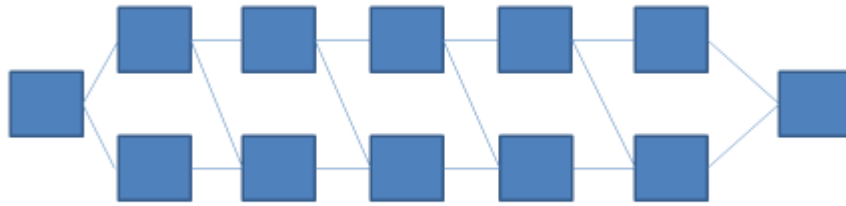


Figure 12. Five Task Elements, Two Critical Paths and Medium Interdependence

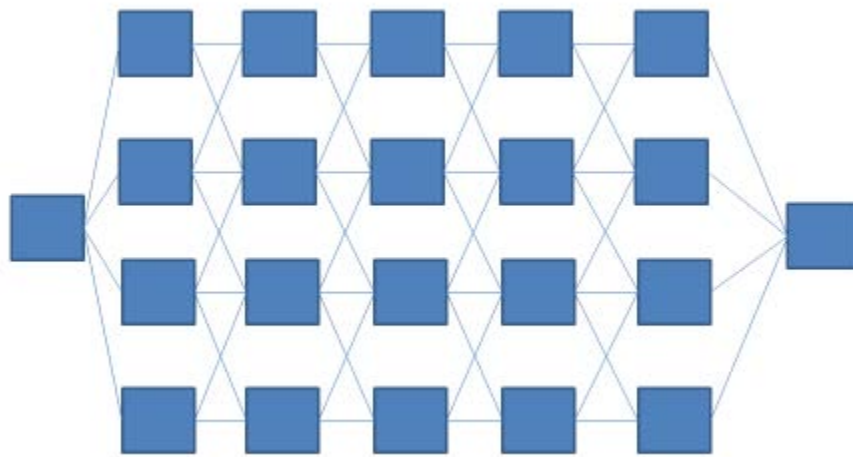


Figure 13. Five Task Elements, Four Critical Paths, and High Interdependence

4. For Risk Uncertainty, the default bounds for subjective distributions from the *U.S. Air Force Cost Risk and Uncertainty Analysis Handbook (2007)* were utilized. Each test case was assumed to have a consistent risk uncertainty for each or its tasks. Five different distributions were utilized. The distributions utilized and their values are shown in Table 2.

Table 2. Default Bounds for Subjective Distributions (from *Air Force Cost Risk and Uncertainty* 2007, 16)

Distribution	Point Estimate Interpretation	Point Estimate and Probability	15%	Mean	85%
Triangle Low	Mode	1.0 (50%)	0.834	1.000	1.166
Triangle Low Right	Mode	1.0 (25%)	0.959	1.122	1.305
Triangle Med Right	Mode	1.0 (25%)	0.931	1.204	1.508
Triangle High Right	Mode	1.0 (25%)	0.903	1.286	1.711
Triangle High Right	Mode	1.0 (25%)	0.876	1.367	1.914

To generate a list of test runs to explore the variables, JMP Pro was utilized for the design of experiment. A screening design with four factors was entered. The factors described above were utilized. This process ensured that each factor was adequately explored to see its effect on a project’s schedule. Table 3 shows the different test runs generated by JMP Pro and analyzed by the model.

Table 3. JMP Pro Design of Experiment Run Definitions

Run Number	Length of Critical Path	Number of Critical Paths	Task Interdependence	Triangle Risk Distribution
1	10	2	Low	Low
2	10	8	Med	Low Rt
3	5	2	Med	Med Rt
4	20	2	Low	High Rt
5	20	2	Med	E High Rt
6	20	8	Med	Low
7	20	4	Low	Low Rt
8	10	4	Low	Med Rt
9	5	4	Med	High Rt
10	10	4	High	E High Rt
11	5	4	High	Low
12	5	2	High	Low Rt
13	20	8	High	Med Rt
14	10	8	High	High Rt
15	5	8	Low	E High Rt

The model was implemented in MS Excel. Each task was assigned the same expected time within a test run. This time was then randomized utilizing a random number generated by MS Excel and adjusted with the risk distribution associated with the test run. Once each task had an adjusted length of time to represent actual run time, the tasks on each critical path were summed to get project duration for each critical path. The longest project duration (i.e., highest total for critical paths) was then recorded for the final project duration. This modeling experiment was repeated 1,000 times for each test run. The test run results are shown in Table 4 and Table 5. The “min sked” column, or minimum schedule, is the lowest possible project duration on the critical path based on the task lengths and the low end of the chosen risk distribution. The “ML sked” column represents the most likely schedule for the project’s critical path based on the task lengths and the chosen risk distribution. The “max sked” column represents the maximum possible schedule for the critical path based on the estimated lengths of the tasks and the chosen risk distribution. The “mean sked” column provides the mathematical mean for the critical path based on the mean tasks times. The “median sked” column shows the mathematical median for the project’s critical path based on the sum of the task lengths and the risk distribution. The “MC Median” column is the median of the 1,000 runs done for each of the 15 different runs.

Table 4. Monte Carlo Simulation Results (Time-Schedule)

test case #	Number of Critical paths	Length of Critical Path	Interdependence	Risk Distribution	Time Schedule					
					min sked	ML sked	max sked	mean sked	median sked	MC Median
1	2	10	Low	Triangle Low	18.680	20.000	23.320	20.667	20.545	20.855
2	8	10	Med	Triangle Low Right	19.180	22.440	26.100	22.573	22.541	23.186
3	2	5	Med	Triangle Med Right	18.620	24.080	30.160	24.287	24.237	24.861
4	2	20	Low	Triangle High Right	18.060	25.720	34.220	26.000	25.933	26.395
5	2	20	Med	Triangle XHigh Right	17.520	27.340	38.280	27.713	27.624	28.243
6	8	20	Med	Triangle Low	18.680	20.000	23.320	20.667	20.545	20.976
7	4	20	Low	Triangle Low Right	19.180	22.440	26.100	22.573	22.541	22.896
8	4	10	Low	Triangle Med Right	18.620	24.080	30.160	24.287	24.237	25.055
9	4	5	Med	Triangle High Right	18.060	25.720	34.220	26.000	25.933	27.491
10	4	10	High	Triangle XHigh Right	17.520	27.340	38.280	27.713	27.624	29.107
11	4	5	High	Triangle Low	18.680	20.000	23.320	20.667	20.545	21.116
12	2	5	High	Triangle Low Right	19.180	22.440	26.100	22.573	22.541	22.926
13	8	20	High	Triangle Med Right	18.620	24.080	30.160	24.287	24.237	25.025
14	8	10	High	Triangle High Right	18.060	25.720	34.220	26.000	25.933	27.468
15	8	5	Low	Triangle XHigh Right	17.520	27.340	38.280	27.713	27.624	30.340

Table 5. Monte Carlo Simulation Results (Probability of Completion)

test case #	Number of Critical paths	Length of Critical Path	Interdependence	Risk Distribution	Probability of Completion on time or early					
					min sked	ML sked	max sked	mean sked	median sked	MC Median
1	2	10	Low	Triangle Low	0.0%	0.0%	100.0%	25.4%	12.6%	50.0%
2	8	10	Med	Triangle Low Right	0.0%	0.1%	100.0%	0.5%	0.3%	50.0%
3	2	5	Med	Triangle Med Right	0.0%	16.0%	100.0%	24.9%	22.5%	50.0%
4	2	20	Low	Triangle High Right	0.0%	13.0%	100.0%	25.3%	22.3%	50.0%
5	2	20	Med	Triangle XHigh Right	0.0%	12.6%	100.0%	24.2%	20.1%	50.0%
6	8	20	Med	Triangle Low	0.0%	0.0%	100.0%	0.5%	0.0%	50.0%
7	4	20	Low	Triangle Low Right	0.0%	1.7%	100.0%	5.8%	4.2%	50.0%
8	4	10	Low	Triangle Med Right	0.0%	2.1%	100.0%	6.2%	4.9%	50.0%
9	4	5	Med	Triangle High Right	0.0%	3.7%	100.0%	5.1%	4.5%	50.0%
10	4	10	High	Triangle XHigh Right	0.0%	2.5%	100.0%	6.6%	5.0%	50.0%
11	4	5	High	Triangle Low	0.0%	0.0%	100.0%	5.1%	2.2%	50.0%
12	2	5	High	Triangle Low Right	0.0%	17.9%	100.0%	25.4%	24.0%	50.0%
13	8	20	High	Triangle Med Right	0.0%	0.0%	100.0%	0.4%	0.3%	50.0%
14	8	10	High	Triangle High Right	0.0%	0.0%	100.0%	0.4%	0.2%	50.0%
15	8	5	Low	Triangle XHigh Right	0.0%	0.1%	100.0%	0.2%	0.2%	50.0%

The run results were then normalized by dividing the results by the Monte Carlo Median (MC Median). These normalized results are shown in Table 6.

Table 6. Normalized Time-Schedule Monte Carlo Simulation Results

Run #	Normalized Time Schedule Percentages (Factor/MC Median)					
	min sked	ML sked	max sked	mean sked	median sked	MC Median
1	90%	96%	112%	99%	99%	100%
2	83%	97%	113%	97%	97%	100%
3	75%	97%	121%	98%	97%	100%
4	68%	97%	130%	99%	98%	100%
5	62%	97%	136%	98%	98%	100%
6	89%	95%	111%	99%	98%	100%
7	84%	98%	114%	99%	98%	100%
8	74%	96%	120%	97%	97%	100%
9	66%	94%	124%	95%	94%	100%
10	60%	94%	132%	95%	95%	100%
11	88%	95%	110%	98%	97%	100%
12	84%	98%	114%	98%	98%	100%
13	74%	96%	121%	97%	97%	100%
14	66%	94%	125%	95%	94%	100%
15	58%	90%	126%	91%	91%	100%

The factors were then compared to the MC Median to create data that was more easily analyzed. The results are shown in Table 7.

Table 7. Probability of Completion Compared to Monte Carlo Median Results

Probability of Completion early or time percentage difference from Median (Factor % - MC Median %)						
Run #	min sked	ML sked	max sked	mean sked	median sked	MC Median
1	-50.0%	-50.0%	50.0%	-24.6%	-37.4%	0.0%
2	-50.0%	-49.9%	50.0%	-49.5%	-49.7%	0.0%
3	-50.0%	-34.0%	50.0%	-25.1%	-27.5%	0.0%
4	-50.0%	-37.0%	50.0%	-24.7%	-27.7%	0.0%
5	-50.0%	-37.4%	50.0%	-25.8%	-29.9%	0.0%
6	-50.0%	-50.0%	50.0%	-49.5%	-50.0%	0.0%
7	-50.0%	-48.3%	50.0%	-44.2%	-45.8%	0.0%
8	-50.0%	-47.9%	50.0%	-43.8%	-45.1%	0.0%
9	-50.0%	-46.3%	50.0%	-44.9%	-45.5%	0.0%
10	-50.0%	-47.5%	50.0%	-43.4%	-45.0%	0.0%
11	-50.0%	-50.0%	50.0%	-44.9%	-47.8%	0.0%
12	-50.0%	-32.1%	50.0%	-24.6%	-26.0%	0.0%
13	-50.0%	-50.0%	50.0%	-49.6%	-49.7%	0.0%
14	-50.0%	-50.0%	50.0%	-49.6%	-49.8%	0.0%
15	-50.0%	-49.9%	50.0%	-49.8%	-49.8%	0.0%

b. Result Analysis

The results of the modeling and simulation (M&S) analysis performed using the DoE described in the previous section were normalized using the MC Median as the reference value and plotted in Figure 14. The horizontal axis of the plot represents each case that was studied and they are referenced in the following format: “Number of Critical Paths X Length of Critical Paths X Interdependence Level.” The lower and upper vertical lines of the boxplots represent the 25% and 75% quartiles respectively. Since the MC Median is used as the reference point for the normalization procedure, it will be equivalent to the 100% accuracy on the schedule estimate. This behavior is not representative of a real life situation since experience dictates that every estimate will inherently produce estimation errors because of multiple factors that can either directly or indirectly impact the outcome of a project. Factors such as: geopolitical stability of different regions in the world, congressional budget procedures, material shortages

among others, are outside the scope of this study. However, they can be indirectly addressed by considering the risk of the different activities in a project through the inclusion of cost and duration uncertainties in the estimation procedures. For the context of this study, it is assumed that using the MC Median will result in more accurate estimates. All the other estimation values are compared to the MC Median.

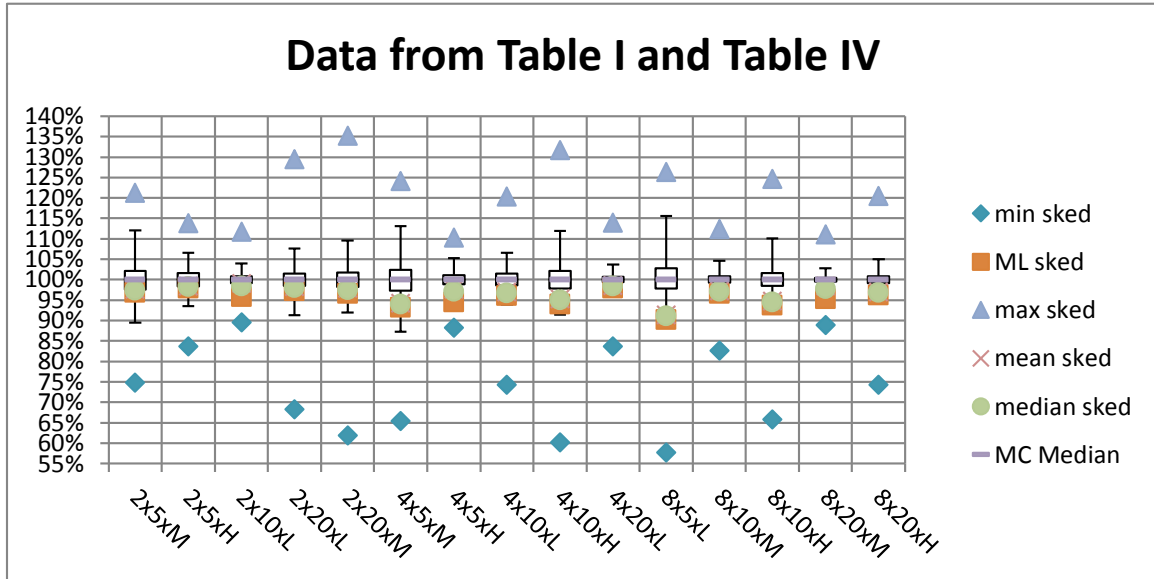


Figure 14. Boxplot of Normalized Results for each Test Case

As it can be observed in Figure 14 regardless of the different characteristics of the project under analysis using the minimum and maximum duration estimates can result in project schedules that over-estimate the duration of the project by roughly 10% to 35% or that under-estimate the duration of the project by roughly 15% to 45%. In many occasions, the results of this estimation processes are used to set the contractual terms that the contracted organization needs to meet, thus setting them up for failure with a project plan that either has too much slack time or that is instantly behind schedule during the execution phase with little or no chance to catch up to the project plan. Using most-likely (ML) estimates, mean estimates and median estimates resulted in project durations that were within 10% of the MC Median. It can be observed in Figure 14 that as the number of critical paths in a project increases, the variability between the MC Median and the other factors used to estimate tasks' duration also increases.

B. STAKEHOLDER ANALYSIS

Several experienced stakeholders involved with NAVAIR project management were interviewed to obtain current project planning and execution process information. Interviewed stakeholders had varying roles in the project management process and worked on several different projects. The project management process varied significantly by project budget size and stakeholder responsibilities and expertise varied by role. Stakeholder roles included, but were not limited to, a project manager for small projects (<\$20M), project manager for large projects (\geq \$20M), Program Management Authority (PMA) Acquisition Product Engineering, and several other key supportive roles such as an EV and scheduling Process Champion.

Small projects under \$20M total budget have relatively little project management oversight aside from the designated project manager's cognizance. There are no EV requirements for small projects and initial rough order of magnitude (ROM) cost estimates are required but in varying document format types which can be problematic and tedious to use for both project planning and execution tracking phases of the project life-cycle. Tools used in the planning phases are limited to what is available to project managers and expensive software for planning is generally not available. Software cost is a significant concern in the confines of federal government budgets. Recently a labor management tool, Workforce Management System (WMS), was planned to be implemented throughout NAVAIR as a cost management tool for some small project managers but is ineffective for real time cost tracking since actual data charges are significantly delayed per interviewee dialogue. Additionally, WMS has no schedule tracking capability. For scheduling, MS Project has been used, but software availability is sporadic due to funding, though it is the most commonly used project scheduling tool at NAVAIR. Uncertainty in cost and schedule management is acknowledged but not rigorously determined. A standing management reserve is held to mitigate cost estimation error in planning. A management reserve is funding that is set aside during the project planning phases for unintended initial underestimates and is usually around 10% of the overall project budget. High schedule risk is accounted for by marginally increasing the slack time into high risk task durations. Although, program managers try to minimize

adding slack time and reliance on management reserves to provide as realistic initial cost and schedule project estimate as possible.

Large projects above \$20M in total budget have substantially more oversight than small projects. These large projects facilitate an independent EV team that assesses project progress outside of their integrated project team (IPT). This independent NAVAIR EV team conducts three different types of assessments on large projects: Integrated Baseline Review (IBR), Schedule Risk Assessment (SRA), and EAC. IBR is only conducted at the start of planning phases or after major engineering modifications are requested by the contractor of the project. SRA is a highly involved project status review and assessment that shows whether projects are progressing as initially planned or if they are tending towards cost and schedule overruns. Officials with project authority use SRA results to make project go/no-go decisions. SRAs occur approximately biannually because they are labor and time intensive and take about six weeks to compile the package. The long duration can be problematic because the actual SRA meeting and review occurs weeks after project issues have been either overcome or worsened. It does not provide a real time project status to project officials. Interviewed stakeholders provided process information for large projects that generally facilitate use of a single prime contractor. Three of the six weeks total SRA package preparation time are needed for this prime contractor to assemble, review, and provide their final cost and schedule data which usually originates from multiple sub-contractors. The data is provided to the project manager and IPT members who will jointly review data. Data includes up-to-date, thorough, review of three point cost and schedule estimates. Three additional weeks are then needed for the independent EV team to process the data and provide their in-depth uncertainty analysis. The reviewed three point data, though slow to obtain, allows the EV team to assess project uncertainty for management officials to make better decisions. Provided risk and estimate data are used to generate histogram plots and cumulative probability distributions or S-curves. EAC is the final project estimate which reviews overall project status at completion.

Though roles and processes differed from interviewee to interviewee, management process commonalities existed between all stakeholders. One commonality

was that all planning phase cost estimating began by establishing an overall project cost ROM. The more time and effort that was expended on a ROM usually resulted in increased accuracy compared to final outcome. For both small and large projects SME's knowledge and experience was facilitated along with limited historical project data to establish ROM estimates. Planning phase schedule is also developed using SME input. Another small and large project similarity is that real time data was difficult to obtain and manipulate to indicate project status. The data used for determining project status needed to be precise in order to properly evaluate project progress. Collecting accurate data from multiple sources took substantial time. Thus, project assessments could not be made in real time and lagged by at least two weeks.

For large projects, multiple software tools exist and are currently used for planning and execution project estimating, scheduling, and managing. Large project tool choice depends on required reporting detail level which is usually driven by events such as IBR, SRA, EAC, or less detailed rhythmic monthly business project status reporting. For small projects, both cost and schedule estimates, are limited to MS Office Suite and the capabilities of respective small project managers. MS Project, Deltek OPP, and Oracle Primavera are software used primarily for large project scheduling during less frequent SRA events. MS Project is advantageous due to its popularity. OPP and Primavera are more costly and less common than MS Project but include additional report features like built-in MC reporting capability. Large project costing software currently used includes Full Monte, a MS Project add-in and Palisade @Risk, a MS Excel add-in. Monthly reporting tools for large projects are less detailed due to required frequent reporting and include Deltek Winsight and NAVAIR developed Performadex web-based portal. Tools facilitated by NAVAIR and their functions are compared in Table 8.

Table 8. NAVAIR Project Management Tool Functions

	Monte Carlo Analysis	Sensitivity Analysis	Histograms	S-curves	Uncertainty Considered	Website
Cost Estimation Software						
Crystal Ball® by Oracle (Utilized by Excel)	yes	yes	yes	Only with Crystal Decision Optimizer Application	yes	http://www.oracle.com/us/products/applications/crystalball/overview/in
@Risk by Palisade (Utilized by MS Excel)	yes	yes	yes	yes	yes	http://www.palisade.com/risk/
Schedule Risk Software						
Open Plan Professional (OPP) by Deltek (utilized with MS Project)	yes (Requires Deltek Cobra to integrate cost)	no	yes	no	yes	http://www.pinnaclemanagement.com/deltek-open-plan/303 http://www.deltek.com/products/ppm/schedule/openplan
Full Monte by Barbecana (Utilized with MS Project or Primavera P6 Enterprise Portfolio Management)	yes	yes	yes	yes	yes	https://www.barbecana.com/
Integrated Risk Software						
Primavera Risk Analysis by Oracle (Utilized with P6 Enterprise Portfolio Management)	yes	yes	yes	yes	yes	http://www.oracle.com/us/products/applications/primavera/risk-analysis
Open Plan Professional and Cobra by Deltek (Utilized with MS Project and Excell)	no	yes	yes	no	no	http://www.deltek.com/products/ppm/cost/cobra/resources
@Risk for Project Mgmt by Palisade ((Utilized with MS Project and Excell)	yes	yes	yes	yes	yes	http://www.palisade.com/projectriskmanagement

Stakeholder interviews identified several project management process and tool gaps for both small and large projects. Many software applications are currently used for project planning and management however the research indicated that no existing tool can incorporate both task-based scheduling with EV cost management. Also small projects do not consider uncertainty because software is not available and no requirements exist to conduct analysis. Multiple stakeholders expressed interest in software that was easy to use and would facilitate easier accessibility since software is difficult to obtain due to cost and is often not compatible with existing software and government computer network information technology (IT). Stakeholder identified gaps were used to generate requirements which will be discussed further in Chapter III.

C. OPERATIONAL-BASED SCENARIO

The main purpose for developing the generic operational-based scenario was to provide a framework to help validate the prototype software application. Using the notes from the stakeholders' interviews, various characteristics and instances were identified as common among the stakeholders. The study used that information to develop generic project planning and execution processes that represent the operational intent of the stakeholders. Figure 15 depicts the overall operational view of the developed prototype software application including the different intended users. The users identified in the different phases of the operational-based scenario represent the different roles involved in the planning and management of engineering projects. These roles can be executed by a single person or by different personnel depending on the organization where the prototype software application is implemented and the complexity of the project under analysis. For simplicity of this study, the scenario only makes reference to the higher level user roles and does not take into considerations the delegation of responsibilities typical in many organizations. For example, the Project Manager (PM), depicted in Figure 15, might delegate the tasks of developing accurate project plans to a Project Planner, who would generate a baseline plan and deliver it to the PM for approval. Also, the Lead Engineer (LE) might need to consult a group of SMEs in order to accurately estimate the duration of a particular activity of the project under analysis. Those

interactions are not within the scope of this study and the operational-based scenario will only address the task for which the PM and LE are responsible.

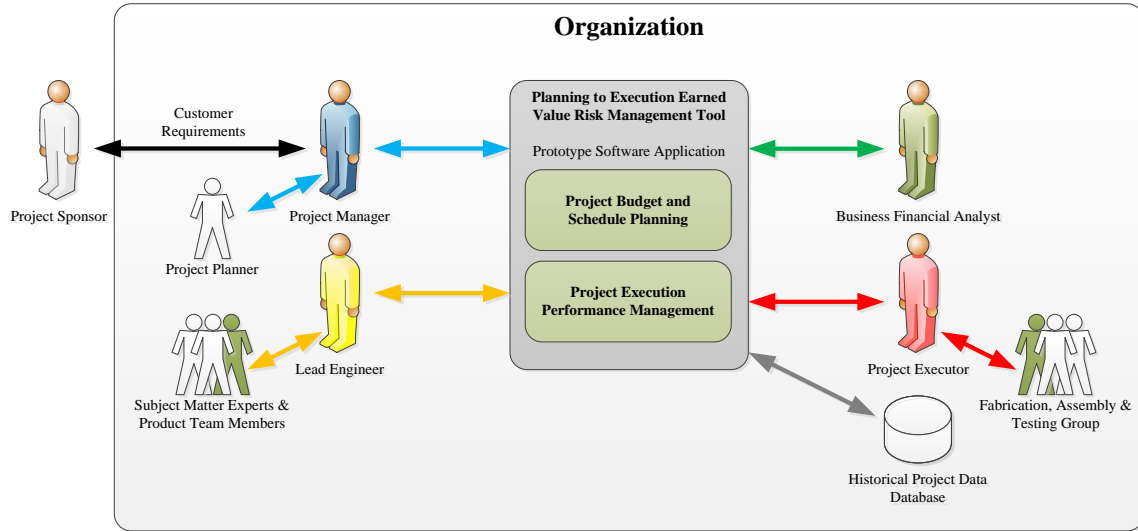


Figure 15. Prototype Software Application Operational View

The operational-based scenario was decomposed into project planning phase and project management phase in order to fully examine the different interactions of the prototype software application and the stakeholders. Figure 16 shows how the project planning phase and the project management phase were further decomposed into activities that detail the exchange between the different users and the prototype software application.

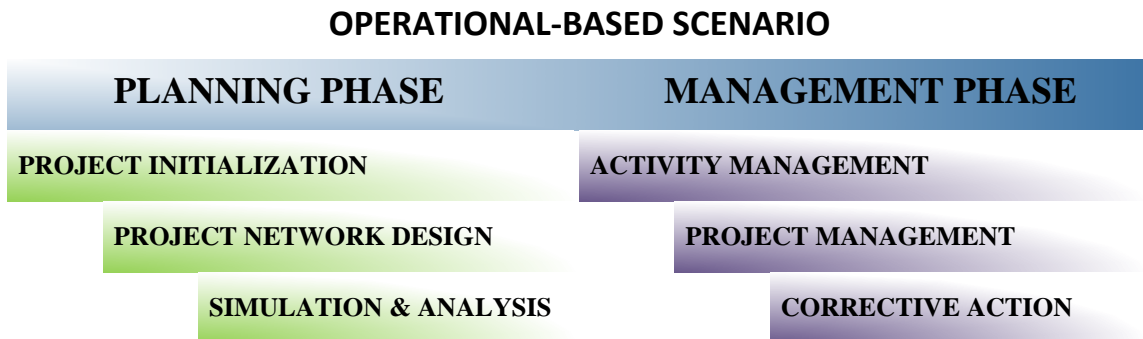


Figure 16. Operational-Based Scenario Decomposition

1. Project Planning Phase

Every organization has its own method of developing cost and duration estimates and does not necessarily adhere to the planning process that is explained in this section. Table 9 lists the steps involved in the planning process used as reference to create the operational decomposition of the project planning phase into the following activities: Project Initialization, Network Design, Simulation and Analysis.

Table 9. Generic Project Planning Process

Step	Process Description
1	A sponsor contacts the PM to request a baseline project plan be created, which includes an accurate estimate of budget and schedule for a specific project.
2	The PM uses the prototype software application to generate a baseline project plan that accounts for cost and duration uncertainties of the different activities of the project under analysis.
3	Once the PM enters the project description data in the prototype software application, the tool generates a notification to the personnel identified (i.e., project executor (PE), business financial analyst (BFA), and LE) as resources for the requested project plan.
4	The LE that was identified in step three (3) establishes the activities required to meet the requirements of the sponsor in the form of a project network diagram. He/she also deliver an estimate of the duration for each of the activities that were previously identified.
5	Once the LE finishes his/her assessment of the required activities, the BFA is notified of his/her involvement in the requested project. The BFA will perform a cost estimate for each activity in the project.
6	With the generated project network diagram and the estimate of duration and cost for each activity in the project, the PM proceeds to run the simulation analysis of the prototype software application in order to develop a baseline report that includes a projection of the budget and schedule for the project under analysis.
7	Once the prototype is done running the analysis and generating the baseline report, it will notify the personnel involved (i.e., PM, LE, BFA) that a report has been generated and is ready for review.
8	The PM gathers the feedback provided by the LE and the BFA in order to make any necessary adjustments to the project plan.
9	Finally, the PM delivers a baseline project plan to the sponsor, which if accepted will become the basis of any type of contract for engineering services between the sponsor and the PM's organization.

Figure 17 displays the major components involved in the planning process and the sequence of the required interactions in order to deliver a baseline project plan to the sponsor of the project.

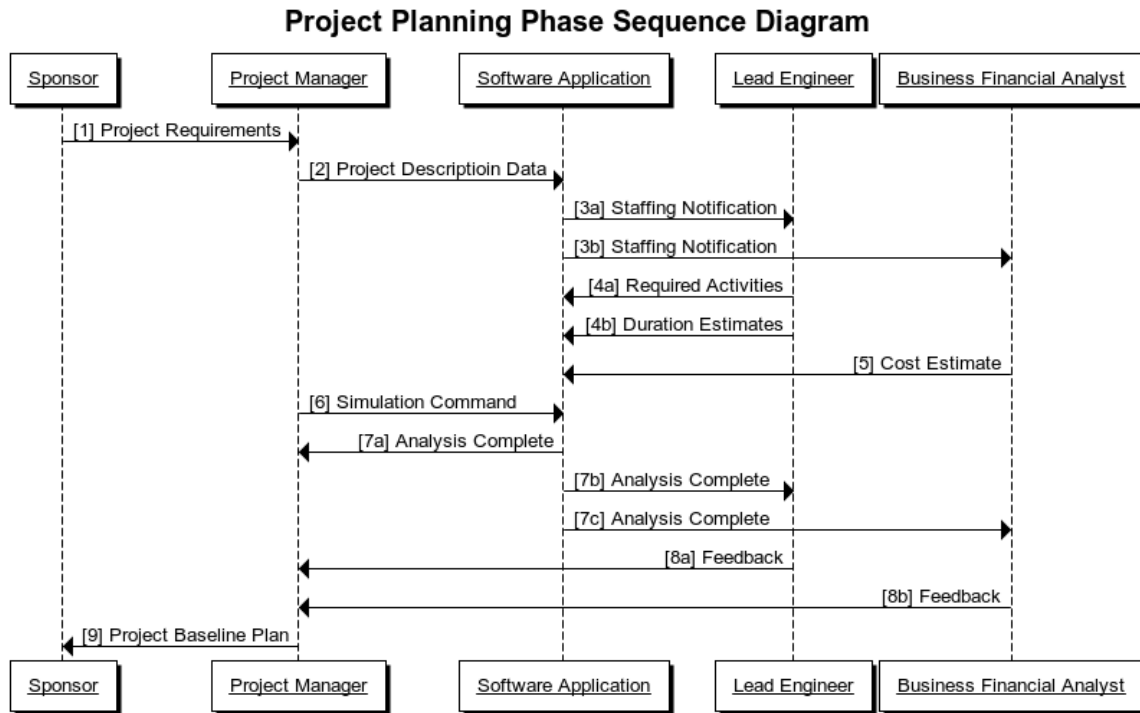


Figure 17. Operational Sequence Diagram for Planning Phase

It is likely that the identified users of the prototype software application will provide more inputs than those shown in Table 9 and Figure 17. For example, the LE will identify the activities that are required for the project under analysis and might be asked by the BFA to provide the cost basis for the raw material to be used in such activities.

a. Project Initialization Activity

The main actor of this activity is the PM who needs to plan and manage the engineering project according to the requirements provided by a sponsor. The other actor involved in this activity is the prototype software application, which receives the inputs from the PM in order to establish a project identification file that can be retrieved later on for further refining. The PM has the option of importing a network diagram previously

built using MS Project and save it as part of the project information that is established in this activity. The trigger of this activity will be the request of the sponsor to the PM in the form of requirements and project scope (i.e., desired cost and schedule). Prior to the beginning to this activity, it is expected that the prototype application will be up and running in the PM’s computer. At the completion of the project initialization activity, the PM will have a project file that is used as an input to other activities in the project planning and management phases. Table 10 lists the normal and alternate flows for the project initialization activity.

Table 10. Decomposition of the Project Initialization Activity

Source	Step	Action
Project Manager	1	Start software application
Software	2	Display “Project Description” tab
Software	3	Request “Project ID Number”
Project Manager	4	Enter “Project ID Number”
Project Software	5	Request “Project Name”
Project Manager	6	Enter “Project Name”
Software	7	Request “Project Manager Name”
Project Manager	8	Enter “Project Manager Name”
Software	9	Request “Lead Engineer Name”
Project Manager	10	Enter “Lead Engineer Name”
Software	11	Request “Financial Analyst Name”
Project Manager	12	Enter “Financial Analyst Name”
Software	13	Request “Sponsor Agency Name”
Project Manager	14	Enter “Sponsor Agency Name”
Software	15	Request “Sponsor Agency POC”
Project Manager	16	Enter “Sponsor Agency POC”
Software	17	Request “Project Start Date”
Project Manager	18	Enter “Project Start Date”
Software	19	Request “Project Cost Objective”
Project Manager	20	Enter “Project Cost Objective”
Software	21	Request “Project Risk Tolerance”
Project Manager	22	Enter “Project Risk Tolerance”
Software	23	Request “Project Description”
Project Manager	24	Enter “Project Description”
Project Manager	25.1*	Press “Import Network Diagram”
Software	25.2*	Display browser window
Project Manager	25.3*	Select MS Project File
Software	25.4*	Import network structure

Source	Step	Action
Software	26	Enables “Save Project File”
Project Manager	27	Click “Save Project File”
Software	28	Display popup window “Project File Saved”

*Denotes optional path.

b. Project Network Design Activity

The main actors of this activity are the LE and the BFA. The LE establishes the required activities or tasks of the project and in consultation of SMEs estimates the probability distribution parameters for the duration of each activity. The identified activities and their estimated duration are captured by the LE in a project network diagram that is used by the BFA to develop estimates of the fixed cost for each activity. Once this data is collected, the PM reviews the information provided by the LE and the BFA through the software application. If such information is complete and deemed satisfactorily by the PM, then he/she proceeds to perform the simulation and analysis described in the next section. The LE or the PM can also assign a PE, who is in charge of updating the actual cost and schedule once the project is in the execution or management phase. The project network design activity is triggered by the completion of the project initialization activity performed by the PM. Once the PM establishes the descriptive data of the project under analysis, there are two possible ways that the information could be entered into the prototype software application. (1) The project network data could have been imported from a prebuild MS Project file by the PM during project initialization or (2) it needs to be built by the LE in coordination with the PM and SMEs. Prior to the beginning of this activity the prototype software is up and running in the LE’s computer and the project description data has been entered. At the completion of this activity the PM will have a fully populated project network diagram in which all activities have been identified with a specific name and their associated cost and duration estimates. At the end of this activity the prototype software application has all the inputs required to run a MC analysis that produces the project budget and schedule forecasts. Table 11 lists the normal and alternate flows for the network design activity.

Table 11. Decomposition of the Project Network Design Activity

Source	Step	Action
Lead Engineer	1	Open the project network tab of the prototype software application
Software	2.1.1	Display the network diagram of the imported MS Project file
Software	2.1.2	Highlight the activities in the network diagram that do not have all cost and duration data.
Software	2.2.1	Display the empty white board and table where the LE can enter the information to build the network diagram directly from the prototype software application.
Lead Engineer	3	Click on a specific activity to enter its corresponding data
Software	4	Display activity information window
Software	5	Display auto-populated Activity Number
Software	6	Request "Activity Name"
Lead Engineer	7	Enter "Activity Name"
Software	8	Request "Duration Probability Distribution Parameters"
Lead Engineer	9	Enter "Duration Probability Distribution Parameters"
Software	10	Request "Fixed Cost Probability Distribution Parameters"
Financial Analyst	11	Enter "Fixed Cost Probability Distribution Parameters"
Software	12	Request "Activity Variable Cost Rate"
Financial Analyst	13	Enter "Activity Variable Cost Rate"
Software	14.1*	Request "Activity Executor Name"
Lead Engineer	14.2*	Enter "Activity Executor Name"
Lead Engineer / Financial Analyst	15	Click "Save" to store the information of the activity
Software	16	Change the color of the specific activity based on which information was entered in the previous steps to denote that it has all the information that it needs to execute the Monte Carlo Analysis.
	17	Repeat steps 3 through 20 until every activity has been identified and their data has been stored.

*Denotes optional path.

c. Simulation and Analysis Activity

The main actors of this activity are the PM, LE and the BFA. The PM will take the information entered by the LE and the BFA and run a MC Simulation in order to generate a budget and schedule that accounts for cost and duration uncertainties of the

different activities or tasks in the project. Once the prototype software application is finished running the MC Simulation, the PM will consult the LE and the BFA to review the generated project schedule and budget forecast. The simulation and analysis activity is triggered once the LE and the BFA are finished entering all the required data for each activity in the project network diagram and the PM approves it. Before the PM can run the simulation, every activity in the project network diagram has to be fully populated, and there should not be any highlighted nodes indicating missing information. At the completion of this activity the PM will have a baseline project plan, based on the inputs from the LE and BFA which can be used in contractual negotiations with the sponsor of the project. If the generated baseline project plan is approved by the sponsor and the PM's organization, then the plan becomes an input to the project execution phase managed from the prototype software application. Table 12 lists the normal and alternate flows for the simulation and analysis activity.

Table 12. Decomposition of the Simulation and Analysis Activity

Source	Step	Action
Project Manager	1	Open the "Simulation Parameter" tab of the prototype software application
Software	2	Request "Simulation Seed Value"
Project Manager	3	Enter "Simulation Seed Value"
Software	4	Request "Simulation Runs"
Project Manager	5	Enter "Simulation Runs"
Project Manager	6.1	Selects to not save simulation log
Project Manager	6.2	Selects to save simulation log
Software	6.2.1	Request directory where the simulation log will be saved
Project Manager	6.2.2	Enter "Simulation Log Directory"
Project Manager	6	Click "Start" button in the simulation control panel
Software	7	Displays the progress of the simulation in a graphical format

Source	Step	Action
Software	8	When the simulation is done, displays the following information: <ol style="list-style-type: none"> 1. The median project duration and cost. 2. Graphical display of the critical paths of the project. 3. Tabulated display of all the activities in the project with the respective simulated median cost and durations. 4. Graphical and tabulated display of the time phase budget and schedule.
Lead Engineer	9	Reviews the duration/schedule data generated by the Monte Carlo Simulation
Financial Analyst	10	Reviews the cost data/budget data generated by the Monte Carlo Simulation
Program Manager	11	Receives feedback from the LE and the Financial Analyst and adjusts the information of the different activities accordingly

2. Project Management Phase

The project management or execution phase uses the generated baseline project plan to capture the actual cost and duration for each activity in the project network. Depending on the organization where the prototype software application is implemented, the person responsible for entering the actual activity data could be the LE or a designated PE. For the purpose of the execution process established for this study, the PE is the lead or supervisor of the Fabrication, Assembly and Testing Group (i.e., technicians, developers, and engineers) that performs the tasks required for each activity. The PE is responsible for capturing the actual data for each activity in the project. The LE is consulted any time there is a deviation of the baseline project plan.

The BFA oversees multiple projects to make sure that no contractual threshold has been violated, and that the expenditures rates are in line with the stipulated baseline project plan. The PM monitors the expenditure rates as well, but he/she will also be in charge of making sure that corrective actions are taken, whenever necessary, in order to maintain the project in line with the generated time-phase budget of the baseline project

plan. The PM constantly monitors the progress of each activity and compares it to the established critical paths of the project to make sure that any management action will result in added value to the project. Table 13 lists the steps involved in the execution process used as reference to create the operational decomposition of the project management or execution phase into the following activities: Project Performance Tracking, Project Management and Corrective Action. Figure 18 displays the major components involved in the project execution process and the sequence of the required interactions in order to manage the project

Table 13. Generic Project Management Process

Step	Process Description
1	The contract is awarded to the PM's organization by the project sponsor.
2	The PM generates a project tracking report using the prototype software application which notifies the personnel involved with the project execution (i.e., PE, BFA, and LE) of their tasking in the project.
3	Once the project starts, the PE enters the actual cost and duration of the activities that are under execution in the prototype software application.
4	The prototype software application updates the cost and schedule trajectory graphics and using that information compares the actual data of the activities under execution to the baseline project plan. Any activity that is running behind-schedule or over-budget is highlighted by the prototype software application. The prototype software application generates a graph that shows the forecast time-phased budget, the actual cost of the work performed up-to-date. It will also display in a tabular format the progress and performance of each activity in terms of the cost and time spent versus its forecasted cost and duration.
5	The PM utilizing the prototype software application generates a project status report that depicts the current progress of the project in an EVM-like format. For example, the project will use the activity's median cost and schedule from the baseline project plan as the BCWS and the actual duration and cost entered by the PE for a specific activity as the BCWP and the ACWP respectively to calculate the cost and schedule variances.
6	When updating the baseline project plan, the PM will have to consult the LE, the BFA and the PE in order to realign the previously generated cost and duration estimates for all the activities in the project. The PM then uses the updated estimates to perform the simulation and analysis process in order to determine if a contract change has to be requested.
7	Once a project is finished, the PM generates a project completion report from the prototype software application that contains project data that can be used by the PM's organization for historical purposes. This project data can be used in future project in order to generate accurate estimates of activities' cost and duration.

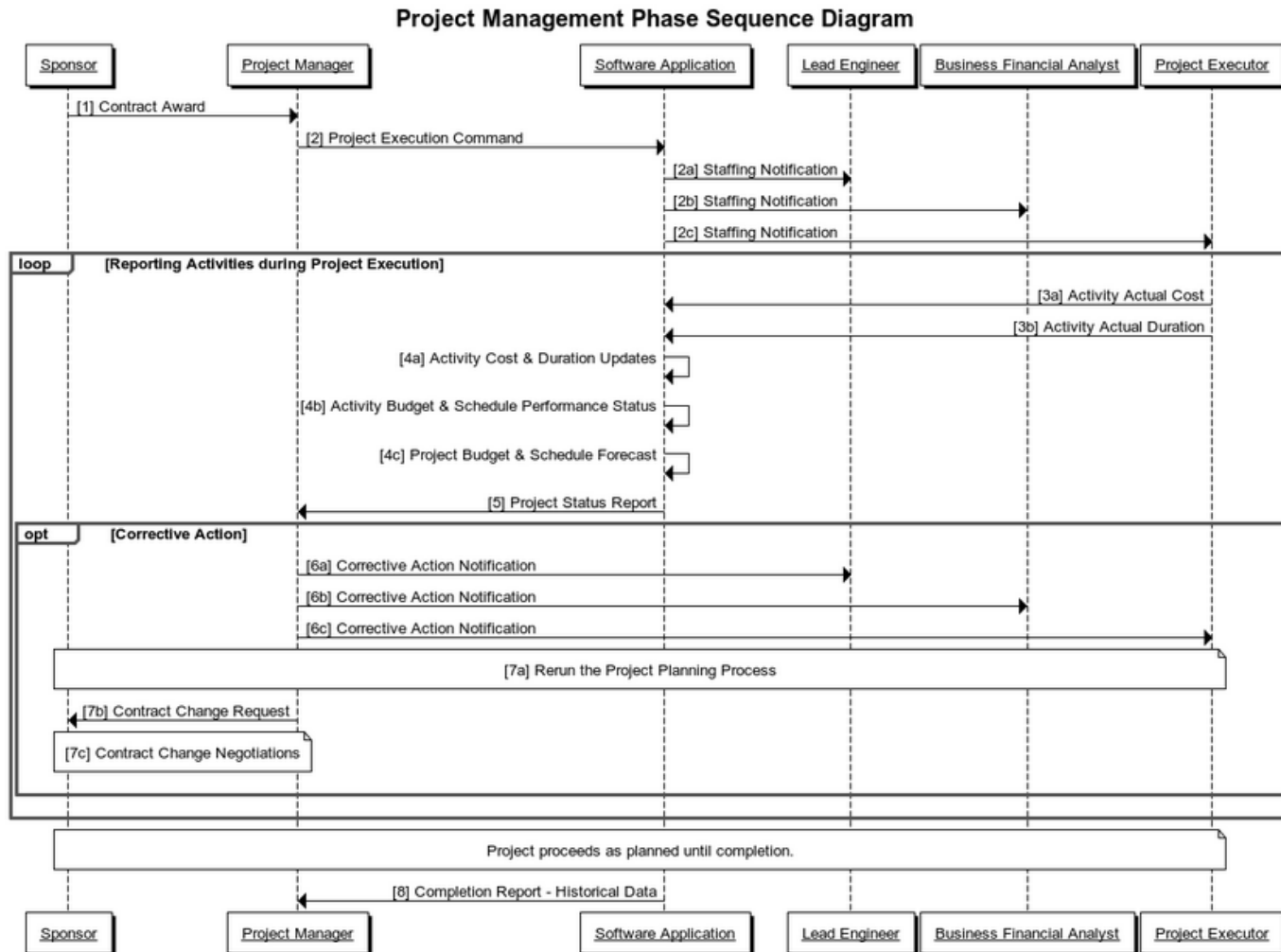


Figure 18. Operational Sequence Diagram for Project Execution Phase

a. Performance Tracking Activity

The main actors of this activity are the PM and the PE. The PM receives a baseline project plan for execution that contains the budget, schedule and staffing information generated during the project planning phase. It is the PM that triggers the project execution command in the prototype software application which generates staffing notifications to the LE, the BFA, and most importantly to the PE. The PE is responsible for the execution of the planned tasks for the project under development. He/she enters the actual cost and duration data for the activity that is being executed into the prototype software application. In the event of discrepancies between the baseline project plan and the execution of the project, the PE enters the actual data for the specific task along with information about the source of the discrepancy. The prototype software application uses that information to update its project tracking information and generate alerts to the PM about any issue or deviation from the baseline project plan.

The project performance tracking activity is triggered by the contract award from the project sponsor to the PM's organization. The PE will be able to start entering cost and duration data for the different tasks in the prototype software application once he/she receives the staffing notification and a copy of the baseline project plan. The project performance tracking activity is repeated throughout the duration of the project under development. After each iteration of this activity is completed, the PM will have a clear picture of the current status of the project and should have sufficient information to make decisions that either add value to the project or correct a negative trend in the project performance. Table 14 lists the normal and alternate flows for the project performance tracking activity.

Table 14. Decomposition of the Project Performance Tracking Activity

Source	Step	Action
Project Manager	1	Open the "Cost and Schedule Trajectory" tab of the prototype software application
Project Manager	2	Click "Enable Project Execution"
Software	3	Send staffing notification to the LE, BFA and PE

Source	Step	Action
		The PE starts executing the activities or tasks that are listed in the “Cost and Schedule Trajectory” of the baseline project plan.
Project Executor	4	Open the “Cost and Schedule Trajectory” tab of the prototype software application
Software	5	Display the tabulated baseline project plan with fields that allow the PE to enter the “Actual Cost,” “Actual Duration” and “Comments” for each activity of the project
Project Executor	6	Enter “Actual Cost” of the activity under execution
Software	7	Compare the actual cost entered by the PE to the baseline plan cost for that activity and highlight the field to indicate whether or not the activity is over-budget
Project Executor	8	Enter “Actual Duration” of the activity under execution
Software	9	Compare the actual duration entered by the PE to the baseline plan schedule for that activity and highlight the field to indicate whether or not the activity is behind-schedule
Software	10.1*	If the activity is over-budget and/or behind-schedule, highlight the “Comments” field for that activity
Project Executor	10.2*	Enter the reasons for why the activity is currently underperforming when compared to the baseline project plan (e.g., logistics delay, inclement weather, work incidents)
Project Executor	11.1.1	Click “Update Cost/Schedule Trajectory”
Software	11.1.2	Update graphical display of the cumulative forecast of cost and schedule for the overall project taking in consideration the actual data for the activities that have been executed at the moment
Software	11.1.3	Highlight those activities that are underperforming in the Project Network Diagram
Software	11.1.4	Send notification to the PM of the new changes added by the PE to the prototype software application
Project Executor	11.2.1	Click “Cancel Changes”
Software	11.2.2	Display notification about the deletion of unsaved changes
Software	11.2.3	If the PE accepts the notification of step 11.2.2, then clear all the information that was entered since the last time the report was saved.

*Denotes optional path.

b. Project Management and Corrective Action Activity

The main actors of this activity are the PM and the BFA. The PM receives the notification that a change has been made in the instance of the prototype software application that is tracking the performance of the project under development. Then, the PM looks for any indication that shows poor performance or that might indicate where improvement can be made in order to address the activities that have been identified as high risk in the critical path analysis performed in the project planning phase as part of the simulation and analysis activity. The BFA will monitor the expenditure levels of the project to make sure that they are within the established contractual parameters established when the sponsor awarded the contract to the PM's organization. In case of a major contract change, the PM will consult the BFA, the LE, and the PE in order to identify the shortfalls that would need to be covered by the change request. The project management activity is triggered by periodic project status verification done by the PM or by notification of deviations sent by the prototype software application.

Prior to execution of the project management and corrective action activity, the contracted project must have been started. The project management portion of this activity will be executed periodically in order to monitor the progress of the project under development. The PM will be able to generate reports that he/she can use to discuss the progress of the project with the sponsor or with other members of the development team like the LE and the BFA. When the PE enters actual data that reflects that project activities are running behind-schedule or over-budget, the software application will alert the PM so he/she can analyze the project status and determine the appropriate path forward. The project status reports include artifacts generated by a sensitivity analysis of the project network to facilitate identification of which activities have the greatest effect on the critical path. One of these artifacts is a Tornado chart which "is useful for identifying the key variables and uncertain parameters that are driving the result of the model" (Vose 2008, 85). In other words, the Tornado chart will highlight those activities in the project that could cause the greatest effects on the schedule of the project. Another feature included in the project status report is a sensitivity diagram of the activities of the project under execution. According to Webb, "a sensitivity diagram shows the magnitude

of the effect of each variable on the measured outcome” (Webb 2003, 67). Different from the Tornado diagram, the sensitivity diagram will highlight those activities that cause the greatest variation in the duration and cost of the project. With this information, the PM can take corrective actions to help correct a negative trend or maintain the project on a path to achieve its requirements on-budget and on-schedule. Corrective actions like: using overtime, adding more resources or a re-baseline of the project plan are outside of the scope for this project, yet the outcome of implementing such actions will be reflected in the project status reports of the prototype software application. Table 15 lists the normal and alternate flows for the project management and corrective action activity.

Table 15. Decomposition of the Project Management and Corrective Action Activity

Source	Step	Action
Software	1*	Send notification of baseline project plan deviation to the PM
Project Manager	2	Open the “Cost and Schedule Trajectory” tab of the prototype software application
Software	3	Display the project activities, in a tabulated format, with the following information: baseline cost and schedule, actual cost and duration data, comments and risk level depending on the critical path analysis performed during the project planning phase
Software	4	Display the cost forecast, in a graphical format, for the planned schedule of the project
Software	5	Highlight in the project network diagram, the tabulated output of any activity that is over-budget or behind-schedule
Software	6	Display EVM metrics (i.e., BCWS, BCWP, ACWP, BAC, EAC, CV, SV, VAC, CPI, and SPI), for the reporting date
Project Manager	7.1	Contacts the BFA to make sure the overall project EVM metrics are within the contract stipulations established between the sponsor and the PM’s organization at the beginning of the project

Source	Step	Action
Out of scope	7.1.1	If the EVM parameters are not within contract stipulations, it will trigger a re-baseline of the project plan at which point the PM in collaboration with the LE, BFA, and PE, will perform the activities of the project planning phase to determine the cost and schedule variances that have to be included in the contract change request
Project Manger	7.1.2	If the EVM parameters are within contract stipulations, the PM will look for any highlighted activities in the project network diagram and in the tabulated display
Project Manager	7.1.2.1	The PM will look for those activities that are highlighted and are identified with a high level of risk during the simulation and analysis activity of the project planning phase
Project Manager	7.1.2.2	Once those activities, that meet the criteria described in step 7.1.2.1, have been identified, the PM can decide to perform any corrective actions available at the time including: overtime, adding more resources, etc.
Project Manager	7.1.2.3	For those activities where corrective actions were executed, the PM will add to their “Comments” field the dates when those actions were implemented and the kind of actions that were taken to improve their performance trend
Project Manager	7.1.2.4	Click “Update Cost/Schedule Trajectory”
Software	7.1.2.5	Update graphical display of the cumulative forecast of cost and schedule for the overall project taking into consideration the actual data for the activities that have been executed at the moment and the comments entered by the PM
Project Manager	8	Click “Generate Progress Report” button
Software	9	Generate project progress report showing the graphical and tabulated format of the project network diagram and the cumulative cost forecasted over the planned schedule of the project.
At completion of the project		
Project Manager	10	Click “Generate Completion Report”
Software	11	Generate completion report that includes important project data saved for historical purposes in order to improve the estimates of future projects

*Denotes optional path.

D. CHAPTER SUMMARY

This chapter presented current practices of project planning and execution processes which answered Research Question #1 and Research Question #2. The market research and stakeholder interviews showed that no single tool is currently used to generate project plans and to manage those projects once they are awarded to the developing organization. Different cost and schedule estimation processes were studied and further analyzed in order to determine the effects and methods for considering cost and duration uncertainties in the project plans. The market research for tools, which can perform the type of analysis needed to develop realistic project estimates that consider the cost and duration uncertainties of different activities, showed that many of the tools are quite cumbersome and not capable of performing such analyses on their own. A commonly accepted cost and schedule tracking process is EVM. The complexity of performing such a process limits its implementation to large projects (i.e., \geq \$20M). Conceptually, EVM does not consider uncertainty since it is based on a single project baseline. To account for this limitation, the managers of the projects implementing EVM establish a management reserve that is based on the initial budget and schedule estimate of the project.

A major problem of plans that rely on early estimations without further considering the inherent risks of the different tasks or activities is that they are usually inaccurate. Many software products will allow the user to generate schedules and estimates based on overly optimistic or pessimistic values (i.e., minimum and maximum cost and duration respectively) which might falsely give the impression that the cost and schedule estimates of a baseline project plan are realistic. PMs do not usually find out until the project is executed that it has been underestimated or overestimated, which could set the project up for failure by making it difficult for it to catch up to the established plan. This chapter presented an experiment that sought an understanding of the cost and duration values (i.e., minimum, maximum, mean, median, most-likely) that most often result in accurate estimates. It is shown that the utilization of minimum and maximum point estimates will deliver unrealistic estimation products that will not add

any value to the project plan; even planning using the most-likely values typically underestimates the overall project budget and/or schedule.

Finally, this chapter described a generic operational-based scenario that was developed to provide a framework of the operational needs and requirements of the prototype software application. The scenario does not contain any information attributable to the stakeholders that participated in this study, yet it is representative of the different processes that are found in NAVAIR organizations. Chapter III uses this information to analyze the users' needs and developed the requirements of the prototype software application.

III. REQUIREMENT ANALYSIS

A HCD approach was used to develop requirements and perform a requirements analysis. HCD is a process of development that focuses on making a product that actively considers the end-user throughout the design process. Stakeholder interaction is instrumental in the HCD process and results in more clearly defined requirements. HCD requirements development and analysis has some similarities to a traditional linear method of requirements development but is much more fluid in nature. The HCD requirements process and the process of refining those requirements are further explained in this chapter.

A. STAKEHOLDER INITIAL REQUIREMENTS

A total of six stakeholders were consulted for this study. Each stakeholder was interviewed using a specific list of questions developed by Team Merica that were designed to ascertain current planning and execution processes; see Appendix A. The goal of the interviews was to analyze how current scheduling and cost estimating are taking place within each organization, where frustrations are occurring for each stakeholder, and what improvements the stakeholders would like over current processes.

After each interview, the notes were hand recorded and placed on RealtimeBoard, www.realtimeboard.com, so they could be broken into categories and not be attributable to any specific stakeholder. RealtimeBoard is a web-based virtual whiteboard that allows users to collaborate and organize thoughts and ideas using virtual sticky notes. The concept adapts old methods of using sticky notes on whiteboards to organize thoughts and ideas and allows real-time virtual collaboration, which is very helpful for geographically dispersed teams. Sections and categories can easily be created on RealtimeBoard to facilitate organization. Notes from the interviews were initially typed on sticky notes below the stakeholder from which they originated. Then the notes were broken down into the following categories: relevant scope, software functions, network and user access, project description data, project planning phase inputs, activity and project models, simulation phase, project planning phase outputs, project execution

phase, error handling, and usability. These categories were created based on analysis of the notes from the stakeholder interviews, previous team member software development experience, and general project flow from planning to execution. Sections were created to organize specific software related notes, general project phase notes, functionality notes, and overarching scope related notes. Within each category the notes were organized based on the assumed level that its associated requirement would fall in a hierarchy diagram. If the note appeared to be a derivative of a higher level note, it would be placed below that note. If more than one stakeholder made similar comments, then the notes were stacked so that the relative importance of that note could be determined. Figure 19 shows a high level overview of the notes breakdown and Figure 20 shows a close-up portion of the Project Execution Phase section. Figure 19 is unreadable in this context but is placed to give relevance to Figure 20. Color coding was initially used to help organize the board and then as notes were turned into requirements the notes were changed to green. Both Figure 19 and Figure 20 are shown in process as notes were being converted to requirements.



Figure 19. Stakeholder Interview Notes on RealtimeBoard

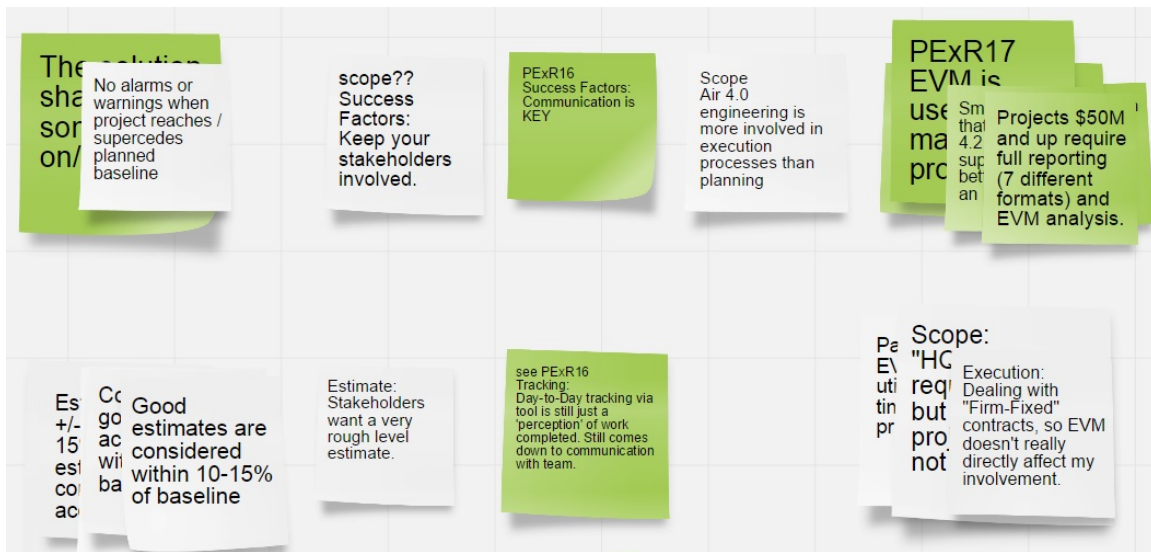


Figure 20. Portion of Project Execution Phase on RealtimeBoard

Once all the notes were distributed among their relative categories each was converted to a stakeholder requirement through a collaborative team effort. A stacked series of notes was converted to a single requirement that addressed all of the notes within the stack. A consolidated list of over 100 stakeholder requirements and derived requirements can be found in Appendix B. The themes which emerged from the stakeholder requirements analysis that provided a strong baseline for developing Team Merica’s prototype are listed below:

1. **Import Functionality:** There was a strong desire among the stakeholders for import capability from applications such as MS Project or Excel into a project planning tool.
2. **Database Utilization:** The stakeholders wanted a method of storing and retrieving data when using the software application. Utilizing historical data from previous projects and being able to retrieve and use data while at different locations were desired by the stakeholders. Having a central repository for data from current and past projects led the team to conclude that a database was the solution desired by the stakeholders.
3. **Collaborative Operations:** Having multiple users working collaboratively on the same project file was strongly desired by the stakeholders. The desire was to have multiple users working simultaneously with real time updates and to have different levels of access rights depending on the user’s level of responsibility.
4. **Geographically Distributed Access:** For many of the stakeholders, project teams are not always co-located so having the ability to access and use project files at different sites was highly desired.

5. **Usability:** Current methods of project planning and execution management are cumbersome and time consuming. Stakeholders wanted a software application that was easy to learn and use. Specifically, the stakeholders wanted a way to visually see the network diagram but also be able to quickly select, view, and add information related to a particular task.
6. **Modifiability** Projects are never static and are frequently subject to change. Stakeholders desired a way to quickly update task information so that feedback of project status was closer to real time.
7. **Uncertainty Consideration:** Current methods of project planning do not adequately address uncertainty. The stakeholders wanted a way to incorporate more uncertainty considerations into project planning so cost and schedule estimations are more reflective of the state of knowledge, less prone to underestimation, but still not overly conservative.
8. **Real-Time Project Execution Status:** A frequent concern of the stakeholders is that project status, in relation to cost and schedule, is always delayed due to slow data insertion and analysis. Stakeholders preferred a method to enter and analyze actual project performance data quickly and closer to real-time.
9. **Quick Project Status Determination:** The stakeholders also requested a way to quickly see and determine the status of the project without having to decipher all of the incoming data by hand.

B. STAKEHOLDER REQUIREMENT PRIORITIZATION

Prioritizing the stakeholder requirements was conducted by analyzing the number of notes that were generated on a specific topic and the relevance to the themes identified in the stakeholder requirements. A requirement was considered a higher priority if the majority of stakeholders addressed it during their interviews. A large stack of notes indicated that the requirement was either a frequent problem for most stakeholders or that it was something that must be included for the prototype software application to be accepted by the stakeholders. Also, the themes identified in the stakeholder requirements section were a strong indication of the stakeholder's priorities. The themes emerged from comments that were not identical but had strong correlations with each other. In many instances these themes overlapped and covered areas where multiple comments were made for a single topic. By analyzing both of these areas, a determination was made regarding which requirements were considered the highest priorities by the stakeholders, see Table 16.

Table 16. Stakeholder Requirements Prioritization

Item	Requirement
1	The software application shall generate EV and display in both textual and graphical format (PExR5)
2	The software application shall allow the user to select a probability distribution that closely models the cost of the project activity. (PPIn5.2.1)
3	The software application shall allow the user to select a probability distribution that closely models the duration of the project activity. (PPIn5.3.1)
4	The software application shall be able to retrieve information from a database. (SF2.2)
5	The software application shall generate a baseline project plan based on user inputs. (PPOR3)
6	The software application shall generate a cost and schedule forecast for the project. (PPOR1)
7	The software application shall allow users to enter actual activity performance data. (PExR1)
8	The software application shall display a Gantt Chart from the project network data. (PPIn1)
9	The software application shall be capable of selecting input files from Microsoft Project. (SF2.1)
10	The software application shall allow multiple users to enter data. (NR3)
11	The software application shall contain different levels of access rights i.e., read, write. (NR3.2)
12	The software application shall allow usage from any location with network access. (NR5)
13	The software application shall have a help button available during use and this will connect the user with the appropriate resources for resolution. (SF11)

C. REQUIREMENT ANALYSIS PROCESS

The requirements process for developing a HCD prototype is more fluid and does not follow the traditional requirements development process. The process for developing the prototype software application began in much the same way as a traditional process, with the stakeholder interviews. As outlined in Section A of this chapter, the notes from the stakeholder interviews were organized on RealtimeBoard, converted into stakeholder requirements, and then prioritized. Many of the stakeholder requirements were directly

transferable to prototype development; others, however, required refinement. For example, the stakeholder requirement that the software application be user friendly had to be converted into derived requirements that could be testable such as:

1. The software application shall display tooltip descriptions of the required inputs.
2. The software application shall allow the user to modify the activities from the project network diagram.
3. The software application shall allow the user to modify the data of the activity from the generated Gantt chart of the project.
4. The software application shall provide to the user the ability to enter the project data by prompting for the required information.
5. The user shall be able to update the display of cost and schedule trajectory with a single button.

Once the stakeholder requirements and derived requirements were established, a list of software specific requirements was created to drive the prototype development. The prototype software application was developed using an agile development methodology which utilizes frequent iterations to improve the prototype. More specifics of the software development will be discussed in Chapter V. The initial list of stakeholder requirements, derived requirements, and software requirements were updated with each iteration after direct stakeholder discussions. The prototype was shown to the stakeholders at each iteration and items that were liked or disliked were noted and the list of requirements was then refined. This methodology allowed more involvement from the stakeholder. Section A and Section B of this chapter outline the stakeholder initial requirements and the methods of prioritization. It should also be noted that after developing the stakeholder requirements, and with each iteration, some requirements were moved to future work due to the time constraints of the software development. Many of these requirements would greatly improve the functionality of the software application and are considered must haves for the stakeholders but their inclusion was beyond the capacity of the development team within the given amount of time.

D. CHAPTER SUMMARY

The use of HCD during requirements development resulted in extensive stakeholder consulting. The end-user was constantly considered as the software application requirements were developed with an emphasis on how to improve upon and mitigate the problems of existing tools. The stakeholder interviews provided the basis for developing the requirements and the subsequent prioritization. The utilization of RealtimeBoard facilitated the organization and visual prioritization of the stakeholders' wants and needs. After careful organization the stakeholders' wants and needs were converted to requirements and the frequency with which a want or need was expressed was directly attributable to its prioritization. During the requirements process several stakeholder themes emerged including desires for import functionality, database utilization, collaborative operations, geographically distributed access, usability, modifiability, uncertainty considerations, real-time project execution status, and quick project status determination. These themes and the frequencies of stakeholder wants and needs facilitated the requirements prioritization that was utilized for developing the software. The requirements described in this chapter along with the operational-based scenario describe in Section C, of Chapter II of this report, provided the foundation for the development of the functional architecture used to developed the prototype software application which is further explained on Chapter IV.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. ARCHITECTURAL DEVELOPMENT

Once the requirements from the stakeholder analysis were analyzed and prioritized based on the interview notes, the system's proposed functionality was placed into a functional architecture diagram. According to Buede (2009, 211):

The functional architecture of a system contains a hierarchical model of the functions performed by the system, the systems' components, and the systems configuration items; the flow of informational and physical items from outside the system through transformational processes of system's functions and on to the waiting external systems being serviced by the system; a data model of the system's items; and tracing of input/output requirements to both the system's functions and items.

A. ARCHITECTURAL MODEL DEVELOPMENT

An IDEF0 modeling technique was used to create the primary elements of the project's functional architecture. The use of this technique was chosen due to its standardized syntax and the group's desire to focus on the flow of data between the project's individual functions. Each of the primary functions was created as a box with data flowing in or out as an input, output, control, or mechanism as seen in Figure 21.

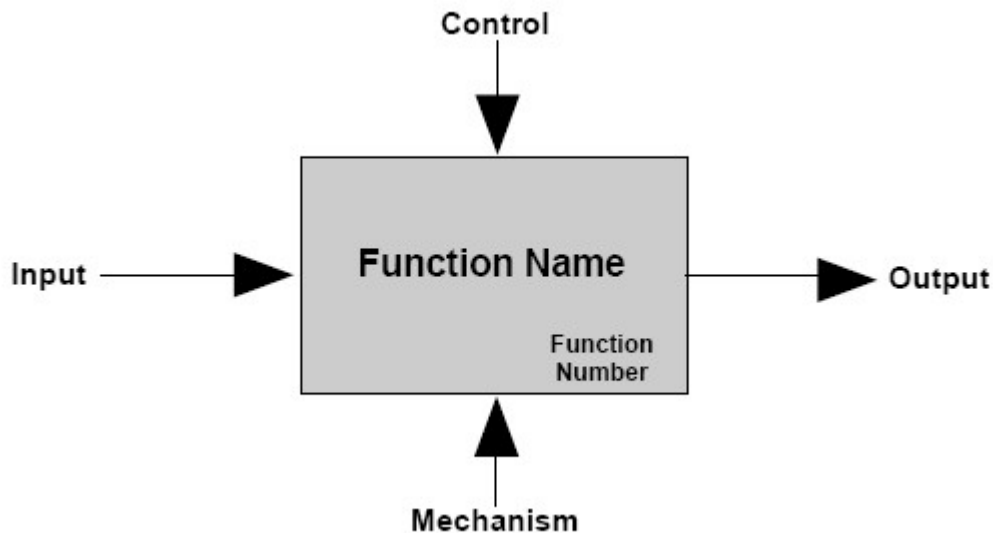


Figure 21. IDEF0 Functional Building Block (from DAU 2001, 51)

The architecture began by identifying the primary functions and functionalities for the prototype software application in order to fulfill the operational concept. These functions were the result of analyzing the operational scenario, generated requirements, and the intended use of the prototype software application to determine how to take the inputs and transform them into the required outputs. With the system's functions resolved the next step was to begin structuring the parts together through the top-down decomposition method. Using this method, the prototype software application was slated as the top level function and then broken down into several sub functions for a closer look at the lower levels. The next step was to focus on the data that would serve as our inputs and outputs for each of these sub functions and how the information would need to flow through the prototype software application. As the functional architecture began to take shape, the last step was to identify and trace the connections between the previously determined input and output requirements to the flow of data elements through the architecture. Each requirement was traced to items or functions within the model to accomplish that particular requirement.

B. FUNCTIONAL ARCHITECTURE

The functional architecture of the prototype software application was created by using the IDEF0 methodology covered above and started by defining the functional hierarchy. Six (6) separate main functions were created to satisfy the previously determined system requirements: Develop Project Model, Compute Plan, Track Project Performance, Identify Deviations, Save Function, and Create Output Reports. These functions each hold a purpose devoted to a different operating mode within the overall prototype software application's functionality. For example, "Develop Project Model" was created to provide the user interface aspect of the system and "Track Project Performance" provides control over the information coming in from various inputs. Each of these primary system functions was then broken down further into several sub-functions, each one created to contribute toward the higher function's objective. The high level functional hierarchy for this project can be seen in Figure 22.

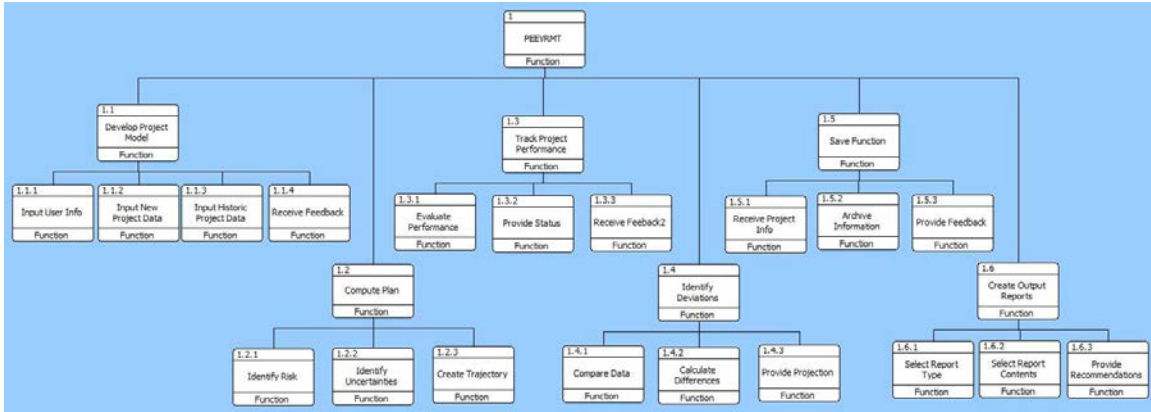


Figure 22. PEEVRMT Functional Hierarchy

With the functional hierarchy entered, the next step was to determine the data flow, controls, and mechanisms required for each function. Each data item was created to satisfy one of the system’s requirements that were created earlier. At the top of the diagram the primary inputs are the base data, both new and old, and the main output of the prototype software application is a custom report tailored to the stakeholder’s preferences. In between the main inputs and outputs the data travels through almost every function, transforming each time. For example, the “Compute Plan” function takes incoming project data in the form of parameters and transforms it into several different cost and schedule trajectories. By creating these connections between each of the functions and determining how the data would flow through the prototype software application, the functional architecture was created and can be seen in Figure 23.

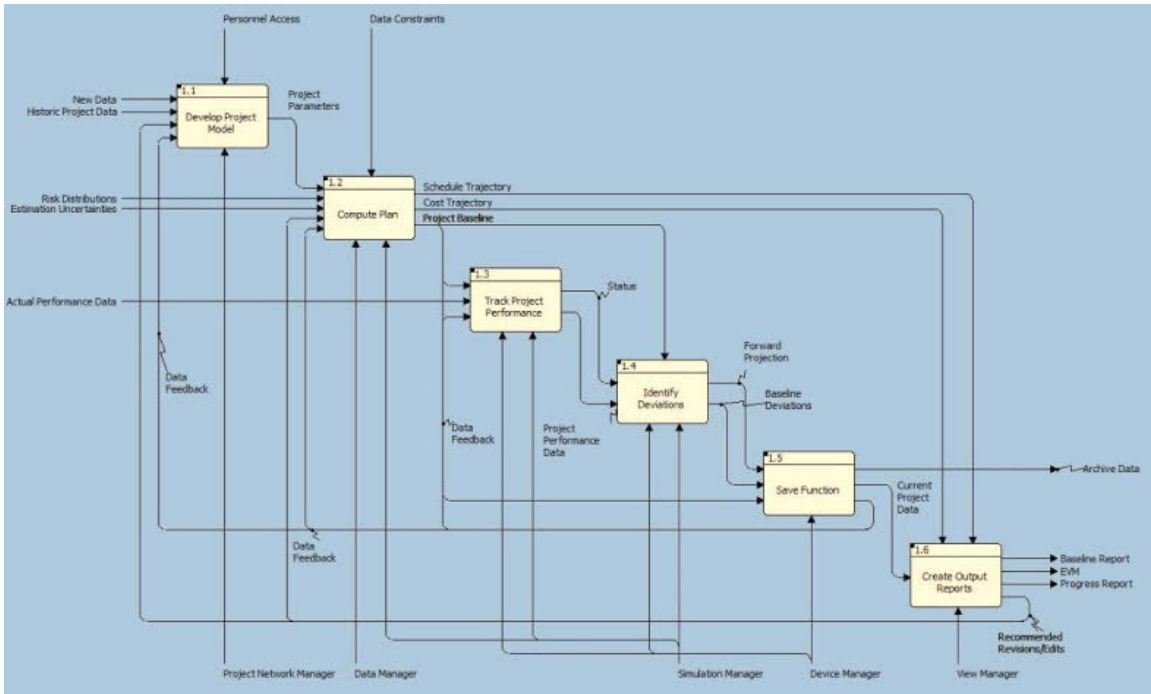


Figure 23. PEEVRMT IDEF0 Diagram

Along with the Input and Output flow of data through the prototype software application, functional controls and mechanisms were added to define the architecture further. Many of the control requirements had not been created at this point in the prototype software application's life cycle so the functional controls were limited during the creation of the IDEF0. Some of the constraints that were added include personnel and data constraints along with cost and schedule trajectories created by the Compute Plan function. The mechanisms for the prototype software application were designed as separate components and/or systems within the software application package in order to perform the individual functions and to complete the various prototype software application requirements. The following items were created to manage each function during the system's operation: Project Network Manager, Data Manager, Simulation Manager, Device Manager, and View Manager. Also, several of these mechanisms tie into numerous prototype software application functions to help govern multiple aspects of the overall functionality. For example, the Device Manager mechanism interacts with three different functions while the View Manager mechanism only interacts with one.

These interactions can be seen in close up portions of the IDEF0 shown below in Figure 24 and Figure 25.

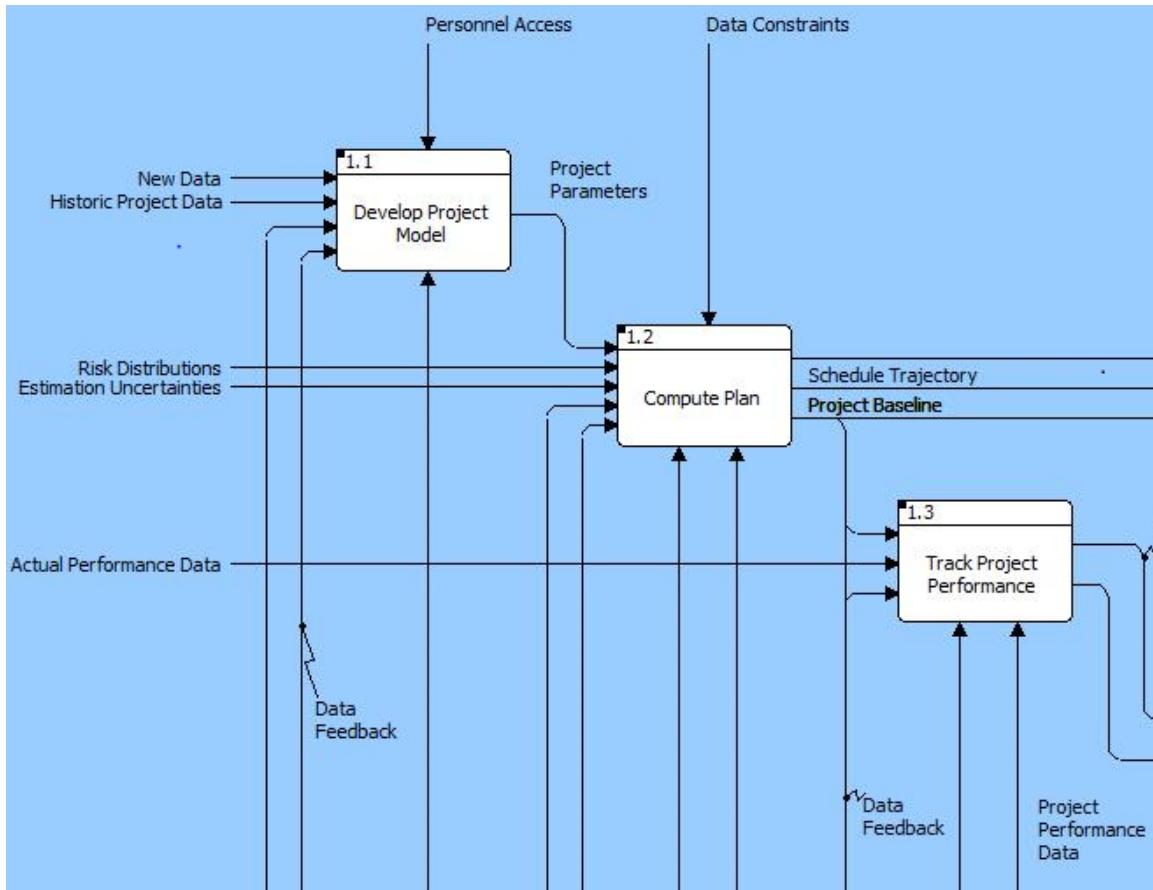


Figure 24. PEEVRMT IDEF0 Diagram Functional Upper Section Close Up

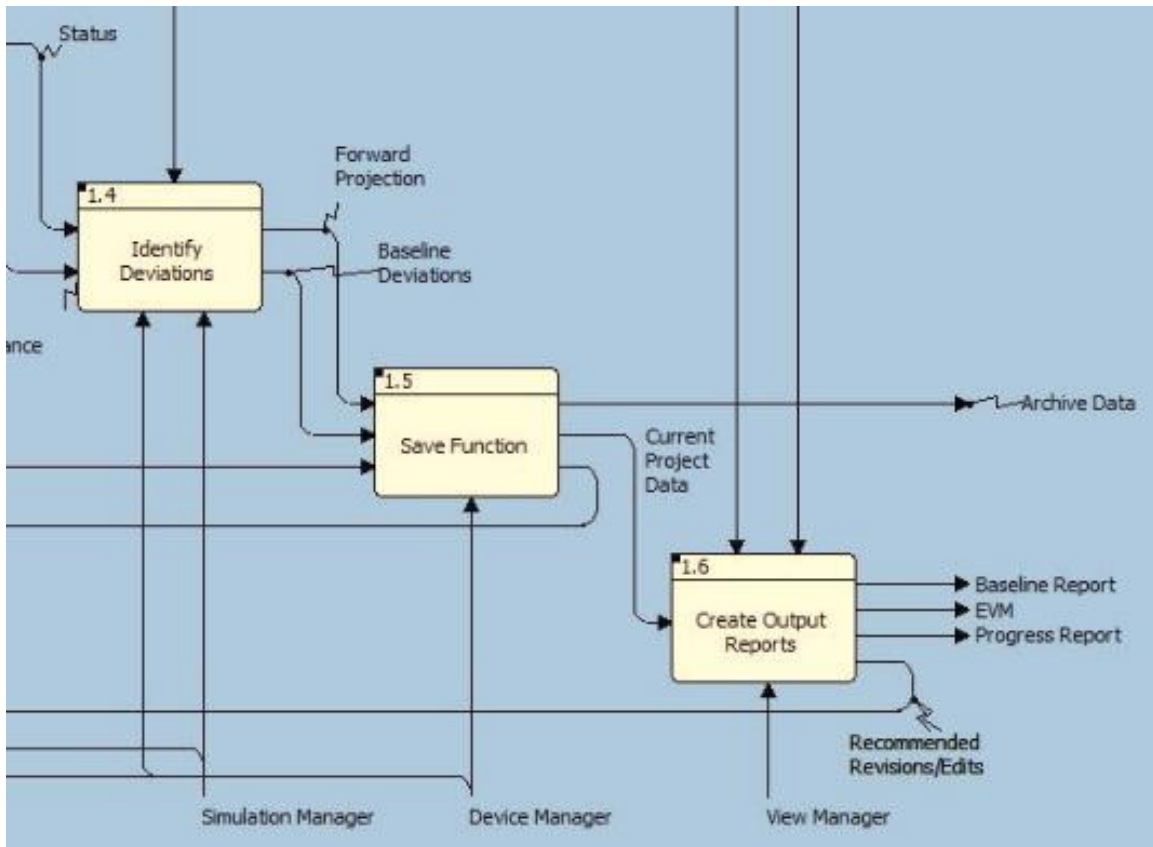


Figure 25. PEEVGMT IDEF0 Diagram Functional Lower Section Close Up

The final aspect of the Functional Architecture that was incorporated into the design was the idea of prototype software application feedback. The notion of continuous feedback while the prototype software application was running was an often-discussed idea during stakeholder interactions and the architecture reflects that requirement. Several functional outputs were designed specifically to provide a closed loop control for some aspects of the prototype software application operation. Data outputs such as Data Feedback and Recommended Revisions/Edits were taken as individual outputs and then re-inserted into the system to provide an internal check of the prototype software application. Ideally, these feedback loops were designed to attempt to close any gaps between the current data output(s) and the data that was desired.

C. CHAPTER SUMMARY

With the stakeholder requirements processed and prioritized from numerous interview notes, the system's functional architecture was designed to accommodate the proposed functionality. Using the IDEF0 modeling technique, the required prototype software application's primary functional elements were identified and organized. A functional hierarchy was used to decompose the prototype software application into numerous primary and sub-functions to provide a top-down view of the functional breakdown. With the hierarchy in place, the IDEF0 was created to provide requirements oversight and ensure that the previously determined requirements were adequately addressed in the architecture. Data inputs and outputs were created to address the functional flow of the prototype software application. Controls and mechanisms were identified to provide the software components required to complete each function. Lastly, a large emphasis was placed on designing feedback into the prototype software application's architecture so that it could identify and correct any deficiencies rapidly across multiple functions.

THIS PAGE INTENTIONALLY LEFT BLANK

V. SOFTWARE DEVELOPMENT

The principal product developed from this study effort was a prototype software application designed to advance the current technological shortfalls present in current industry project planning tools. Through the integration of modern systems engineering principles, a software application concept was developed, capable of accounting for uncertainty within individual work packages of a project WBS and reporting EVM metrics accounting for those aforementioned uncertainties.

This chapter details the methodology, and system specification interpretation for the prototype. Also discussed is the specific prototype development process with special emphasis on the HCD principles utilized in development and test verification and validation (V&V). Finally, this chapter will compare and contrast the implemented concepts against commercial alternatives.

A. SOFTWARE DEVELOPMENT PROCESS

Early in the development effort, the Scrum process was determined to be the most appropriate development model for this prototype. Scrum, being a derivative of traditional agile principles, was a logical choice due to the time constraints allocated for development, the size of the overall developmental effort, and the size of the developmental team, which consisted of three members of Team Merica who were familiar with software application development. The Scrum process complements HCD philosophies through the involvement of stakeholders throughout design and implementation, and is adaptive to changes in requirements and scope as the product vision is communicated over time.

The process pattern consisted of weekly scrum meetings with the development team and key stakeholders. Completed developmental activities for the week were merged into a working subset of the final solution and demonstrated. After demonstration, the development team, requirements officers, and stakeholders would go through a process of reprioritizing the project requirements, clarifying the objectives from

the point of view of the end user, and finally allocating responsibility for the subsequent development sprint.

Early iterations of these sprints produced ancillary products to the final prototype such as:

1. High level graphical story boards
2. MS Excel interactive prototypes
3. Architecture views, and functional flow

While some of these products would not be considered software in terms of final deliverables, these products were viewed in the same light as formal requirements and as such facilitated in removing layers of separation between the customer and implementers.

Later iterations provided stakeholders with stable solutions that served as a common launching point to discuss features yet to be implemented as well as early informal suitability tests. The resultant discussions from these sprints emphasized intuitive workflow, comprehensible output, and comprehensive functionality.

B. OBJECT-ORIENTED PARADIGM

The architecture development, through the allocation of functional modules to “physical” components, lent to the selection of an Object-oriented Programming (OOP) language. Several languages were considered based on stakeholder criteria, and implementation flexibility. Ultimately, due to interoperability requirements from the stakeholders to integrate existing project planning methodologies and software engineer’s familiarity with syntax and structure, the C# programming language was selected as the development environment for the final product. This decision limits some of the cross-platform capabilities for the prototype; however, based on stakeholder feedback, the vast majority of project planning activities are conducted on systems operating in the Windows Environment.

Figure 26 shows the structure of the prototype’s main objects as a modified Model-View-Controller architectural pattern. Users interact with the program through objects within the view manager while functional invocations are propagated through the

application manager, disseminated to specialized objects within other managers, and ultimately changes to the view are filtered back to the view manager.

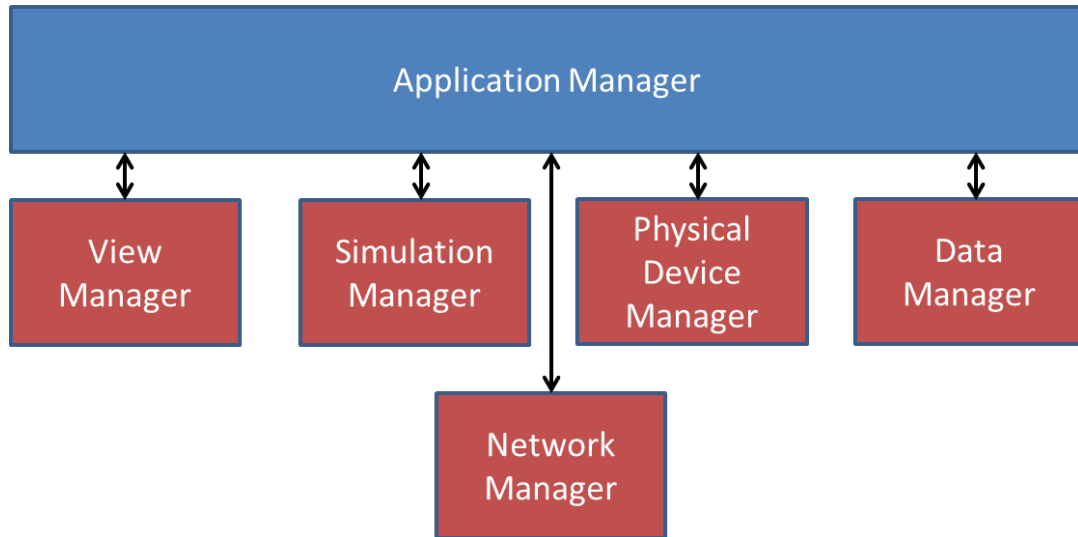


Figure 26. Prototype Software Application Model-View-Controller Architectural Pattern

The strength of this pattern allows for the rapid integration of additional features. Additional views can be introduced through a child of the View Manager Class for example. This way of visualizing the programmatic structure and flow of information facilitated communication with the stakeholders and allowed developers to maintain consistency between the designed system architecture and developed code. Traceability between the functional requirements and physical implementation could be followed through the project design down through the physical implementation.

C. DOCUMENTATION AND SOURCE CONTROL

Working in a geographically distributed environment posed unique challenges to collaborative development. Wide varieties of tool sets exist to aid teams in rapidly and effectively work in this setting. The Naval Postgraduate School (NPS) provided developers with repository space in their GitLab, which allowed for powerful versioning control as well as configuration management. A member of Team Merica was designated as the lead integrator. The lead integrator tasking included managing the main

developmental trunk, and integrating individual contributions to the current configuration. This task was completed weekly at the outset of the Scrum sprint meetings. Git was chosen due to ease of use and adoption as an industry standard for source control. Adoption by NPS as their preferred method of source control enabled easy integration into their infrastructure.

Documentation was a critical activity for this development since highly technical source code has a tendency to become obfuscated. Comments were needed to effectively communicate the purpose of functional modules and programming interfaces to consumer objects, while similarly, documentation was needed for product capabilities and end user consumption.

The development team chose the open-source tool Doxygen, which is a documentation generator used with programming languages such as: C++, C, and C# among others, to properly document the source code of the prototype software application and maintain consistency between developer comments. This tool utilized standard comment markups in code which conveyed information sufficient to produce final user documentation as well as programming schemas for technical developers. By embedding documentation in code during development, the developers eliminated a major hurdle to creating consistent and up-to-date documentation. Doxygen scanned the source code and automatically generated user manuals as well as HTML sitemaps that completely described the scope of the software's most up-to-date iteration.

D. SYSTEM SPECIFICATION DEVELOPMENT

System specifications for this project were intentionally designed to be high level and, as is the case with many Scrum projects, refined over the course of project development. The HCD methodology relies on the ability of project developers to see the scope of the project from a number of different viewpoints, the most important one being the point of view of the end user. With HCD, the design is quickly sketched out and presented to the stakeholders several times over the development. This design philosophy departs from the traditional systems engineering process which mandates detailed and exhaustive requirement definition, mainly at the beginning of the project. Team Merica

maintained the top level perspective, keeping the end user in mind, while participating in the ideation process with the stakeholders to refine the requirements and concept of operations. This refinement process produced more buy-in to project outcomes from the perspective of the stakeholders, while enabling the development of a system which meets the needs of the user.

During the discussions with the stakeholders, the foremost concern for this effort was the ability to estimate the impact to cost and schedule over a planned program lifetime when there was uncertainty within the program. To facilitate creating this estimate(s), the development team envisioned the software program would use MC simulation to analyze cost/schedule networks created by the user and present the results in such a way that was intuitive and allow the user to determine whether their program was executing according to the parameters specified in the cost/schedule network.

To summarize, the prototype was to be a software product that maintains compatibility with current tools, enables the creation and modification of scheduling networks, accounts for statistical uncertainty, generates reports for decision makers concerning programmatic risks, and allows for progress tracking through accepted EVM metrics or through metrics conveying similar information in a more readable form. From this high level concept, the team's software engineers began the iterative process of refining and developing supporting requirements detailed in Appendix B and produced several different low-fidelity prototypes.

E. PROTOTYPE DEVELOPMENT

The prototype development was composed of an Initial Prototype, which was used to illustrate current shortfalls in estimation practices and to develop the computational engine of the developed software application. The next step was the generation of story boards, which is part of the ideation process of the HCD method used to address the users' interactions with the prototype software application. Lastly, a Final Prototype was developed to address the stakeholder requirements and to demonstrate the benefits of considering cost and schedule uncertainties in an integrated project management tool.

1. Initial Prototype

To facilitate understanding between stakeholders and the design team, a semi-functional prototype was developed to illustrate shortfalls in current scheduling practices as well as propose a solution to evaluate programmatic risks. Figure 27 shows an example scheduling network from this primitive prototype. In this example, a sample static network configuration is displayed as well as triangular distribution parameters for both cost and schedule for each task. Paths through this network were manually determined providing traceability between the first and final tasks.

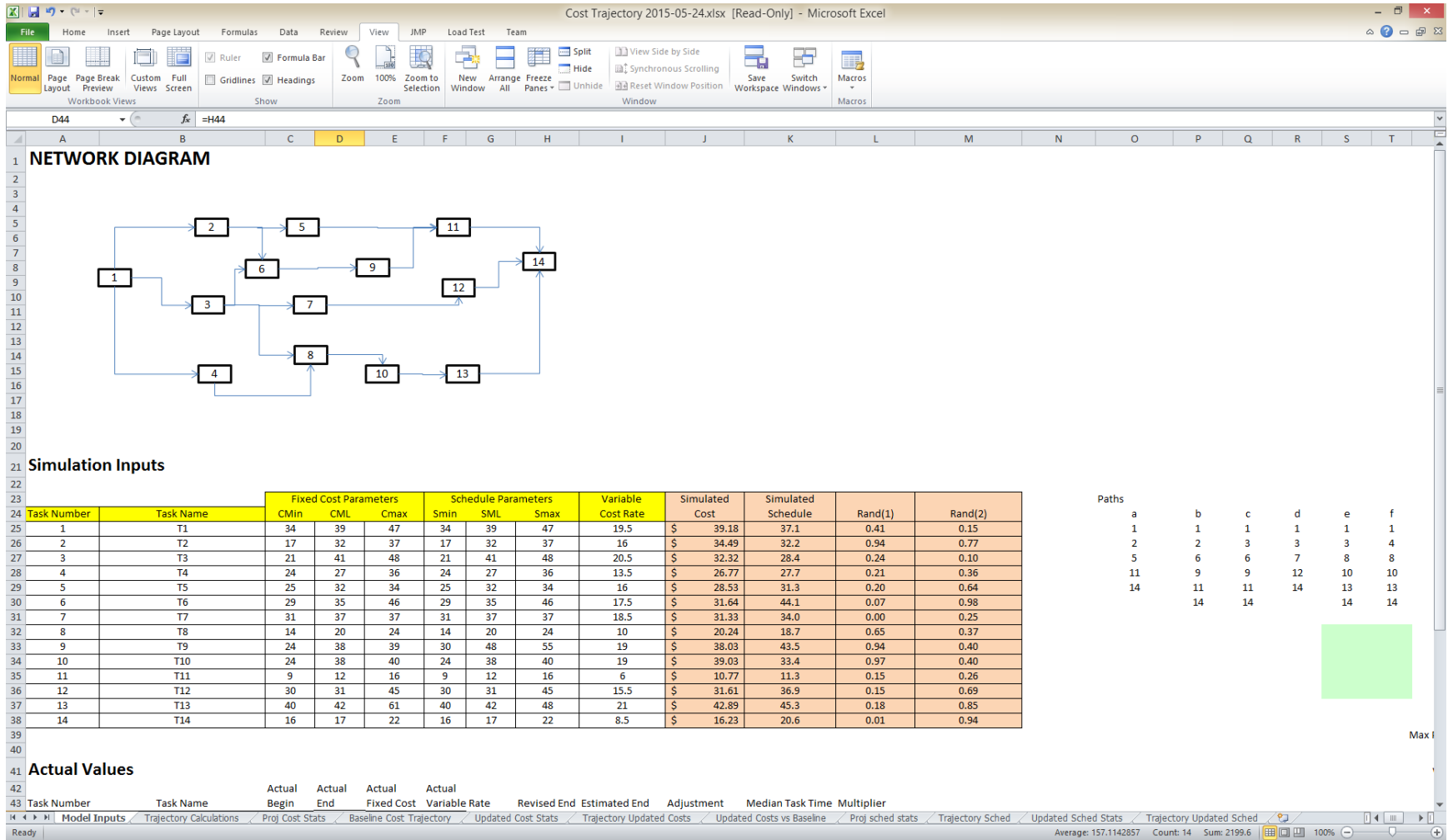


Figure 27. Initial Cost-Schedule Analysis Demonstration Prototype in MS Excel

Project costs, for the purposes of this prototype, were classified in two categories: fixed cost and variable cost. Fixed cost, as intended for project management purposes, represent one time investment costs for a given work item. Material costs for instance would fit the intended usage of a fixed cost. Variable costs represent costs that are directly tied to schedule. Hourly rate and working overhead are examples of costs which increase linearly with schedule. In projects where there is a requirement to estimate material costs, distribution parameters can be defined through the “Fixed Cost” columns. Similarly schedule durations to complete a task can be given in the “Schedule Parameter” columns. While these cost estimates are fairly basic and probably not all inclusive for a real project, they were considered sufficient at this stage of development.

Each of these cost and schedule parameters were provided to the MS Excel prototype to describe the intended distributions. The COTS software Risk Simulator developed by Real Options Valuation Incorporated, was used to provide the simulation engine for this prototype consisting of random number generation, graphical output, and replications. On completion of the simulation replications, output was stored on a separate worksheet in MS Excel and cost phasing algorithms, allocating total cost incurred at specific schedule intervals, were overlaid on the output predicted by the simulations. The simulation output shows the minimum value, the 20th percentile, the median, the 80th percentile and maximum of the simulated cost and schedule range for each of the specific schedule intervals. During the course of project execution, a user would enter actual cost and schedule data and the tool would provide output indicating the forecasted schedule and cost as seen in Figure 28. Models built in MS Excel were able to be saved and edited.

There were several limitations or drawbacks for the initial prototype. The first limitation using the MS Excel model is the amount of manual work required to setup the conditions of the simulation. The paths through the network had to be determined manually. The second limitation is that the critical path(s) could be determined but the path(s) had to be among the paths already identified by the user. The number of potential paths through a network grows very quickly as the number of tasks and complexity increases, which could easily lead to missing paths through the network. The third

limitation is that complex networks can exceed the computation capabilities of MS Excel. Finally, as stated previously, the network diagram is a static picture, and not a dynamic display.

The strengths of prototype were the integration of MC Simulation on the dataset. Secondly, MS Excel provided a rapid ability to create output required by the stakeholders in the form of graphical charts. The initial prototype was also capable of capturing progress of a product. Finally, MS Excel provided users with Save/Edit Functionality as a standard user function requiring no specialized development to implement.

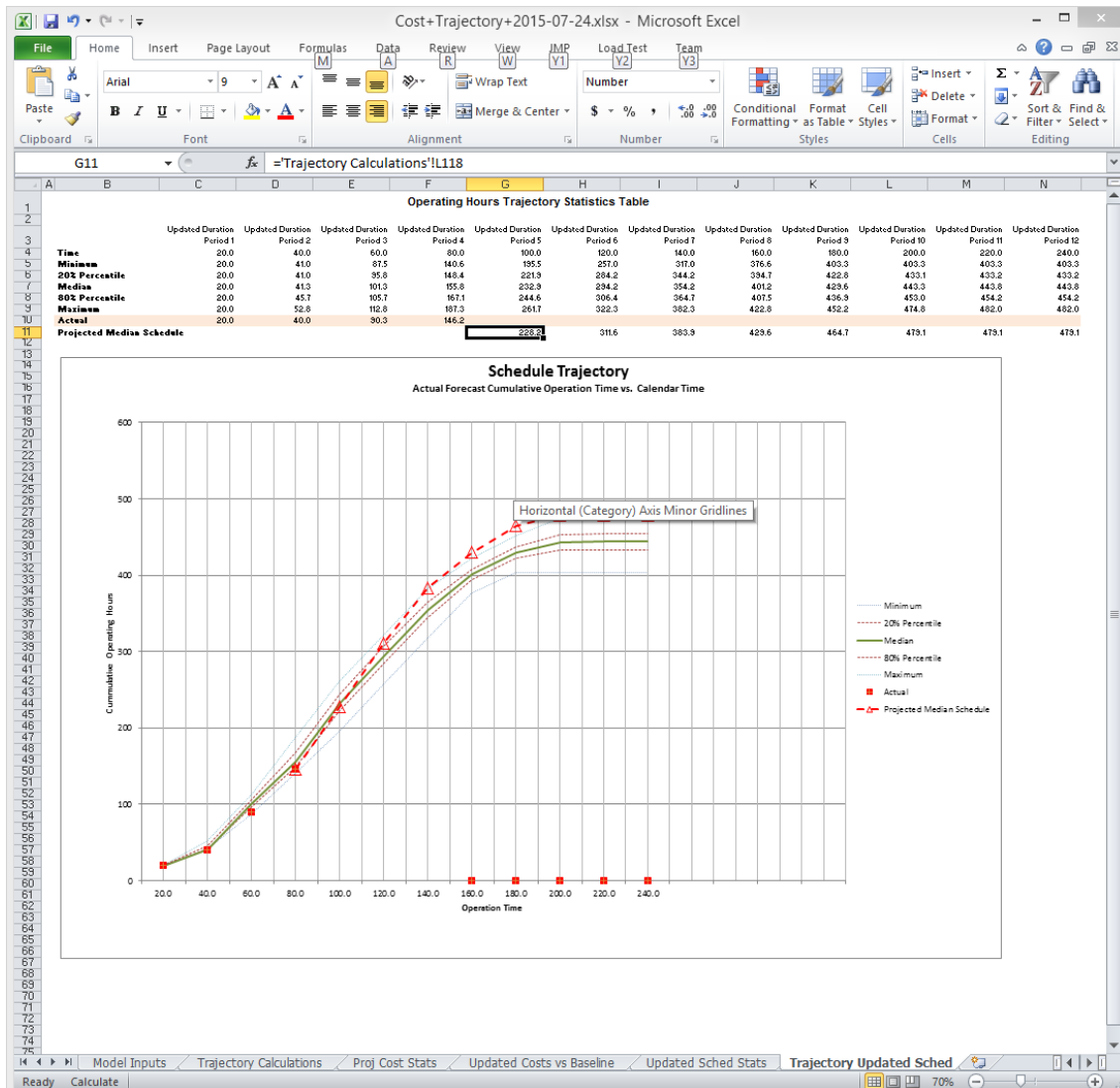


Figure 28. Cost-Schedule Forecast and Trajectory Demonstration Prototype

2. Ideation Prototype

The static nature of the initial prototype was the principal drawback for large scale adoption. It did not integrate with existing products, and would need to be developed from scratch for each program, and in many cases, for programs with large and complex networks, the number of potential paths from start to finish would exceed the capability of both MS Excel, and would be infeasible to set up. The ideation prototype took the development to the next stage through storyboards and further development of the MS Excel models.

Through the stakeholder interview process and subsequent ideation sessions, storyboard mockups were developed in conjunction with attempts to dynamically create MS Excel files which could be generated through processing MS Project documents. Figure 29 shows the Project Description tab of the prototype concept that was created through the ideation process in conjunction with the design team and stakeholders. It also shows the ability to capture the high level project description information as well as the ability to import a project network from a third party source.

PEEVRTM APPLICATION

Project Description | Project Network | Simulation Parameters | Planning Tab | Cost Trajectory | Help

PROJECT DESCRIPTION

Project Manager:

Lead Engineer:

Financial Analyst:

Project Start Date:

SPONSOR INFORMATION

Sponsor Agency:

Sponsor POC:

PROJECT DESCRIPTION

Figure 29. Prototype Software Application – Project Description Tab

Figure 30 shows the proposed user interface for viewing the current network as well as the capability to design and modify a network and associated cost/schedule parameters. By right clicking a task on the network diagram, a dialog window would be presented to the user which allowed the manipulation of the statistical parameters as well as the data that defined the task parameters in relation to the WBS. During discussions, one stakeholder desired the ability to import data for tasks that were built from subordinate network models. A “Use Task Model” placeholder was placed on this user interface for that purpose.

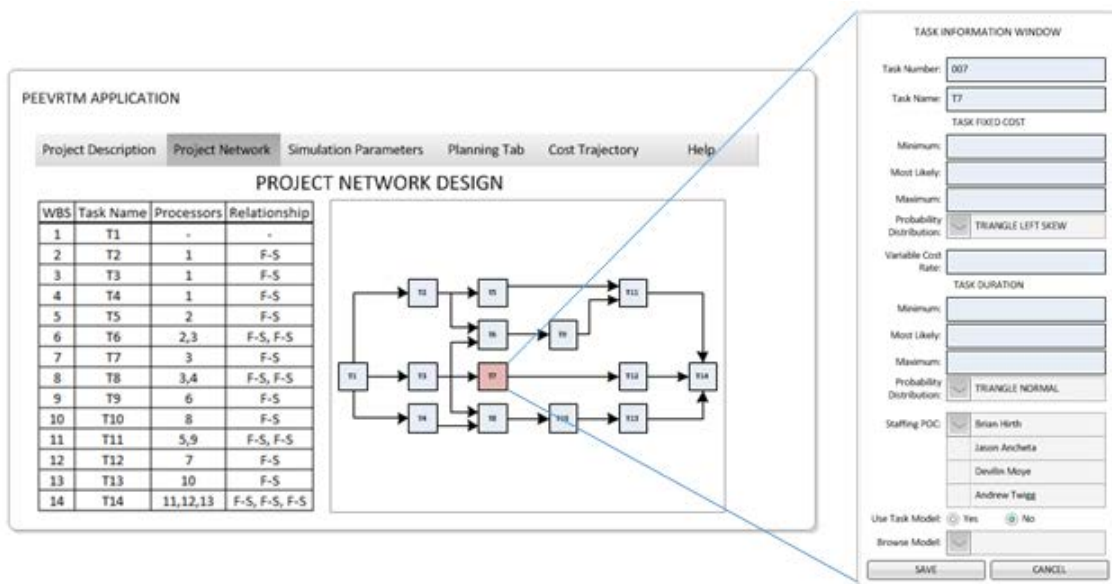


Figure 30. Prototype Software Application – Project Network Tab

The Planning Tab shown in Figure 31, shows how the user envisioned being able to run the simulation once all parameters were entered, as well as a summary of simulation results for high level verification. The currently planned critical path would also be presented in a different color from the rest of the network and would be updated as the user altered the network design providing real-time situational awareness to the properties of the system, and a baseline point estimates of cost and schedule. Management reserve, a risk mitigation technique currently implemented to address programmatic uncertainty, is not provided in this application. This study sought to

provide better estimates for program execution through statistical analytics, thus all activities need to have an associated uncertainty distribution.

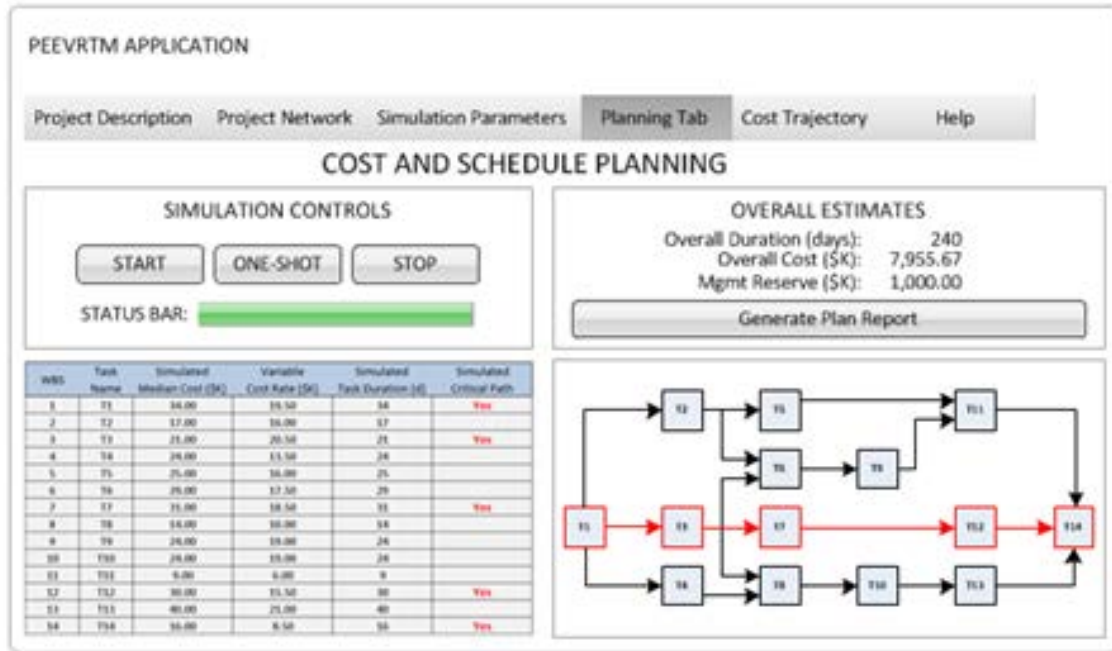


Figure 31. Prototype Software Application – Project Planning Tab

Figure 32 shows the functional view that illustrates different output views as well as provides a user with the mechanism designed for establishing the baseline plan of a project and tracking its progress. An alternative output display using EVM was requested by some of the stakeholders. The development team envisioned using the average or median values of each task distribution to build the EVM output, but the team ran out of time before they could create that part of the prototype.

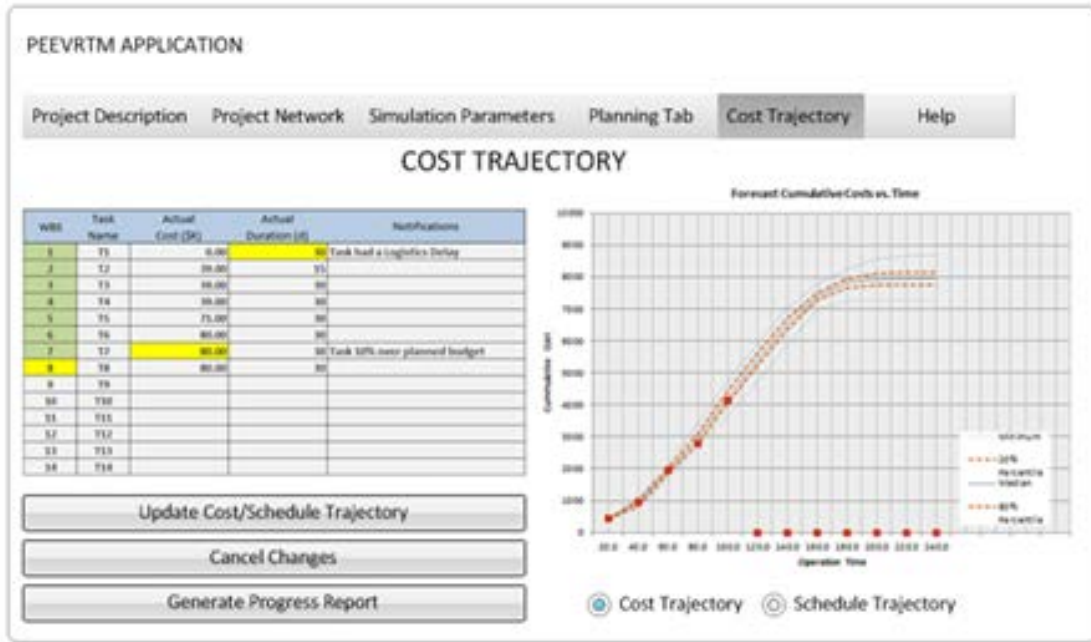


Figure 32. Prototype Software Application – Project Cost Trajectory Tab

To further the automation of the ideation prototype, the developers designed a methodology for reading MS Project files and completing MC simulations using the schedule networks imported from the MS Project files. In Figure 33 the scheduling network derived from the MS Project files, are represented as an adjacency matrix. Task durations are automatically entered, and standard distributions were applied using the *U.S. Air Force Cost Risk and Uncertainty Analysis Handbook (2007)*. As the storyboard concept developed through several evolutions, MS Excel proved to be a valuable resource for extremely rapid prototype development on key concepts. Various parts of the storyboards which required calculation were further refined within these MS Excel models.

While not functional at this point, the ideation prototype accounted for all functionality envisioned to fulfill the intended outcome of this project, specifically a proof of concept software application that accounts for cost/schedule uncertainties, provides realistic and achievable estimates for the duration of the project and provides stakeholder management staff a method to track actual project performance in terms of

cost and schedule in relation to simulated estimates. Features of the ideation prototype include: dynamic project network display, simulation on cost-schedule parameters, reporting products, critical path analysis, actual cost-schedule tracking, save/edit functionality, and the ability to handle complex networks.

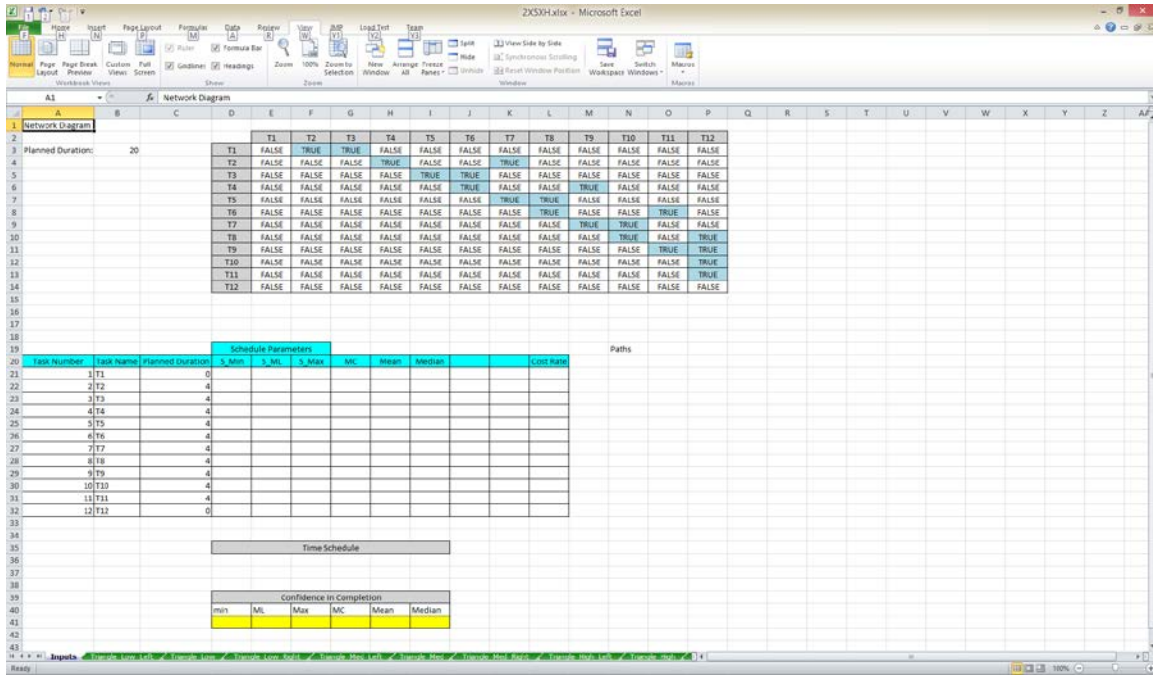


Figure 33. Prototype Software Application – Simulation Output Demonstration

3. Final Prototype

The final prototype represents the team’s best effort in incorporating the requirements of the stakeholders into a working product which, given future development, could represent a significant leap forward in the systems engineering tasks of project management cost/schedule predictions.

The screen views for this prototype were designed specifically to mimic the storyboard designs in Figure 29 through Figure 33.

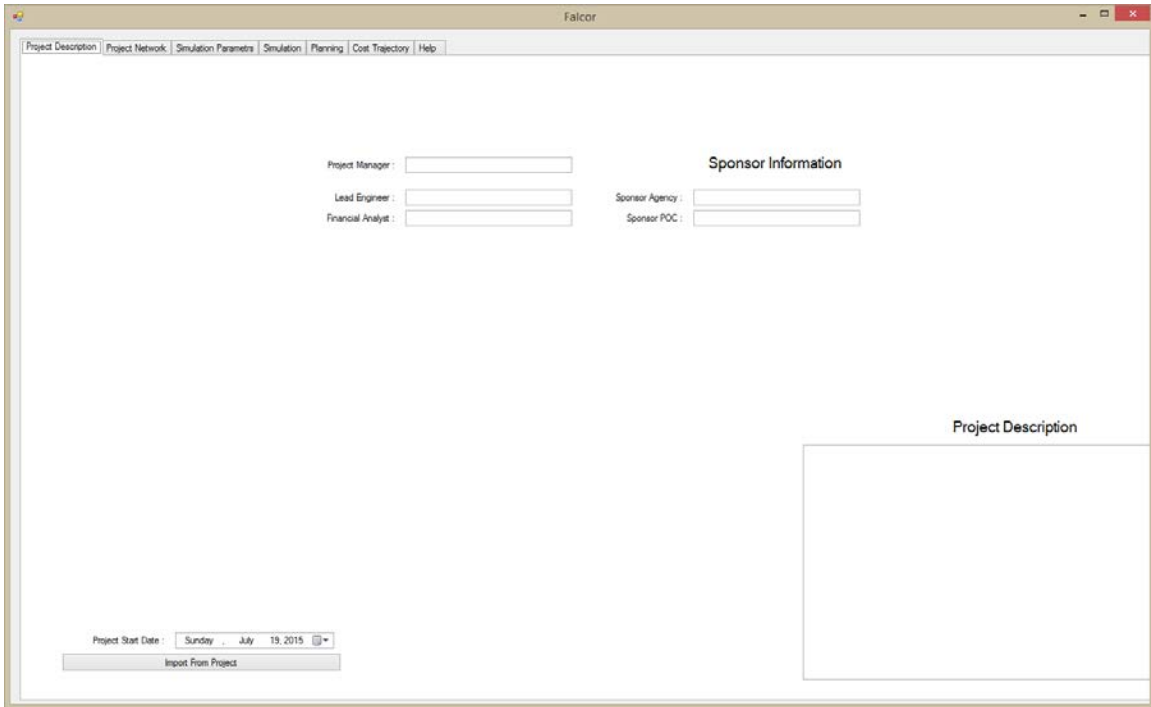


Figure 34. Software Application – Project Description

Figure 35 shows the intuitive nature in which project networks can be visualized. The left panel of his figure allows for users to dynamically edit the WBS, task name, and planned task durations. Graphically, users can modify dependencies between tasks and modify the displayed layouts through a click-drag interface. More complex scheduling networks are also supported in this design through allowing for task decomposition as seen in Figure 36.

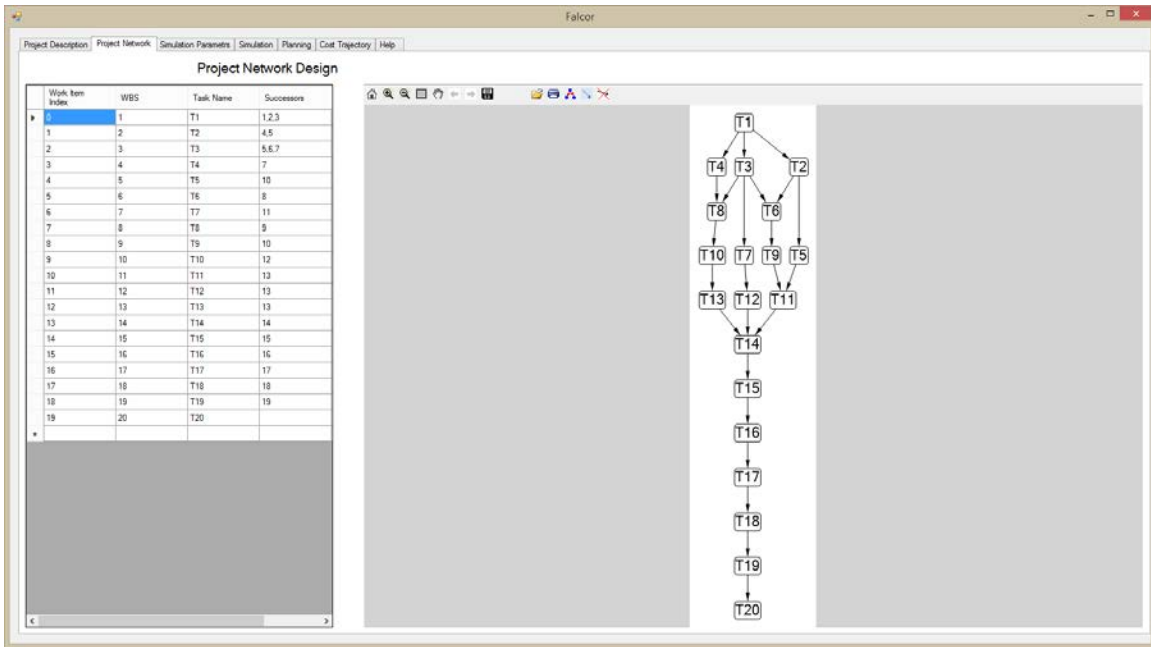


Figure 35. Software Application – Project Network Design

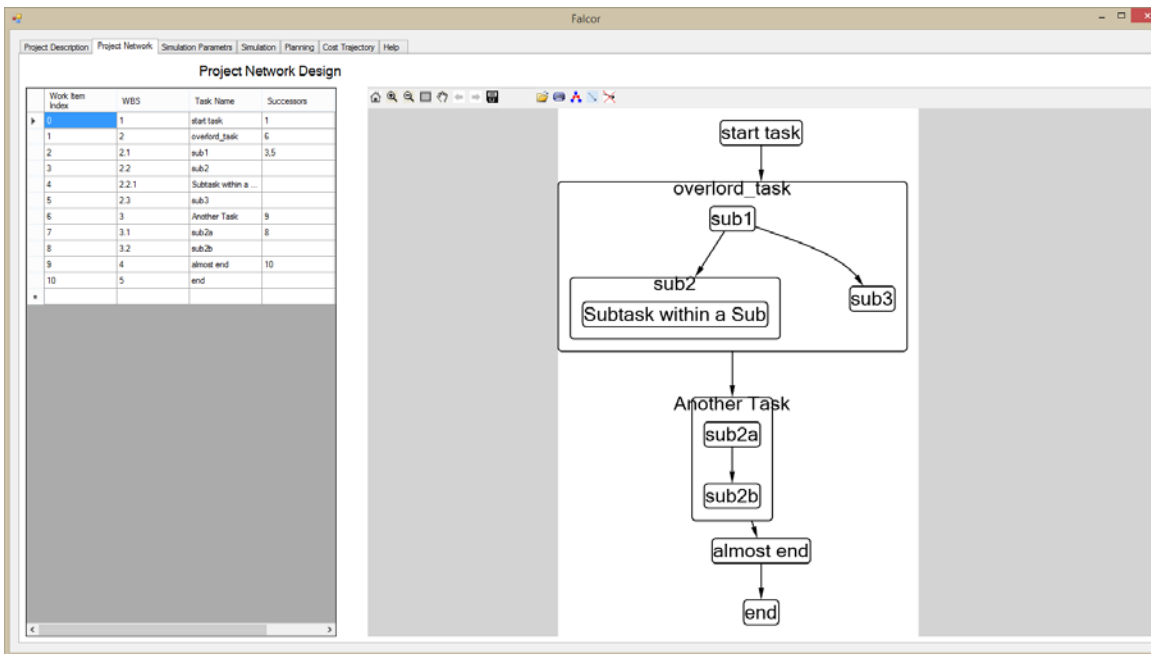


Figure 36. Software Application – Project Network Design Alternate View

In this case, complex project networks can be visualized in a structure which adds clarity and context to the project in a way not currently supported through MS Project.

In Figure 37 the user has access to the Simulation Parameters which can be changed and subsequent output can be stored locally. By increasing the number of simulation runs, planners can be confident that output obtained by this program has their desired level of statistical confidence associated with it. Selection of identical seed values between runs gives other users an ability to replicate results independently. This solution utilized an algorithm developed by L'Ecuyer and referenced by Law (2007) that combines multiple recursive generators with 10,000 possibilities for seed values. The strength of this generator ensures stable statistical properties over many random variate generations (Law 2007).

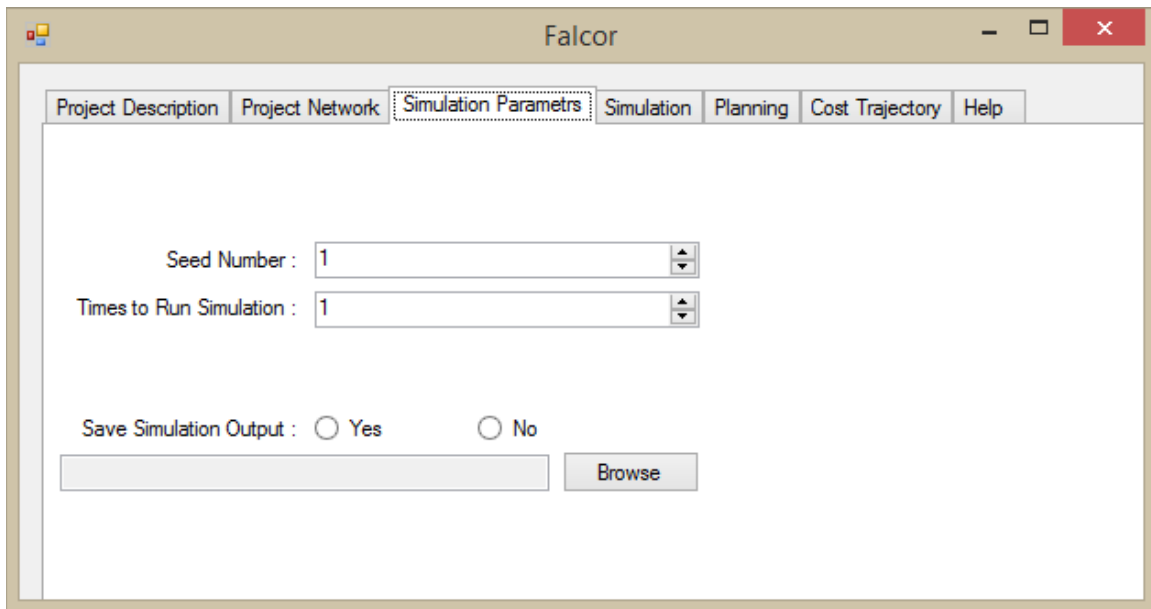


Figure 37. Software Application – Simulation Parameters

The Project Planning tab shown in Figure 38 informs users of simulation results and individual WBS cost estimates using median cost as the point estimate for the required resources each work item requires. The unimplemented features for this tab consist of aggregating the simulation point estimates and then displaying a color coded network diagram showing critical and near critical path variants as determined by the simulation runs. This could give a program manager early situational awareness to high risk tasks that could dramatically affect overall project cost and schedule.

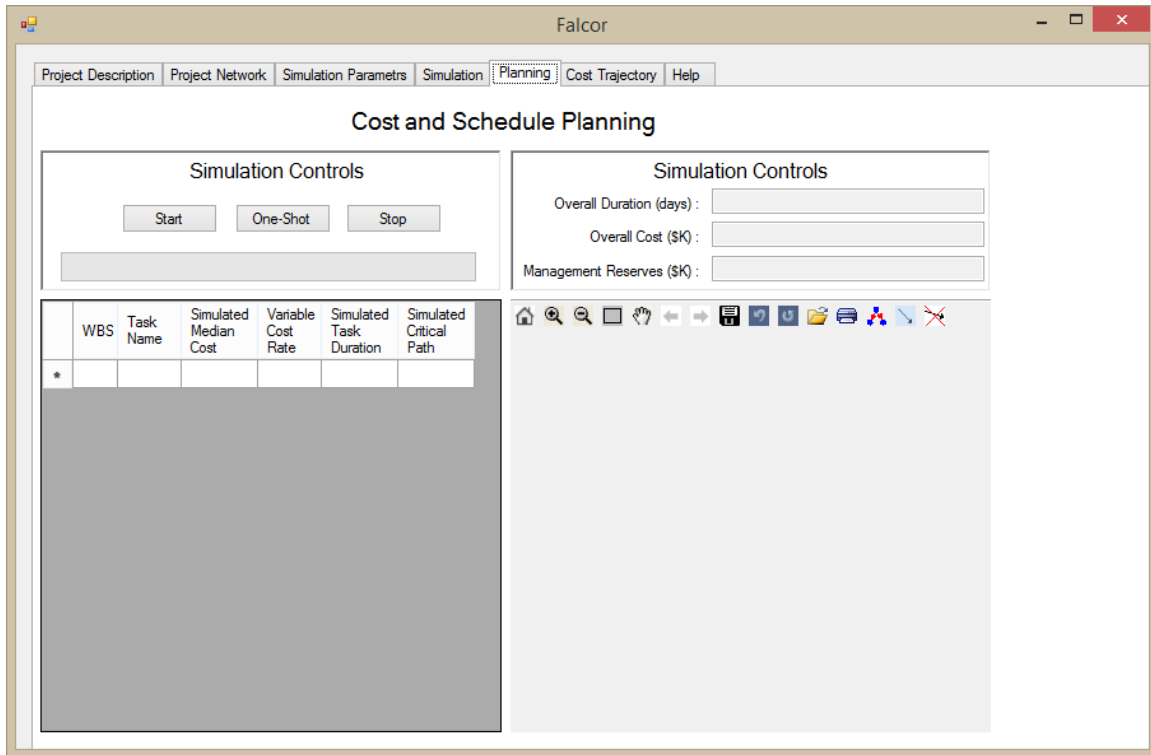


Figure 38. Software Application – Project Planning

Figure 39 shows the unimplemented tracking feature and cost/schedule phasing graphs which provides decision makers with early indications of budget and schedule risks. The stochastic nature of these projections naturally favors early estimates since the cost/schedule uncertainty ranges widen with increasing time. A user has the option to revise a project estimate to get a clearer picture of the near and far term uncertainties.

With regard to a stand-alone final implementation as envisioned and designed by Team Merica, certain functionality needs to be implemented prior achieving full capability. The ability to produce random numbers following a triangular distribution is present in software, however, the linking of the simulation manager to the network manager remains as future work in this delivery. The ability to produce report products has limited functionality in that dependency exists between the simulation execution and this functionality. Cost and schedule tracking also provides limited functionality, input into these data fields need to be tied to the Save/Edit Functionality to reach completion in this prototype. Critical path analysis needs to be implemented in this prototype iteration

as was accomplished in the DoE screening tool. This was a developmental challenge due to the original implementation, as this functionality disregarded complex network analysis.



Figure 39. Software Application – Project Cost Trajectory

F. PERFORMANCE AGAINST EXISTING TOOLS AND PROCESSES

Due to time constraints and the iterative process for HCD prototype development, not all features for this project were implemented within a single prototype activity. The product concept and demonstrated features, however, satisfy the intent of this study and represent significant progress in the field of project management.

Current industry tools are complex to use, sometimes requiring months of on the job training to obtain proficiency. This prototype software application maintained a level of user-conscious workflow and insulated the user from the complexities inherent in other products. Network diagram visualization is greatly enhanced from other tools, and statistical projections for the estimation of cost and schedule are merged in this concept in a way not currently satisfied in industry.

The data obtained from this product assumes a level of familiarity with statistics and an ability to infer risk levels based on programmatic decisions. The lack of this familiarity may cause apprehension for wide adoption, due to general unfamiliarity with the mathematical concepts. However, this hurdle can be managed through training and through increased exposure to the analysis methods. Finally, this product is capable of providing estimates only as good as the assumptions made by the user. While the triangular distribution was viewed by the participants of this study as the most intuitive way to represent this uncertainty, overly optimistic minimum and/or overly pessimistic maximum values will skew the results obtained from the output, thus limiting the utility of this software application.

G. CHAPTER SUMMARY

This chapter detailed the software and prototype development involved in this study. HCD principles were analyzed and explored in the creation of the proof of concept tool. Table 17 captures the resultant functional implementation achieved by the design team of this study.

Table 17. Prototype Software Application – Team Merica Functional Implementation

Functional Decomposition	Initial Prototype	Ideation Prototype		Final Implementation
		DoE Screening Tool	Story Board Mockup*	
Dynamic Project Network Display	NI	I	I	I
Simulation on Cost/Schedule Parameters	I	I	I	NI
Reporting Products	I	NI	I	LF
Critical Patch Analysis	NI	LF	I	NI
Cost/Schedule Tracking	I	LF	I	LF
Save/Edit Functionality	I	I	I	LF
Complex Project Network Analysis	NI	NI	I	I

* Design to Specification

Legend

Not Implemented	NI
Limited Functionality	LF
Implemented	I

Dynamic project network display was determined to be feasible as it was implemented in the final prototype. The application programming interface used to draw

and update these networks provided capabilities to color code sections of the project map that could be used to display critical paths along a project network. Within the display portion of the networks, complex project networks were able to analyzed and shown in structure to the planned project relation and was assessed to be feasible. Simulation for the cost and schedule parameters was also feasible, as it was implemented in all but the final prototype. Reporting products were found to be achievable provided the stakeholder can agree upon the acceptable format for the output. Critical path analysis, despite not being implemented in the final prototype, was found feasible as algorithms for their determination are widely proliferated and well established in the field of graph theory. Actual cost and schedule tracking was determined to be feasible, requiring periodic reporting and methods to overlay the actual incurred costs and update the cost/schedule projections. Being able to save and edit work created in the prototype software application is feasible and can be accomplished by storing data within a database or some other local file format.

The software engineering process and methodology was discussed and implementation of Scrum was justified given the structure and nature of this study. Documentation and source control principles and methodologies were established, and followed the best practices in industry. An analysis of alternatives for language selection was discussed and justified. For a copy of the source code developed for this study see the point of contact information in Appendix C.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. SUMMARY AND CONCLUSION

A. SUMMARY

This study has developed a proof-of-concept prototype software application that incorporates cost and schedule uncertainty within a project, which provides information required to the stakeholder about activity/task coordination on a more frequent basis than current practice, and allows the stakeholder insight into the performance of the project against cost and schedule.

Research questions were developed to understand current budget and schedule estimation practices of the stakeholder and the results are summarized below.

Research Question #1: What are the influencing factors and constraints of the current planning process?

Creating a project plan that can accurately estimate and track the cost and schedule requires significant inputs to provide some level of fidelity in project management. Additionally, managers currently rely heavily on SMEs and personal experience for both developing the tasks involved with projects, and for estimating the durations and costs of those tasks. These SMEs are critical in estimating the time and materials that are required to complete each activity of the project but the SMEs can be limited in their breadth of experience. For example, many practitioners have a lack of experience with probability distributions and uncertainty estimates for project activities. For new systems, the lack of SME experience can lead to additional program risk. This study found that most of the stakeholders interviewed performed their project planning and execution processes in separate software platforms (i.e., planning in MS Project and management in MS Excel) which make it difficult to change and/or modify project plans that are being executed. In most cases the project plans were developed using the SME's estimates of the duration and cost of the activities without using historical data of similar activities. Known tools and processes, such as establishing a project WBS and an EVM program, help the PM to track schedule and budget status. However, these methods are usually reserved for large projects (\geq \$20M) and are not part of a single integrated tool.

For larger projects, budgeting mandates the use of EVM, but for smaller projects, such costs are not included for tracking project status and can result in oversights. Another factor that limits the collaboration of program members is the inability to access the project planning files from various locations or even use the files concurrently. Many software applications in use today are tedious to use, which can result in delays of two weeks or more in monitoring and reporting project status.

Research Question #2: What are the required interfaces of the current cost and schedule planning system with focus on hardware, software, and human systems?

Currently, the interfaces of the cost and schedule planning systems with a focus on hardware, software and human systems are labor intensive and complicated due to multiple interdependencies. Most stakeholders use a hardware interface with Windows-based computers and compatibility with this type of computer was found to be necessary for the stakeholder to accept the prototype software application. Most stakeholders are currently using MS Project and MS Excel independently to perform the planning and tracking the execution of tasks within projects. These two software programs were determined to be desired software interfaces.

Following the principles of HCD, the primary users of the cost and budget planning systems were identified as the PMs, LEs, BFAs, and PEs. The human interfaces need to be specifically designed to meet each of their respective needs. For planners, developing a new project plan requires building a Gantt chart or network diagram to show the sequence and interaction of each project task. There is currently not an easy way to rearrange the sequence with the Gantt chart or network diagram in MS Project. The planner must tediously manipulate the schedule data using entry into the task list table. A graphical user interface allowing the visualization and manipulation of the task sequence is desired. The LEs and PEs help calculate the project baseline estimates so they frequently interface with the software to provide task estimates. The data entry for each task is cumbersome and most software applications only allow entry by one individual at a time. An interface with the capability of multiple simultaneous user access is a desired feature. PMs and BFAs primarily need to monitor project performance. The tracking system interfaces need to provide information about whether the project is on-

budget, on-schedule, and if contractual constraints are being met. It must do so in a clear, concise and timely manner to cue any needed management actions. Current methods require multiple human interfaces across several software tools and often result in time-late information passed to the PMs and BFAs which can affect time critical decisions.

Research Question #3: How are the stakeholders tracking and analyzing the progress of planned activities in a project?

For any program that is in progress, keeping the project schedule up-to-date is a challenge. Reporting has to be generated by multiple teams in different areas of the program and processed by the staff in charge of tracking the project performance. The project schedule and budget are then updated and verified before being releasing to the groups responsible for reviewing the baseline project plan. Once the baseline project plan is established, there is reluctance to change the plan.

A detailed uncertainty analysis is often not performed in order to identify potential sources of schedule and budget variations within the plan. When uncertainty is not adequately considered, there is an increased risk for underestimating the needed budget and schedule. Another risk of not considering uncertainty in the baseline project plans is that during the execution phase of the project, the PM might focus on decreasing the length of the expected critical path without realizing that there are other activities in the project that lie on alternative critical paths caused by variance in the activities cost and schedule. These alternative critical paths can negate the PM's schedule and cost-cutting mitigation efforts and may have significant impact to the overall budget and schedule. Even when uncertainty is considered in a baseline project plan, the variation of duration and cost of the activities are not taken into account when tracking the execution of the project activities. Individual project task duration and cost vary and may exceed the most likely values but still be within the normal variation expected. Some management actions may be taken to address apparent over-runs when it is not necessary because the task is still within the expected normal range. Only a few scheduling tools display the normal variance expected within a schedule baseline. Most display only a scalar value, typically the most likely or mean value entered by the planner. Displaying the normal variance within the schedule and when an individual task's expected variance

is exceeded during execution was determined to be necessary to cue necessary management action.

Research Question #4: What information would be considered valuable for depicting an overall picture of the cost and schedule estimates of the project under analysis?

Information gathered from research question two helped identify the information that the stakeholders found valuable for depicting the overall cost and schedule estimates. A Gantt chart or network diagram that shows the sequence and interaction of each project task and which also allows for quick rearrangement on the graphical interface was desired by the stakeholders. Quick views of task information and overall project budget and schedule status was also desired. A graphical display of the critical path and the near critical paths of the project would help the management staff avoid potential pitfalls from a delayed activity and reduce the chance of encountering undesirable situations. Identifying which activities are on the critical path or near critical paths would also allow the project team to determine which tasks would have a cascading effect and incur additional risk for the project if they go over cost or schedule. Incorporating data from similar projects would allow the development of more accurate estimates by utilizing a probability distribution that more closely resembles the actual work to be performed. After integrating uncertainty into the budgeting and planning process, information about which project activities have the greatest impact and contribute the greatest variation in cost and schedule of the project was also considered important. Displaying the normal variance within the schedule and individual task variance was also considered useful. Most stakeholders are familiar with EVM and having a graphical depiction of EVM would facilitate project member communications and help the PM manage project performance.

Research Question #5: How can the current process of planning and project management be modified to account for the uncertainty of cost and duration?

Based upon the observations from the previous research questions a prototype software application was developed by Team Merica to address many of the observed

pitfalls. Although not fully implemented in the final prototype, the concepts and methodologies of the prototype software application were found to be feasible. The envisioned application is capable of addressing many of the areas of concern identified by the stakeholders. The prototype software application uses the SME's estimate as a starting point, and then adds uncertainty along a selectable probability distribution based on how likely a task is to be completed on-budget and on-schedule. Planning using a Graphical User Interface (GUI) and allowing the importation of schedule data from MS Project are expected to improve the ability to build network diagrams and Gantt charts. The analysis with task uncertainties gives a more complete picture of the critical and near critical paths that could cause unplanned cost and schedule growth. It allows the users to forecast which activities are causing the greatest impact in the probability of completing the project on-budget and on-schedule. With this information the PM can adjust the resources of the project in order to correct a negative trend or speed up the progress of the project. The prototype software application allows all users of the project plan to enter the actual performance data (i.e., cost and duration) of each activity as the project progresses through the execution phase. This feature provides to the PM timely information that can be incorporated into updated forecasts of project cost and schedule, which can ultimately be used to adjust resources and communicate with the stakeholder as needed. Additionally, the prototype software application has the ability to display an information box for every activity that is selected by the user in the Network Design tab. The Activity Window lists all relevant information including notes that will capture any assumptions made during the development of the baseline project plan. During the project execution phase, these activities will display the actual data (i.e., cost and duration) entered by the PE along with any notes that communicate additional information related to the execution status of each particular activity. In order to identify quickly negative performance trends in the project execution phase, the prototype software application highlights those activities that are behind-schedule or over-budget.

Additional items that were planned for the prototype software application but were not completed were a database that could store activity models and the ability for the PM to generate color-coded project status reports, sensitivity analysis, and EVM

reports. The proposed application provides a single integrated solution which requires less time spent on training for the different users due to simpler interfaces to create, manage and modify baseline project plans and track execution performance.

B. FUTURE WORK

This section outlines some suggestions on future work to carry this prototype to a wider audience. As noted in Chapter V, the purpose of this project was a proof of concept to demonstrate the viability of a software product which could meet the goals of providing better estimates in the domains of programmatic cost and schedule. The design team has met this goal through the demonstration and development of a number of prototypes created through HCD methodologies. Due to time constraints, the design team was unable to produce a single, fully functional prototype capable of meeting the stakeholder's requirements. However, through iteration of many design phases and Scrum sprints, all core functionality was implemented within the context of a prototype subset (see Chapter V Section E).

Given this, future work can be viewed within two separate domains, (1) the completion of the current prototype and (2) follow on development, capturing more of the features desired by stakeholders. In the following sections the development team outlines the required steps to achieve full functionality for this latest prototype and provides suggestions for the future vision for items that were considered out of scope to the core functionality.

1. Final Prototype Development

A particular challenge encountered by developers was the display of network diagrams in a way which was intuitive and communicated the overall structure of the project which was being analyzed. The developers incorporated third party open source graphics libraries developed by Nachmanson (2015) as part of the MS Research Group to overcome these challenges. It became evident during the integration of this product to the final deliverable that these tools were themselves immature in terms of the functionality required for this application. A significant portion of the development time on the final

prototype centered about the integration and the modification of the MS Automatic Graph Layout Libraries.

Future development required in the display of these graphs include critical path color coding to dynamically display the sequence of work tasks which compose the critical path. To accomplish this, an algorithm which accounts for a nested project network needs to be developed. An elementary form of this algorithm is implemented in the DoE screening prototype, but it did not account for nested project networks. By incorporating simulation data one could color code run-specific critical paths which could be different from the planned path due to the introduction of uncertainty. Another desired feature in the display of networks is the ability to collapse and expand these diagrams to reduce visual overload to users that would occur in large network representations.

The simulation data aggregation is not present in the final prototype. Development of the proper statistical analysis would require aggregating cost and schedule data for each task, propagating these metrics to the parent subtask, and finally applying the results to the whole of the analysis. This feature would allow interested parties to draw conclusions at arbitrary levels within a project hierarchy to aid in estimation. This task is clearly processor intensive, and depending on the size of the network would require significant resources.

Progress tracking and updating baseline estimates would show how a given project or task is measuring up with estimated projections. This feature would allow early insight into poorly formed assumptions in activity cost and schedule baseline plans and allow planners to collect metrics on the accuracy of these estimates.

Finally, there was desire to import commonly executed subnetworks into a current task project. Incorporating this feature could reduce the time required to build up a network from scratch at every level and provide better estimates which are based on empirical execution instead of opinions of subject matter experts.

2. Future Vision

This project has potential in government and in industry to provide project managers and project planners the information required to do their jobs with greater accuracy. In the environment of distributed workforces, it would be desired to host this information as a web application. Separate user views based on program roles could be assigned thus presenting a user with only the information which they require in the execution of their duties.

Additional statistical analysis methods could be applied based on the application to better formulate alternative project plans. Sensitivity analysis and tornado diagrams could provide users with the capability to perform what-if analysis and find innovative alternatives with the intent of reducing programmatic risk in the cost/schedule spectrum.

This application, in the current design, only considers static cost which is unscaled for inflation or cost reduction due to increased technological capabilities. Having the simulations project future year dollars would be another feature that could aid in total project cost estimates particularly for projects operating with a long time horizon.

The potential applications and additions to this body of work could go much further than presented here. Taking the original objectives only as subset of Systems Engineering methodologies makes room for a system which could exist as a central and interconnected Systems Engineering repository that provides traceability to requirements schedule, and performance projections to better manage this complex discipline. Though no one tool exists that encompasses all of the project management phases (i.e., planning and execution), the feasibility of one may be a question of future research since much effort is taken to tailor a project and make the systems engineering effort appropriate to the scope of this study.

C. CONCLUSION

While a completely integrated, fully operational prototype was unable to be developed within the time span of this study, the component pieces and ultimate outcome of the study allows us to conclude that the tool, once completed, would allow for stakeholders to view and forecast a projects' status more accurately than existing

methods. The ability to obtain this information in readily understandable graphic displays provides decision-making stakeholders additional situational awareness into the projects they oversee. Early modeling and simulation scenarios of cost and schedule estimates with uncertainty confirmed suspicions that point estimates provided by more traditional project management software were unsatisfactorily inaccurate, varying from 35% over-estimated to 45% under-estimated. Further qualitative confirmation of these facts obtained from the stakeholder interviews resulted in the generation of the 118 requirements utilized to design the system. These requirements drove the development of our IDEF0 functional architecture, which in turn informed the design and development of our software prototype.

Many of the 118 requirements were system features and capabilities commonly requested by interviewed stakeholders that augmented or supported the core functionality intended for the system, but were not deemed of high enough priority to devote the teams' very limited resources for inclusion into the proof-of-concept prototype software application. While the team has included them within its functional design, there is currently no data to assess the effectiveness of those features within this report. The most notable requirements which were able to be integrated within the prototype include: the GUI, the Project Network Designer, the import/export from MS Project, an internal database using XML and a standard schema, and graphical chart outputs. Although the efforts of the team and the fruits of this report provide the necessary research, design, and early development groundwork, there still remains significant additional development work to bring the tool up to the level of quality expected by the stakeholders.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. STAKEHOLDER RESEARCH QUESTIONS

This appendix shows the questions that were used to conduct the stakeholder interviews. Chapter III outlines the requirements that were generated and derived from the output of these interviews.

Introduction:

We would like to ask you about your organizations current project planning and execution processes. More specifically, we would like to learn about your organizations current scheduling and cost estimating processes and how project progress throughout execution gets tracked.

Organizational information with focus on current processes:

- 1) Planning Process - What are influencing factors and constraints of the current planning processes?
 - a) What factors are considered in your organization project planning processes?
 - i) What are the inputs/outputs of the organization project planning process? Please provide examples.
 - ii) How is the input/output data received in the system and in what format?
 - iii) What tools are used to collect cost and schedule data needed to create project baselines during the project planning phase?
 - iv) Are these the same tools that are used during execution? If not, what are the execution tools that are used?
 - v) What tools are used to generate the cost and schedule baseline during the planning phase?
 - vi) What is the role of the person that provides the required data for the project planning process?

- vii) Can we have examples of completed projects with the original plans, schedule, tracking data (EVM or other methodology)? Could you please walk us through the complete example?
- b) In what way does the current project planning process work well? What are common challenges of the current project planning process and why is considered a challenge?
- c) What features make one planning system better than another?
- d) During the planning phase, how does the organization account for uncertainty in the estimated cost and schedule baseline?
- e) If uncertainty in the planning phase is accounted for using probability distributions, what statistic(s) does your organization uses in the planning process (Mode, Mean, Median, Other?)
- f) If uncertainty in the planning phase is not accounted for, but it does exist for a specific task, and it can be represented in a probability distribution of the cost and duration to complete such task, what number (statistic) would your organization use in their planning process?
- g) Do the current project management tools accept and accurately maintain uncertainties in planned project budget and schedules? Explain.
- i) How is the organization currently determining the levels of uncertainties for each individual task of the project in the planning phase? Who (what role or title but not name) normally performs this work?
- ii) What are the established planning phase parameters to determine if a project is not going to meet the planned budget and schedule? When would your organization take management actions? For example, for EVM, what would be the SPI and CPI that would trigger management action? For non-EVM, how much schedule deviation (percentage) or cost deviation (percentage) would trigger a management action?

2) Tracking Process - How are the stakeholders tracking and analyzing the progress of the planned activities in a project?

a) Are the same tools and inputs/outputs of the project planning phase used to track the actual progress of the project? Explain why or why not.

b) How does the organization track project progress against the established baseline in terms of cost and schedule?

c) How accurate have previous project baselines been when compared to their project progress?

i) What project progress data is used to report progress? Explain which format is used and why.

ii) What tools are used to track progress of the project? Why was this particular tool selected?

iii) What is the role of the person that collects the actual cost and/or schedule duration of the project? What is the role of the person that uses this information? What difficulties or hardships occur in the process of collecting the data?

iv) In what way is the tracking information used within the organization?

v) Provide an example of a time where progress was tracked and please walk us through the example.

d) What features make a project tracking system better than another?

e) During the execution phase, how does the organization account for uncertainty in the estimated cost and schedule baseline?

f) If uncertainty is accounted for in the execution phase using probability distributions, what statistic(s) does your organization use in the planning process (Mode, Mean, Median, Other?)

- g) If uncertainty is not accounted for during execution and uncertainty exists for a task, and the task cost or schedule duration is presented in a probability distribution, what number (statistic) would your organization use in their tracking process?
- h) Do the current execution project management tools allow and accurately maintain uncertainties? Explain.
- i) How is the organization tracking execution cost and schedule data during project execution? Who (what role or title but not name) normally performs this work?
- ii) What percentage of projects has your organization completed on time/schedule as defined by planned baseline schedule/plan?
- iii) How does your organization handle projects that are exceeding or deviating from the planned budget and schedule?
- iv) If your organization is routinely deviating from the baseline plans (cost and schedule), does it make adjustments in the plans of follow-on projects? Do you document lessons learned?
- 3) How could considering uncertainties on budgets and schedule help the organization complete projects on-budget and on-schedule?
- 4) If there were one or two things that you would recommend in a new planning and execution tool, what would it be?
- 5) When the conceptual model is ready for review, we will be asking,
- This model shows how we are thinking about redesigning an estimation tool for considering uncertainty in the planning and execution processes of a project.
- a) How would this model enhance the project cost and schedule estimates in the organization?
- b) How much time do you estimate that it will take the management staff to generate a budget and schedule plan with the prototype?

- c) How much time do you estimate that it will take to generate project progress reports with the prototype?
- d) What roles are needed for the successful operation of the model from project planning to execution?
- e) How does this model compare to other scheduling tools used in your organization?

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. STAKEHOLDER REQUIREMENTS

This appendix list the requirements that were generated as a result of the notes generated from stakeholder interviews. These notes were placed and organized in RealtimeBoard as discussed in Chapter III, and used to generate this list of requirements. The list is a combination of direct stakeholder requirements generated from the notes and derived requirements created by Team Merica.

Requirement Number	Description
Software Functionality	
SF1	The software application shall have the option to automatically link and update data from selected data files.
SF1.1	The software application shall allow the user to decompose higher level project activities into sub-activities.
SF1.2	The software application shall aggregate the activities into higher level project activities.
SF2	The software application shall allow the user to enter activity duration and cost data.
SF2.1	The software application shall be capable of selecting input files from Microsoft Project.
SF2.2	The software application shall be able to retrieve information from a database.
SF3	The software application shall generate a project cost and schedule forecast based on pertinent historical project information.
SF4	The software application shall have the capability of being deployed on multiple mobile platforms.

Requirement Number	Description
SF5	The software application shall allow the user to search and select applicable project activities from the database for reuse.
SF6	The software application shall allow the user to save schedule and cost network data for specific project(s).
SF7	The software application shall allow the user to search and track specific task metrics.
SF8	The software application shall provide the user the ability to select the form of output documents to be either Microsoft Word and/or Adobe PDF.
SF9	The software application shall identify and track any changes to project course after initiation.
SF10	The software application shall notify the user via error message of any problems occurring due to an error in linking, uploading, viewing, or storing data.
SF11	The software application shall have a help button available during use and this will connect the user with the appropriate resources for resolution.
SF11.1	The software application shall have documentation resources for training.
SF12	The software application shall be able to store all information in a database that is accessible by geographically dispersed users.
Network and User Access	
NR1	The software application data shall be accessible by multiple user(s). Threshold: one user at a time Objective: simultaneously
NR2	The software application output reports shall be accessible by multiple users.
NR3	The software application shall allow multiple users to input data.

Requirement Number	Description
NR3.1	The software application shall track users responsible for data changes.
NR3.2	The software application shall contain different levels of access rights i.e., read, write.
NR4	The software application shall allow the user to generate a report that is capable of being transmitted through email (threshold < 5 MB).
NR5	The software application shall allow usage from any location with network access.
NR6	The software application shall be compatible with Microsoft operating systems.
Project Description Data	
PR1	The software application shall allow the user to enter the magnitude of the budget management reserve.
PR2	The software application shall allow the user to enter and modify project description data.
PR3	The software application shall contain a project or task description field.
PR4	The software application shall allow the user to enter/change project start date.
PR5	The software application shall allow minimum and maximum project cost limitations.
PR6	The software application shall contain fields to identify key users involved in the project under analysis.
Project Planning Inputs	
PPIIn1	The software application shall display a Gantt Chart from the project network data.

Requirement Number	Description
PPIn2	The software application shall calculate the start and end date for each activity in the project network data.
PPIn3	The software application shall display the project network data in a network diagram format.
PPIn4	The software application shall allow the user to manually build network diagrams.
PPIn4.1	The software application shall allow the user to modify existing network diagrams.
PPIn4.2	The software application shall allow the user to enter project data for specific activities in the network diagrams.
PPIn5	The software application shall allow the user to enter/modify information for each project activity.
PPIn5.1	The software application shall allow the user to enter/modify activity name.
PPIn5.2	The software application shall allow the user to enter/modify activity cost.
PPIn5.2.1	The software application shall allow the user to select a probability distribution that closely models the cost of the project activity.
PPIn5.2.1.1	The software application shall allow the user to enter the distribution parameters for the cost of each activity in the project.
PPIn5.2.2	The software application shall allow the user to enter the variable cost for each activity in the project.
PPIn5.3	The software application shall allow the user to enter/modify activity duration.
PPIn5.3.1	The software application shall allow the user to select a probability distribution that closely models the duration of the project activity.

Requirement Number	Description
PPIn5.3.1.1	The software application shall allow the user to enter the distribution parameters for the duration of each activity in the project.
PPIn6	The user shall be presented with a questionnaire designed to assess the anticipated uncertainty of a particular task if the level of uncertainty for that task is unknown.
PPIn7	The software application shall accept activity/project cost information from different sources of data.
Project Planning Outputs	
PPOR1	The software application shall generate a cost and schedule forecast for the project.
PPOR2	The software application shall be able to display previously generated cost and schedule forecasts.
PPOR3	The software application shall generate a baseline project plan based on user inputs.
PPOR4	The software application shall display the critical paths of the project.
PPOR4.1	The software application shall display the critical paths of the project network as a color coded diagram.
PPOR4.2	The software application shall display the critical path of the project network in tabular form.
PPOR5	The planning report shall display the project description data.
PPOR6	The software application shall generate a probability cumulative distribution for the total schedule of the project.
PPOR7	The software application shall generate a probability cumulative distribution for the total cost of the project.

Requirement Number	Description
PPOR8	The planning report shall display the network diagram and/or table with the activity data for each task.
PPOR9	The planning report shall display Estimate At Completion (EAC) date and cost.
PPOR10	The software shall allow for manual schedule task duration overrides.
PPOR11	The planning report shall have the ability to display the activity data for each task which shall include:
PPOR11.1	Name of the task
PPOR11.2	Start date
PPOR11.3	Duration (simulation output)
PPOR11.4	Resource POC
PPOR11.5	Fixed Cost
PPOR11.6	Variable Cost
PPOR11.7	Probability of success
PPOR11.8	Estimated number of hours to complete task (simulation output)
PPOR12	The planning report shall display the simulation parameters.
PPOR13	The planning report shall display tabulated and/or graphical forecast of the cost and schedule of the project.
Activity and Project Models	
AMR1	The software application shall allow the user to auto-populate the information of the project activity using a model built from historical project data.
AMR2	The software application shall allow the user to generate activity models.
AMR3	The software application shall be able to access the database which contains activity models from previous projects.

Requirement Number	Description
AMR4	The software application shall allow the user to update the activity models from previous projects.
Simulation and Analysis	
SimR1	The Software simulation analysis shall consider the median time to complete each activity to categorize the different paths of the project network diagram by level of criticality.
SimR2	The software application shall allow the user to vary the number of simulation runs.
SimR3	The software application shall allow the user to modify the seed value of the simulation.
SimR4	The software application shall perform a Monte Carlo analysis using the project data entered by the user to determine critical paths, median cost and schedule of the project under analysis.
SimR5	The software application shall display the total project cost for each of the simulation runs in a histogram format.
SimR6	The software application shall display the total project duration for each of the simulation runs in a histogram format.
Project Execution	
PEXR1	The software application shall allow users to enter actual activity performance data.
PEXR1.1	The software application shall allow users to only enter actual performance data for tasks they are specifically assigned.
PEXR1.2	The software application shall allow geographically dispersed users to review and update tasks.
PEXR2	The software application shall display the cost and schedule trajectory based on the actual data entered by the user.

Requirement Number	Description
PExR3	At the start of project execution the software application shall use the baseline project plan as an input.
PExR4	The software application shall be scalable based on the size and duration of the project.
PExR5	The software application shall generate Earned Value (EV) and display in both textual and graphical format
PExR5.1	The software application shall calculate the Budgeted Cost of Work Schedule (BCWS).
PExR5.2	The software application shall calculate the Actual Cost of Work Performed (ACWP).
PExR5.3	The software application shall calculate the Cost Performance index (CPI).
PExR5.4	The software application shall calculate the Schedule Performance index (SPI).
PExR5.5	The software application shall generate the activity schedule from the user input project data.
PExR6	The software application shall calculate and display, based on user entered performance data, actual vs planned work performed in both a textual and graphical format.
PExR7	The software application shall have a dashboard that displays and allows selection of user generated and assigned projects.
PExR8	The software application shall generate a one page printable summary of project status.
PExR9	As actual performance data is inserted into the software application it shall display if the project and specific tasks are ahead, on, or behind schedule via text and color coding.

Requirement Number	Description
PExR10	The software application shall allow users to enter comments for a specific task.
PExR11	The software shall display a notification if the critical path changes due to actual performance.
PExR12	The software shall allow the baseline project cost/schedule to be updated with actual data during the execution phase.
PExR13	The planning report shall provide actual work performed (hours), hours available per task.
Error Handling	
ERR1	The software application shall not allow the user to run a Software Simulation Analysis until all the information of the activities have been entered
ERR2	The software application shall not allow the user to enter incorrect data types into fields.
ERR2.1	If a data entry error is present the software application shall not allow selection of “Run Simulation.”
ERR2.2	The software application shall reject a duration value of zero for activities.
ERR2.3	The software application shall display an error if division by zero is encountered.
ERR2.4	The software application shall display an error message if a user attempts to enter a letter or symbol in a number field.
Usability	
UR1	The software application shall display tooltip descriptions of the required inputs.
UR2	The software application shall provide default values for information that is not currently known.

Requirement Number	Description
UR2.1	The software application shall alert the user that the generated forecast may contain low fidelity estimated values.
UR3	The generated reports shall display the project information that was used to generate them.
UR4	The software application shall allow the user to modify the activities from the project network diagram.
UR5	The software application shall allow the user to modify the data of the activity from the generated Gantt Chart of the project.
UR6	The software application shall allow the user to add/remove activities of the project.
UR7	The software application shall allow the user to generate different types of reports.
UR8	The software application shall provide to the user the ability to enter the project data by prompting for the required information.
UR9	Packaging the project documentation for portability
UR10	The user shall be able to update the display of cost and schedule trajectory with a single button.

APPENDIX C. SOFTWARE SOURCE CODE

For a copy of the prototype software application, please contact research advisor, Professor Mark M. Rhoades (mmrhoade@nps.edu).

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

- Buede, Dennis M. 2009. *The Engineering Design of Systems: Models and Methods 2nd Edition*. Hoboken, New Jersey: John Wiley and Sons.
- Hillson, David. 2004. "Earned Value Management and Risk Management: A Practical Synergy." Paper presented at the annual Project Management Institute (PMI) Global Congress, Anaheim, California.
- Hulett, David T. 2004. "Integrated Cost and Schedule Project Risk Analysis." Paper presented at the annual Project Management Institute (PMI) Global Congress, Prague, Czech Republic.
- . 1996 "Schedule Risk Analysis Simplified." *PM Network*: 23–30.
- Humphreys & Associates. 2012. "EVMS Education Center: Basic Concepts of Earned Value Management (EVM)" Irvine, California: Humphreys & Associates Inc. <http://www.humphreys-assoc.com/evms/basic-concepts-earned-value-management-evm-ta-a-74.html>.
- Law, Averill M. 2007. *Simulation Modeling and Analysis*. 4th ed. New York: McGraw-Hill.
- MITRE. 2014. *Systems Engineering Guide*. Bedford, Massachusetts: The MITRE Corporation.
- Nachmanson, Lev. 2015. "Microsoft Automatic Graph Layout" Microsoft Research Group. <http://research.microsoft.com/en-us/projects/msagl/>.
- Pritchard, Carl L. 2015. *Risk Management: Concepts and Guidance*. 5th ed. Boca Raton, Florida: CRC Press Taylor & Francis Group.
- RealtimeBoard. 2015. www.realtimeboard.com. RealtimeBoard.
- Suter, Richard. 2006. "Measuring Uncertainty in Earned Value." Paper presented at the d Annual Acquisition Research Symposium of the Naval Postgraduate School, Monterey, California: Naval Postgraduate School.
- Under Secretary of Defense (AT&L). 1999. *Earned Value Management Support to System Acquisition Program Managers*, Audit Report No. 99-216. Arlington, Virginia: U.S. DOD Office of the Inspector General.
- U.S. Air Force. 2007. *Air Force Cost Risk and Uncertainty Analysis Handbook*. Hansom AFB, Massachusetts: U.S. Air Force Cost Analysis Agency.

- U.S. Defense Acquisition University. 2001. *Systems Engineering Fundamentals*. Fort Belvoir, Virginia: Defense Acquisition University Press.
- . 2015. S.v. “DAU Gold Card.” Last Modified July 6
<https://acc.dau.mil/CommunityBrowser.aspx?id=19577>.
- U.S. Department of Defense. 2014. *Department of Defense Risk Management Guide for Defense Acquisition Programs, 7th Edition*. Washington, DC: Office of the Deputy Assistant Secretary of Defense for Systems Engineering.
- U.S. Government Accountability Office (GAO). 2009. *GAO Cost Estimating and Assessment Guide: Best Practices for Developing and Managing Capital Program Costs* GAO-09-3SP. Washington, DC: U.S. Government Accountability Office.
- Vose, David. 2008. *Risk Analysis: A Quantitative Guide*. 3rd ed. Hoboken, New Jersey: John Wiley & Sons.
- Webb, Alan. 2003. *The Project Manager’s Guide to Handling Risk*. Burlington, Virginia: Gower.
- Williams, A.J. 2013. “5 Benefits of Integrate Project Estimating.” Project Management Block, MSS Projects Analytics:
<http://projectmanagementblog.mssprojectsanalytics.com/5-benefits-of-integrated-project-estimating/>.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California