

# **An Integrated Model-Based Approach to Improving Project Control in Department of Defense Acquisition**

by

**Christopher E. Carson**

B.S. Mechanical Engineering, University of Saint Thomas, 2007  
Master of Business Administration, Weber State University, 2011

Submitted to the System Design and Management Program  
in Partial Fulfillment of the Requirements for the Degree of

**Master of Science in Engineering and Management**

at the

Massachusetts Institute of Technology

February 2021

© 2021 Christopher E. Carson  
All rights reserved

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Signature of Author \_\_\_\_\_

Christopher E. Carson  
System Design and Management Program  
January 25, 2021

Certified by \_\_\_\_\_

Bryan R. Moser, Ph.D.  
Thesis Supervisor  
Academic Director and Senior Lecturer, MIT System Design and Management

Accepted by \_\_\_\_\_

Joan Rubin  
Executive Director, System Design and Management Program

THIS PAGE INTENTIONALLY LEFT BLANK

**Disclaimer:**

The views expressed in this document are those of the author and do not reflect the official position or policies of the United States Air Force, Department of Defense, or Government.

THIS PAGE INTENTIONALLY LEFT BLANK

# **An Integrated Model-Based Approach to Improving Project Control in Department of Defense Acquisition**

by

**Christopher E. Carson**

Submitted to the System Design and Management Program on January 25th, 2021  
in Partial Fulfillment of the Requirements for the Degree of  
Master of Science in Engineering and Management

## **ABSTRACT**

The United States no longer has the luxury of overspending on military weapon systems. Military programs have steadily cost more, taken longer, and delivered less. How can the Department of Defense reverse this trend? The Department of Defense prescribes the use of an Earned Value Management System (EVMS) to control large, complex engineering projects. According to academic literature, the earned value method can be an effective project control technique but also has significant flaws. Modern integrated project models allow for innovative new approaches to project control which may be superior to the earned value method. Department of Defense policy reveals that integrating cost, schedule, and scope; accurately forecasting project status to allow for proactive decision making; and effective risk mitigation are the most important features of a project control method. This thesis reviews earned value method research and Department of Defense EVMS policy. This thesis also evaluates four project control methods through an experiment that uses an integrated project model. Subject to the specific conditions represented in the model, a Multiple Risk Level model-based control method enabled more proactive decision making than a modified version of the earned value method in the experiment. However, the Multiple Risk Level model did not forecast or enable risk mitigation as well as the modified earned value method in the experiment. The results of this analysis suggest that the ideal project control technique depends on the goals, nature, and environment of the project. Therefore, the Department of Defense should use integrated project models to tailor project control strategies to best suit acquisition programs.

Thesis Supervisor: Bryan R. Moser

Title: Academic Director & Senior Lecturer, MIT System Design and Management

THIS PAGE INTENTIONALLY LEFT BLANK

## **Acknowledgements**

I would like to thank the Air Force and SAF/AQ for granting me the privilege and opportunity to complete this program. Thank you to the mentors and peers from whom I have gained over the years. Thanks to the fellows in my cohort for their friendship, experience, and wisdom. I am truly the better for it. Thank you to the SDM faculty and staff for their tireless work in making this experience rewarding, exciting, and enlightening.

A huge thank you to my advisor, Bryan Moser, for his patience and guidance throughout this thesis journey. I have learned much from him in this brief period of time. I look forward to leveraging those lessons in my future endeavors.

I am forever grateful for my son, Vincent. He is my hope and inspiration. A deep appreciation also to my mother and father, Sandy and Steve, sisters Katlyn and Gina, and brother Markus.

Most importantly, I thank my wife, Lauren, for her love and support. I can only begin to express my gratitude for her patience and understanding through the many long days, nights, and weekends. It would not have been possible to complete this curriculum without her sacrifice, hard work, advice, and encouragement.

THIS PAGE INTENTIONALLY LEFT BLANK



## Table of Contents

<b>ABSTRACT .....</b>	<b>5</b>
<b>1 INTRODUCTION .....</b>	<b>16</b>
1.1 Motivation .....	16
1.2 Scope .....	18
1.3 Objective .....	20
1.4 Thesis Structure.....	21
<b>2 BACKGROUND.....</b>	<b>22</b>
2.1 Project as System .....	22
2.2 Modeling to Understand the Project System.....	24
2.3 Systems Analysis Methodologies .....	27
2.3.1 Systems Thinking and System Problem Statements.....	27
2.3.2 Design Structure Matrix.....	27
2.3.3 Object-Process Methodology.....	28
2.3.4 Project-Product Lifecycle Management Framework .....	31
2.3.5 Project Design.....	32
2.4 Project Management Process .....	34
2.5 Project Control Process .....	37
2.6 DoD Acquisition Process as System .....	40
2.7 Project Control in DoD Acquisitions .....	43
2.8 Literature Review: Earned Value Method .....	47
<b>3 RESEARCH QUESTION AND APPROACH.....</b>	<b>50</b>
3.1 Project Control Needs .....	50
3.1.1 Hopes .....	55
3.1.2 Targets.....	57
3.1.3 Truths .....	58
3.2 Research Question.....	59
3.3 Hypotheses .....	61
3.4 Research Approach .....	61
3.4.1 Integrated Project Model.....	61
3.4.2 Experiment.....	62
3.4.3 Experimental Design.....	62
<b>4 MODELING AND SIMULATION.....</b>	<b>64</b>
4.1 Development of an Integrated Project Model .....	64
4.1.1 Model Scope and Concepts.....	64
4.1.2 Modules and Functions .....	72
4.1.3 Model Validation .....	78
4.2 Experimental Simulations .....	84
<b>5 RESULTS.....</b>	<b>86</b>
5.1 No Control.....	88
5.2 Simple Control .....	90
5.3 Earned Value Management with Earned Schedule .....	92

5.4	Multiple Risk Levels .....	94
5.5	Control Method Result Comparisons .....	96
<b>6</b>	<b>ANALYSIS AND DISCUSSION .....</b>	<b>100</b>
6.1	Evaluation of Methods: Basis of Value.....	100
6.2	Effect of Experimental Variables.....	101
6.3	Method Comparisons .....	104
6.3.1	No Control .....	104
6.3.2	Simple Control.....	104
6.3.3	Earned Value Management with Earned Schedule.....	105
6.3.4	Multiple Risk Levels.....	106
<b>7</b>	<b>CONCLUSION AND RECOMMENDATIONS.....</b>	<b>110</b>
7.1	Conclusion.....	110
7.2	Recommendation 1: Explore Innovative Project Control Methods .....	111
7.3	Recommendation 2: Eliminate Prescriptive EVMS Policy.....	111
7.4	Recommendation 3: Implement Integrated Project Models on DoD Programs.....	111
7.5	Recommendation 4: Create a Project Control Selection Tool .....	112
7.6	Recommendation 5: Publish a Digital Project Management Strategy .....	113
<b>8</b>	<b>FUTURE WORK.....</b>	<b>114</b>
8.1	The Value of Integrated Models.....	114
8.2	Limitations .....	114
8.3	Additional Experiments.....	116
	<b>APPENDIX A: Acronyms.....</b>	<b>118</b>
	<b>APPENDIX B: Computational Model Documentation.....</b>	<b>119</b>
	<b>APPENDIX C: Validation Trial &amp; Base Model Settings.....</b>	<b>131</b>
	<b>APPENDIX D: Integrated Project Model Code.....</b>	<b>140</b>
	<b>BIBLIOGRAPHY .....</b>	<b>193</b>

## List of Figures

Figure 1: "Periodic table" of DSMs and DMMs, forming an MDM (adapted from [60]) .....	28
Figure 2: OPM diagram of a generic engineering project .....	29
Figure 3: OPM diagram of generic system problem statement (adapted from [61]) .....	29
Figure 4: OPM model of a generic project construct (reprinted from [65]) .....	31
Figure 5: Object WBS Example (reproduced from [65]) .....	32
Figure 6: Project Design element relationships (reproduced from [19]) .....	33
Figure 7: Project Management Process Groups supporting the Project Life Cycle (reproduced from [16]) .....	34
Figure 8: Project Life Cycle (reproduced from [66]).....	35
Figure 9: Generic project life cycle (ISO/IEC/IEEE 15288:2015) (reproduced from [26]) .....	35
Figure 10: Department of Defense Adaptive Acquisition Framework Pathways (reproduced from [67]) .....	35
Figure 11: NASA Project Life-Cycle Phases (reproduced from [68]).....	36
Figure 12: Project Control Feedback Loop (reproduced from [27]).....	37
Figure 13: Control in project management phases (reproduced from [28]).....	38
Figure 14: Monitoring and Controlling Process Group – detailed view (reproduced from [16]) .....	38
Figure 15: OPM diagram of Defense Acquisition System system problem statement .....	41
Figure 16: OPM diagram of EVMS within the Defense Acquisition System .....	51
Figure 17: Abstract OPM diagram of a Project Control Process .....	52
Figure 18: OPM Diagram of Proposed DoD Project Control System Problem Statement .....	55
Figure 19: Integrated Project Model Architecture .....	64
Figure 20: Task, Team, Resource, and Scope Item Vector Architecture .....	65
Figure 21: Scope and Task Planning OPM Diagram .....	73
Figure 22: Task Executing OPM Diagram .....	75
Figure 23: Validation Trial Results with No Edge Cases .....	82
Figure 24: Validation Trial Results with Edge Cases .....	82
Figure 25: Reference Cases .....	83
Figure 26: Composite Project Score Regions .....	87
Figure 27: No Control Scatter Plot - Scope Targets Met by Cost and Time.....	88
Figure 28: No Control - Median Project Score vs Combined Settings .....	89
Figure 29: No Control Contour Plot – Project Score by Accuracy and Uncertainty .....	89
Figure 30: Simple Control Scatter Plot - Scope Targets Met by Cost and Time .....	90
Figure 31: Simple Control Plot - Median Project Score vs Combined Settings .....	91
Figure 32: Simple Control Plot - Median Project Score vs Latency Settings .....	91
Figure 33: EVM-ES Control Scatter Plot - Scope Targets Met by Cost and Time.....	92
Figure 34: EVM-ES Control Plot - Median Project Score vs. Combined Settings.....	93
Figure 35: EVM-ES Control Plot - Median Project Score vs Latency Settings .....	93
Figure 36: MRL Control Scatter Plot - Scope Targets Met by Cost and Time.....	94
Figure 37: MRL Control Contour Plot – Project Score by Accuracy and Uncertainty.....	95
Figure 38: MRL Control Plot - Median Project Score vs. Combined Settings .....	95
Figure 39: Combined Scatter Plot - Scope Targets by Cost and Duration.....	97
Figure 40: Box Plot - Project Score by Method.....	98
Figure 41: Scatter Plot: Time vs. Cost by Method and Uncertainty .....	98
Figure 42: Bar Plot - Scope Targets Met by Combined Settings.....	98
Figure 43: Bar Plot - Scope Targets Met by Method.....	98
Figure 44: Contour Plot –Project Score by Uncertainty and Latency (All Methods) .....	101
Figure 45: Contour Plot -- Project Score by Accuracy and Latency (All Methods) .....	102
Figure 46: Contour Plot - Project Score by Accuracy and Latency (All Methods) .....	102
Figure 47: Box Plot - Project Score by Method, No Uncertainty, Low Accuracy.....	103
Figure 48: Box Plot - Project Score by Method and Uncertainty .....	106
Figure 49: Pie Plots - Scope Targets Met by Method and Uncertainty.....	107
Figure 50: Pie Plots - Scope Targets Met by Method and Accuracy .....	108

Figure 51: MRL Scatter Plot - Cost versus Time by Accuracy .....108

## List of Tables

Table 1: OPM Symbols as Defined in ISO 19450 (Reproduced from [64]).....	30
Table 2: The Basic Project Management Life Cycle (reproduced from [28]) .....	35
Table 3: DoD EVM Application Requirements (reproduced from [76]).....	43
Table 4: DoD EVM Reporting Requirements (reproduced from [77]).....	43
Table 5: ANSI/EIA-748 EVMS Guidelines [79].....	44
Table 6: DoD Reasons for EVMS .....	46
Table 7: DoD Goals for EVMS grouped by value process .....	53
Table 8: Design of Experiments .....	63
Table 9: Project Model Risk Descriptions .....	68
Table 10: Project Multidomain Matrix .....	71
Table 11: Control Activity Taken by Control Methods.....	77
Table 12: Experimental Variable Settings .....	79
Table 13: Validation Trial Settings.....	80
Table 14: Validation Trials Summary .....	81
Table 15: Reference Case Variable Settings.....	83
Table 16: Experiment Simulations .....	85
Table 17: Example Confusion Matrix .....	87
Table 18: No Control Confusion Matrix.....	89
Table 19: Simple Control Confusion Matrix .....	91
Table 20: EVM-ES Confusion Matrix.....	93
Table 21: MRL Confusion Matrix .....	95
Table 22: Summary of Outcome Trial Result Statistics by Method .....	96
Table 23: Summary of Forecast Trial Result Statistics by Method .....	96
Table 24: Number of Project Trials Resulting in Unmet Scope Targets by Cancellation Outcome .....	98
Table 25: Control Method Ranking by Evaluation Criteria.....	100
Table 26: Statistics for no uncertainty, low accuracy trials .....	103
Table 27: Comparison of EVM and MRL results.....	110
Table 28: Base Scope Item List .....	120
Table 29: Base Product Breakdown Structure .....	120
Table 30: Base Task List .....	121
Table 31: Base Work Breakdown Structure .....	122
Table 32: Base Scope Dependency Structure Matrix .....	123
Table 33: Base Task Dependency Structure Matrix .....	124
Table 34: Base Scope to Task Domain Mapping Matrix.....	125
Table 35: Scope Item Vector Structure.....	126
Table 36: Task Vector Structure .....	127
Table 37: Human vector structure .....	127
Table 38: Resource vector structure .....	128
Table 39: Validation Settings - Low Scope Quantities.....	131
Table 40: Validation Settings - Moderate Scope Quantities.....	131
Table 41: Validation Settings - High Scope Quantities .....	132
Table 42: Validation Settings - Humans per Task .....	132
Table 43: Validation Settings - Resource Requirements per Task .....	132
Table 44: Validation Settings - Resource Requirements per Scope Item .....	132
Table 45: Validation Settings - Flat Scope DSM.....	133
Table 46: Validation Settings - Flat Task DSM.....	134
Table 47: Validation Settings - Full Network Scope DSM .....	135
Table 48: Validation Settings - Full Network Task DSM .....	136
Table 49: Validation Settings - Edge Case Team Sizes.....	136
Table 50: Validation Settings - Edge Case Task Resource Requirements.....	136

Table 51: Validation Settings - Edge Case Scope Item Resource Requirements .....	136
Table 52: Validation Settings - Edge Case Scope Quality Targets.....	137
Table 53: Validation Settings: Extreme-Low Scope List .....	137
Table 54: Validation Settings - Extreme-High Scope List .....	138
Table 55: Validation Settings - Quality Targets and Experimental Variables .....	139

THIS PAGE INTENTIONALLY LEFT BLANK

# I INTRODUCTION

This thesis attempts to evaluate specific project control methods using an integrated project model developed specifically for that purpose. Throughout the course of that effort, systems thinking is applied to better understand projects, project models, and project management processes. This thesis specifically analyzes the project control process as a system and attempts to represent it at a meaningful level of abstraction. Formally studying the process as a system allows for a better understanding of how the process works internally and with other processes to create emergent outcomes.

## I.1 Motivation

---

*“We must use creative approaches, make sustained investment, and be disciplined in execution to field a Joint Force fit for our time, one that can compete, deter, and win in this increasingly complex security environment.”*

*~ Jim Mattis [1]*

---

The United States military’s need for capability is greater than it has been for the last 30 years. [1] [2] The urgency of that need is also greater than at any time in the recent past as a result of the resurgence of great power competition. [1] [2] Meanwhile, the ability and appetite of the U.S. government to provide funding is limited. Funding is necessary to produce the systems that provide needed capability. Funding is limited because of the need to provide fiscal support to the domestic economy in order to address the economic crisis caused by the COVID-19 pandemic. [3] [4] Congress also appears unlikely to provide sufficient defense funding for these capability needs because the U.S. is entering a political climate reminiscent of the period which produced the Budget Control Act of 2011. When considering the confluence of these factors, defense programs must execute efficiently to provide as much capability as rapidly and as affordably as possible.

Defense programs have historically struggled to contain costs and deliver products which meet requirements within acceptable timeframes. [5] [6] [7] The Department of Defense has implemented several initiatives over the past 30 years aimed at correcting these issues with limited success. [8] [7] For example, the acquisition reform efforts of the early 2000s were meant to leverage the best practices from corporate America and reduce cost growth. [8] The reform was unsuccessful and resulted in increased cost growth in some circumstances. [8] The Better Buying Power series of initiatives have been aimed at reducing the cost of defense programs by managing cost risk and capitalizing on opportunities. [7] [9] In 2018, the Department of Defense published a Digital Engineering Strategy with the purpose of achieving “greater performance and affordability to meet current and future challenges.” [10] Most recently, the Department of Defense has embarked on a wholesale rewrite of the policy documents governing defense acquisitions, collectively known as the “5000 series,” in response to new methods authorized by Congress. [11] There is some preliminary evidence to suggest



that these latest series of initiatives may be having some success in reducing cost growth and ensuring programs deliver within acceptable timeframes. [11]

Achieving better outcomes in terms of cost, schedule, and technical performance requires additional change in the way that the Department of Defense conducts engineering projects. The preliminary results following dramatic changes in the Defense Acquisition System and the implementation of a Digital Engineering Strategy are encouraging, but more can be done. The initiatives mentioned above are attempts to change the way that engineering projects are initiated, planned, managed, and executed. The way that defense engineering projects are controlled has remained largely unchanged for several decades, however. [12] This thesis is motivated by the potential to improve defense engineering project outcomes by changing the way that they are controlled.

The Department of Defense acquisition community commonly refers to programs while the project management community commonly refers to projects. Programs are a set of “related projects, subsidiary programs, and program activities managed in a coordinated manner to obtain benefits not available from managing them individually.” [13] In systems thinking terms, programs are sets of related projects managed together in a way that produces emergent benefits. These emergent benefits may result from economies of scope or scale, or they may result from interrelationships in the functionality of the product systems. The product of one project may enable or be dependent upon the product of another project, and in such a case the benefit of managing the projects within a program would be to ensure they are fielded in conjunction.

The Department of Defense cites capability needs as the basis for projects. [11] In systems thinking terms, capability needs are the functional requirements for a system of systems. Several systems are operated in conjunction and their functions combine in complex ways to provide a military capability as an emergent behavior. Viewing the products resulting from projects as the systems which perform interdependent functions reveals the mechanism by which acquisition programs provide military capabilities. If it is possible to better control the constituent acquisition projects, the resulting product systems will field at higher quality, more rapidly, and/or more affordably. This will lead to better military capabilities which are available to the warfighter faster.

## 1.2 Scope

The scope of this thesis is confined to the evaluation of methods that may improve Department of Defense acquisition project control outcomes. Project control is inseparable from the broader project management process, and so this thesis will consider the broader project management process with the aim of understanding the role of project control in context. This thesis will adopt the Project Management Institute definition of a project as “a temporary endeavor undertaken to create a unique product, service, or result.” [13] For the sake of brevity, this thesis will refer to the intended outcome of a project as a *product* while acknowledging that this term does not perfectly suit all projects. There are several types of projects, and this thesis is limited to considering *novel engineering* projects. Repetitive activities such as production or operations are beyond scope, as are projects related to non-engineering disciplines.

Project management is the set of activities conducted in pursuit of a successful product. Project control is the subset of project management activities conducted with the aim of adjusting project parameters in response to or anticipation of project performance.

This thesis will consider project performance in terms of the three classic elements of the “triple constraint”: scope, cost, and time. [14] Scope is the result of project work and includes both intermediate (necessary but not externally valued) and target (valued by the customer) output. [15] Scope also includes flawed output that may be reworked, discarded, or included in the final product. The Guide to the Project Management Body of Knowledge includes quality as a fourth factor in defining project success, but this thesis will consider quality as a measure of the state of a scope item’s ability to meet a specification. [15] [16] Other factors have been proposed for evaluating projects such as risk, benefits, and resources. [17] Each of these can be decomposed and ultimately translated into one of the three basic elements of scope, cost, and time.

This thesis applies the principles of systems thinking to project activities. The project is viewed as a system. The product which is the target outcome of the project is also viewed as a system. The project management process is viewed as a system, and finally the project control process is viewed as a system. Each of these systems are interconnected and influence one another in ways that are not obvious or fully understood. This thesis will consider each of these systems and more but will focus primarily on the project control process.

Office of Management and Budget policy directs the use of earned value management as the principal project control method for federal executive agencies. [18] This thesis will review the Department of Defense application of this policy on projects and explore the potential for alternative methods in light of a recent shift to embrace innovative and disruptive approaches to acquisition.

This thesis will model a notional defense engineering project to evaluate the effectiveness of different project control methods. The model is purely theoretic and is not based on empirical data. The embodied principles, assumptions, and limitations of the model will be clearly stated. The evaluation of different control methods will be focused broadly on the relative effectiveness of the control methods and will not attempt to explain in detail the mechanisms causing any differences.

Engineering projects are complex sociotechnical systems composed of teams of humans in an organization, product scope in various stages of being realized, and processes which ultimately result in product scope. [19] The model which forms the basis of analysis for this thesis emphasizes the product and process elements of the project while omitting many of the organizational and human elements. The organizational and human elements of project behavior are not only highly complex but also the subject of robust debate. [20] [21] [22] [23] [24] [25] These factors are only briefly considered within the model. Organizational and team elements are nonetheless critical to a comprehensive accounting of project behavior and should be addressed in future work.

Project control is tightly linked with several other project management processes. The most closely associated processes are performance measurement and planning. It is beyond the scope of this thesis to evaluate the potential interactions between the different planning or measurement methods and the selected project control methods, but it is also acknowledged that such evaluation would be interesting and valuable. This is a potential topic for future work.

### **1.3 Objective**

At the beginning of its formulation, the objective of this thesis was to evaluate the potential for model-based project management to improve traditional Department of Defense project control methods. As the research process progressed, the objective has become increasingly focused. The topic of model-based project management narrowed to integrated project modeling. The Earned Value Method was selected as the baseline project control method for comparison based on its prescribed use in large Department of Defense acquisition projects. The criteria for improving project control are defined in terms of value derived from Department of Defense project control policy. These criteria are established in Section 3.1 using systems thinking methods.

The revised objective of this thesis is thus:

*To evaluate the potential to improve Department of Defense project control by comparing alternative methods with earned value management using an integrated project model.*

## **1.4 Thesis Structure**

This thesis is divided into eight chapters and four appendices. The first chapter introduces the motivation, scope, and objective of the thesis. It explains the need for improved defense acquisition project outcomes and how the project control process can contribute to those improvements.

The second chapter provides background on several topics and concepts critical to the remainder of the thesis. It applies a systems perspective to projects, project management processes in general, and project control in particular. It also contains an overview of several modeling and analytical approaches which are useful for studying project systems. Finally, it reviews the current state of research on earned value.

The third chapter explains the contribution that this thesis makes to project control research. It begins with a stakeholder needs analysis necessary to perform the research. It also states the research question, hypotheses and research approach.

The fourth chapter covers the integrated project model developed as part of this thesis. It details the concepts included and excluded from the model, how the model functions, and how the model was validated. It also covers the experiment conducted to evaluate four selected project control methods. It gives the experimental design and explains how the experiment was conducted.

The fifth chapter contains the results of the experiment. The sixth chapter analyzes those results and compares the performance of the four control methods. The seventh chapter lists several recommendations based on the insights gained from the model and experiment. The eighth chapter lists the various efforts related to this thesis that warrant additional work. These efforts should be the basis of future research.

The first appendix lists the acronyms used in this thesis. The second appendix provides additional documentation on the integrated project model. The third appendix contains information on the model settings used for the validation and experimental trials. Finally, the fourth appendix contains the complete code for the integrated project model used for the experiment.

## 2 BACKGROUND

### 2.1 Project as System

What is an engineering project? According to the Project Management Institute, “a project is a temporary endeavor undertaken to create a unique product, service or result.” [13] [16] INCOSE defines a project as “an endeavor with defined start and finish criteria undertaken to create a product or service in accordance with specified resources and requirements.” [26] The 1995 NASA Systems Engineering Handbook defines a project as the entity which “encompasses the design, development, and operation of one or more systems, and is generally managed by a NASA field center.” [27] In *The Handbook of Project-based Management, 3<sup>rd</sup> Edition*, Turner describes a project as “a temporary organization to which resources are assigned to do work to deliver beneficial change.” [28] The first two definitions highlight the time-definite process nature of the project which has a specific outcome as its goal (product, service, result). The NASA and Turner definitions retain the goal-orientation of the project and add the element of organization.

Collectively, an engineering project can be considered as the complex interaction of three fundamental elements: 1) the objective outcome (product, service, result, operation), which will be referred to in this thesis as the “product” for the sake of brevity; 2) the process by which that outcome is achieved (endeavor, activity, set of tasks); and 3) the organization that executes that process, which is a group of humans relating to one another in structure. [19] [29] In its most abstract form, *an engineering project is a system consisting of a process executed by an organization to create a product*. Each of these elements can be considered a domain of the project. Historically, project research has treated each domain separately in order to simplify the problem of analysis and isolate individual relevant factors and parameters. [20] [21] [25] [29] [30] [31] [32] [33] [34] [35]

In addition to product, process, and organization, Danilovic and Browning proposed two additional domains – the goal and tool domains. [36] This thesis argues that goals are actually a characteristic of each of the primary three domains because they must be expressed in terms of the domain(s) that the goal concerns. Tools are a subset of the organizational domain alongside other assets such as human resources, equipment, facilities, and so on. This thesis will thus consider a project in terms of three domains – product, process, and organization.

Project outcomes are measured in terms of cost, time, and scope. [14] Cost represents those factors in a project which can ultimately be represented in monetary terms such as budget and resources. Time represents those factors which can ultimately be represented in temporal terms such as duration and schedule. [14] Scope represents the factors of a project which are accomplished in the course of creating the final product, which are the result of work and the basis for value. [15] A foundational principle of project management theory is the concept of the triple constraint which governs the space of feasible combinations of these three outcome variables for a given project design. The triple constraint asserts that there exist pareto-optimal isoclines of project outcomes where, if waste and slack have been eliminated from a project, a desirable change in one variable necessarily drives an unfavorable change in one or both of the other two. [14]

These three variables – cost, time, and scope – are interrelated in complex ways which are specific to each project and which are never fully understood. Resources and personnel drive costs, create constraints which affect duration, and impact the quantity and quality of scope which can be realized. A solution for one project could easily be an obstacle for another. For example, the use of model-based systems engineering methods may increase the cost of software tools procured for a project and increase the duration and related scope of design tasks while the models are being constructed but may also reduce the costs and durations of modification or redesign activity and improve the quality of scope later in the life of the project. Whether the use of a particular method improves the aggregate cost-time-scope outcome of a project depends on a myriad of factors such as the complexity of the product, the processes put in place to employ and enable the method, and the organizational approach taken to ensure the project personnel can and do make efficient use of the method. In other words, the elements of any particular project interact in innumerable ways that create emergent outcomes of cost, time, and scope which are valued by the stakeholders in accordance with their unique priorities.

Scope deserves a bit more attention because it represents the reason that a project exists – the final product. The term also tends to be used vaguely to refer to different concepts which are related to scope but are not scope. Scope in the context of a project takes many forms. Information, code, hardware, buildings, or decisions made are all examples of possible scope within a project. Scope is the set of task *outcomes* accomplished to realize the end product, but scope is not the set of tasks themselves. [15] The same scope can be realized by two different sets of tasks that require different amounts of effort. Scope includes intermediate scope items which may not be valued by the stakeholders demanding the final product but which is created pursuant to it. [15] Scope may be intentional or accidental, planned or unplanned, useful or wasted. [15] A flawed design is scope, and that same design reworked is additional scope. The set of scope items which are demanded by the stakeholders chartering the project is the target scope. All other project scope is intermediate. [15]

Scope is the result of work and coordination and is the outcome of project tasks. [15] Scope items can be defined in quantity of units at a particular level of quality. [15] The criteria for measuring quality will reflect how the scope is valued by the stakeholders demanding it. For target scope, these stakeholders are typically referred to as customers, while for intermediate scope, these stakeholders may be the teams responsible for downstream project activities. Producing a quantity of a scope item at a particular quality requires several factors in combination such as tools, resources, facilities, equipment, knowledge, skills, people, and coordination. How these factors can be combined to produce the scope item are constrained by the nature of the scope item, such as the selected design, composition, and complexity. Complexity is a measure of the information needed to realize a given scope item and “corresponds to the proportion of coordination activity to nominal work activity required to transfer the scope across dependencies.” [15] How an organization chooses to obey or violate these constraints will determine the amount of exceptional activity and the resultant quality of the scope realized. [19] How an organization chooses, structures, and execute a set of tasks in order to ultimately produce the target scope will determine the total amount of scope realized during the project as well as the project’s total cost and duration. These decisions represent the

design of the project, and the specific project design leads to uniquely emergent cost, time, and scope outcomes through the highly complex interactions of the project elements throughout the course of the project. [19]

## **2.2 Modeling to Understand the Project System**

Since the mechanisms within a project that produce emergent cost, time, and scope outcomes are complex and non-intuitive, it is useful to use models with the aim of better understanding and predicting these outcomes.

Early process depictions evolved from the deterministic representations of Scientific Management and the Gantt Chart [30] [31], to combinations of deterministic and probabilistic precedence diagrams in the Critical Path Method, the Program Evaluation and Review Technique (PERT), and the Graphical Evaluation and Review Technique (GERT). [32] [33] [37]

More recent process representations rely on a higher level of abstraction which depicts the project as a set of activity phases. The Guide to the Program Management Body of Knowledge (PMBOK) describes a project in terms of a life cycle consisting of starting the project, organizing and preparing the project, carrying out the work of the project, and ending the project. [16]

Each of these methods focuses primarily on the time variable of projects, emphasizing effective ways to structure and manage project activities to meet milestones or ensure successful project outcomes. These models are thus primarily concerned with the process domain of a project.

Projects employing these methods often model project cost and schedule separately. This practice results in distortions when control measures are applied. [38] [39] This phenomenon drove the development of Earned Value Management as a model for representing a project as a series of scheduled tasks with planned costs and durations which can be compared to measured values as the project progresses. The earned value method and its extensions link activity, cost, and time in an attempt to understand, forecast, and control project outcomes. [40] [41] [42] [43] [44]

More recent process research focuses on addressing uncertainty and risk in project execution, optimizing project design elements such as task and resource planning, and using dynamic scheduling to adapt to random events. [45] [46] [47] [48] [49] This research has improved the ability to model the process domain of a project, and while most process models include references to the organizational and product domains, meaningful representations of the product and organization are largely absent from these process models.

Product models have historically been developed and maintained separate from the project responsible for them. Product models also take many forms and may be a combination of many separately maintained models representing a certain type of information. For example, a CAD model may represent the physical geometry, interfaces, and material composition of a product, while a data model may represent the how information is transmitted, received, stored, and manipulated within a product. [50] Integrated product models allow for analysis of multiple



characteristics simultaneously either by combining the information in an integrated database or by interfacing with subordinate models and presenting the information in an integrated fashion.

A recent trend in integrated product modeling is the adoption of model-based systems engineering methods. One of the hurdles to widespread adoption has been the uncertainty in whether regulatory agencies will embrace the use of these methods. [51] The Department of Defense published a digital engineering strategy in 2018 which will reduce this uncertainty and lead to increasing levels of adoption among defense contractors. [10] The Department of Defense Architectural Framework (DoDAF) version 2.0 also promotes model-based systems engineering methods through the prescription of the DoDAF Meta-Model (DM2). DoDAF is an example of how multiple representations of a product can be combined to provide an integrated model. [52] Despite the significant advances in integrated product modeling, the influence of the process and organizational elements of a project are largely absent. [29] 5D Business Information Modeling (BIM), a technique becoming increasingly popular in the construction industry, combines the 3D product modeling of CAD with time and cost modeling. [53] Such a model has the potential to link the product domain to both process and organizational domains, but more work remains before this technology could be used to support integrated project modeling.

Organizational and team models have developed more slowly than process and product models. Early organizational models were consistent with and backdrop to the process models they were associated with. [19] The deterministic period of the early twentieth century embraced a mechanical model of the organization based on a static organization chart with little or no emphasis on personal human factors. [19] As process models evolved, McGrath's Input-Process-Output model became the basis for representing a wide range of organizational behavior. [54] Project organization models began to consider a wide range of factors such as cognitive limits [21], individual and collective motivations [20], the span of team goals [55], resources [35], team interdependence [22], knowledge and learning [56] [57] [58] and many others. There has yet to emerge a consensus on what the most important factors in organizational project behavior are or how the relevant factors interact with one another. Since team behavior is the product of the complex interactions of individuals within the structure of an organization, much work must still be done before a stable unifying organizational project theory will emerge. Similar to how product models are evolving to integrate ever larger sets of factors, organizational models should expand to consider multiple factors and their interplay simultaneously. This will allow for more accurate representations of how organizational factors influence project behavior and outcomes.

While not widespread, some project research has produced model frameworks and methods which capture the product, process, and organizational elements of a project. The Project-Product Lifecycle Model approach developed by Dori and Sharon is one such framework which employs the Object-Process Methodology. [29] Project dynamics, a form of systems dynamics adapted to projects, offers a methodology capable of representing the interactions between organization, product, and process through the approximation of positive and negative feedback cycles. [59] Agent-based modeling has also been applied successfully to simulate projects where

product, process, and organizational factors are represented. The agent-based modeling approach developed by Moser and Wood emphasizes the sociotechnical nature of projects where emergent outcomes are driven by the “dynamic interaction of technical and social characteristics.” [19] Multidomain Matrices (MDMs), which combine two or more Design Structure Matrices (DSMs) from different domains, can be used to model the relationships between product, process, and organization in the context of a project. [60] Each of these methods can be used to construct a project representation for conceptual modeling and simulation to explore how various project choices may manifest in cost-time-scope outcomes. Because these models consider a more complete set of project factors than traditional models, they have the potential to more accurately and realistically forecast project behavior. They also have the potential to facilitate more effective interdisciplinary project coordination because they capture how the parameters of interest to different disciplines interact with and influence one another, enabling a broader basis for common understanding and communication between those with different perspectives on the working of the project. [36]

## 2.3 Systems Analysis Methodologies

This section includes a brief introduction to the methods which are leveraged in the research and analysis portion of this thesis. The concepts underlying these methods inspire both the framing and analysis of the problem motivating this thesis.

### 2.3.1 Systems Thinking and System Problem Statements

“Systems thinking” is the practice of “thinking about a question, circumstance, or problem explicitly as a system.” [61] Key to the systems thinking method is viewing a system in terms of both its form (what it is) and function (what it does). [61] Systems thinking involves viewing a system both internally and externally. An internal view involves decomposing the system into subordinate parts and functions and understanding how those constituents relate and interact to create an emergent functionality which is greater than their sum. [61] An external view involves understanding the context of the system in order to define how it delivers value and how it influences and is influenced by its environment. [61] The application of systems thinking reveals the architecture of the system. [61]

In defining the architecture of a system, it is valuable to define the system problem statement, which is the “single assertion of what the system is intended to accomplish to deliver value.” [61] A canonical system problem statement is based on a sequence of statements in the form of “To-By-Using.” [61] The “To” statement expresses the solution-neutral transformation that the system is meant to effect. [61] The “By” statement expresses the solution-specific process that the system employs to accomplish this transformation. [61] The “Using” statement specifies the form of the system that enables the solution-specific process. [61]

### 2.3.2 Design Structure Matrix

The Design Structure Matrix (DSM) “is a network modeling tool used to represent the elements comprising a systems and their interactions, thereby highlighting the system’s architecture (or designed structure).” [60] DSM can be used to represent any type of relationship between a set of  $N$  elements within a system. Consider an  $N \times N$  ( $N$ -squared) matrix  $\mathbf{A}$ .  $\mathbf{A}$ , the DSM for the system, contains information in the cell  $a_{ij}$  representing the relationship between the  $i^{th}$  and  $j^{th}$  element. Typically, this information concerns some type of dependency where one element requires information, energy, matter, etc. from the other. A convention is established by the author of the DSM where either rows or columns represent the source and the other represents the sink.

It is interesting to note that research on DSMs is focused on systems in general versus projects specifically, Eppinger and Browning classify four types of DSM models which account for nearly all DSM models. [60] The first three DSM types precisely match the project domains identified in the preceding Background section: Product, Organization, and Product DSMs. The fourth type results from a blend of at least two of the first three. When two or more DSMs representing different domains are combined, the resulting matrix is called a Multidomain Matrix (MDM). The subspace of such a matrix that represents the intersection of two domains is known as a Domain Mapping Matrix (DMM) and shows the projection of one domain onto the other. Figure 1 shows an example of an MDM containing three DSMs – one for product, one for process, and one for organization – as well as three DMMs – one for the intersection of the

product and process domains, one for the intersection of the product and organization domains, and one for the intersection of the process and organization domains. [60]

<b>Product DSM d x d</b>	<b>Product - Process DMM d x p</b>	<b>Product-Organization DMM d x o</b>
	<b>Process DSM p x p</b>	<b>Process-Organization DMM p x o</b>
		<b>Organization DSM o x o</b>

Figure 1: "Periodic table" of DSMs and DMMs, forming an MDM (adapted from [60])

### 2.3.3 Object-Process Methodology

The Object-Process Methodology (OPM) provides a domain-agnostic framework for representing a system using symbols that represent either objects or processes, their relationships and their states. [62] OPM diagrams can be expressed using natural language consistent with the symbols in the diagram with the goal of being easily understandable and communicable. [62] OPM has been formalized for use as a standard in ISO 19450. The key symbols used in OPM are shown in Table 1.

OPM has been used to express systems architectures in a form which is intuitive and easy to understand. [61] An OPM depiction of the generic definition of an engineering project which was given in the "Background" section, *an engineering project is a system consisting of a process executed by an organization to create a product*, is shown in Figure 2. An OPM representation of a generic system problem statement is shown in Figure 3. The OPM diagrams and language created for this thesis are constructed using OPCloud Version 3.1. [63]

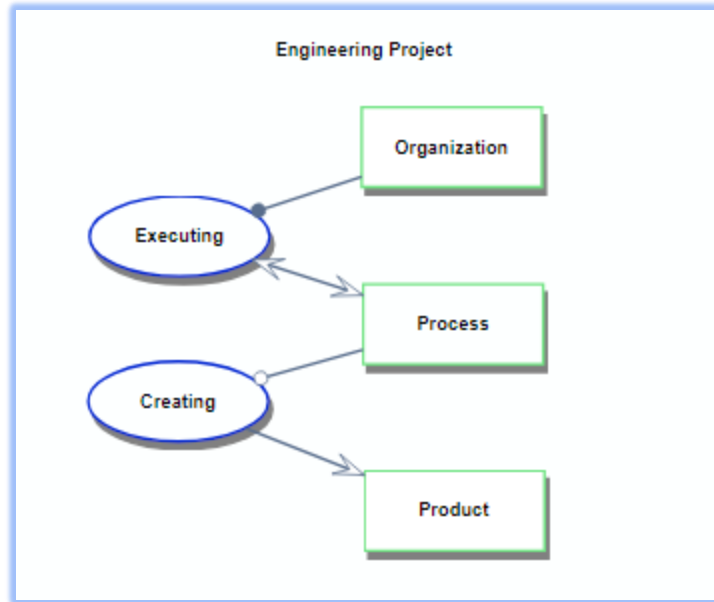


Figure 2: OPM diagram of a generic engineering project

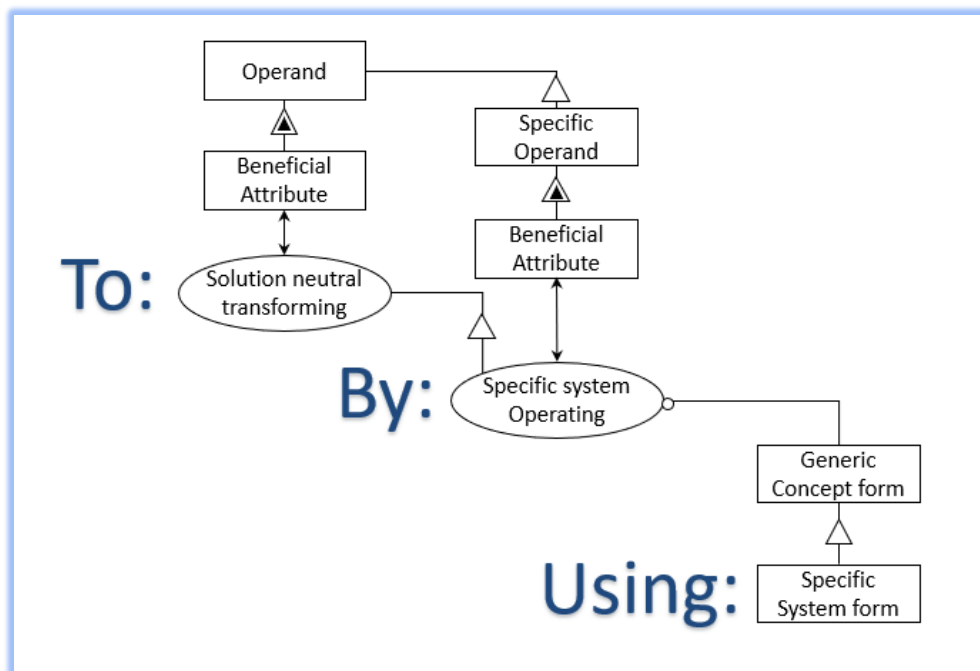















Figure 3: OPM diagram of generic system problem statement (adapted from [61])

Table 1: OPM Symbols as Defined in ISO 19450 (Reproduced from [64])












Entities

	Object: A thing that exists physically or informatically
	Physical object: An object in the physical world (default is informatical)
	Environmental object: An object that is not systemic but interacts with the system (default is systemic)
	Process: A thing that transforms an object by creating or consuming it or by changing its state
	Physical process: An object in the physical world (default is informatical)
	Environmental process: A process that is not systemic but interacts with the system (default is systemic)
	State: A situation at which an object can be (called value of the object is an attribute of another object)

Structural Links

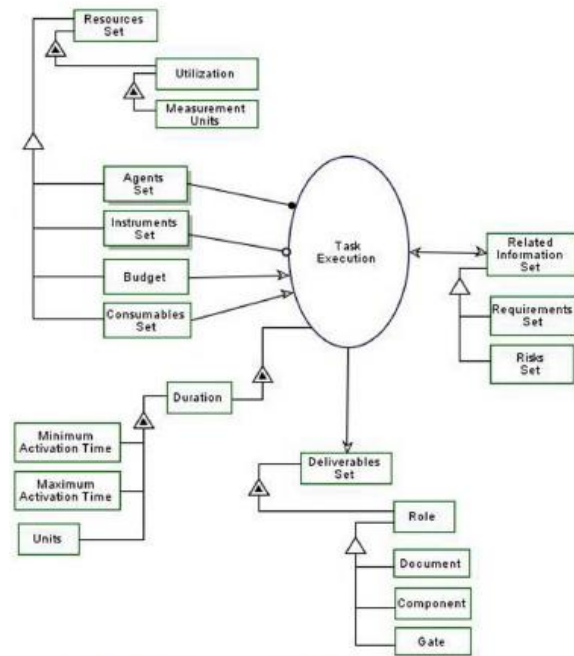
	Aggregation-Participation: A link between a whole thing (object or process; linked to top) and its parts (linked to bottom)
	Exhibition-Characterization: A link between an exhibiting thing (linked to top) and its features (attributes and/or operations; linked to bottom)
	Generalization-Specialization: A link between a general thing (linked to top) and its specializations (linked to bottom)
	Classification-Instantiation: A link between a class of things (linked to top) and its instances (linked to bottom)
	Unidirectional tagged structural link: A user-defined tag that expresses the relation between two things
	Bidirectional tagged structural link: A user-defined pair of tags that express the forward and backward relations between two things

Procedural Links (edges show source and destination)

	Agent Link: A link from an agent (a human enabling object, which is not affected by the process) and the process it enables
	Instrument Link: A link from an instrument (a non-human enabling object) and the process it enables
	Effect Link: A link between a process and an object whose states change as a result of the process execution
	Consumption Link: A link from a consumed object to the consuming process
	Result Link: A link from a generated object process to the generating process
	Input-Output Link Pair: A pair of links from (1) the input state (left) to the affecting process, and (2) from that process to the output state (right)
	Invocation Link: A link from an invoking process to an invoked process
	Instrument Event Link: An instrument link from the triggering object (or state, not shown) to the invoked process (the letter e may be beside to circle)
	Consumption Event Link: A consumption link from the triggering object (or state, not shown) to the invoked process, which, if executes, consumes the object
	Instrument Condition Link: An instrument link from the conditioning object (or state, not shown) to the conditioned process, which is skipped if the object does not exist (or not in the state) (the letter c may be beside to circle)
	Consumption Condition Link: A consumption link from the conditioning object (or state, not shown) to the conditioned process, which is skipped if the object does not exist (or not in the state), and consumes the object otherwise

### 2.3.4 Project-Product Lifecycle Management Framework

Dori and Sharon introduced the project-product lifecycle management (PPLM) framework in 2008. [29] The focus of this approach is to integrate the product, project (i.e. process), and enterprise (i.e. organizational) domains for systems engineering purposes. [29] PPLM uses Object-Process Methodology (OPM) to represent and relate the product, process, and organizational elements within a project at a level of abstraction useful and manageable by the systems engineering team within a project. [29] The project-product model is generated and maintained as an authoritative record of both the project and product design from the project's inception. The model can then be used for systems engineering and program management functions such as requirements decomposition, interface management, work package planning, etc. [29] Because the model represents multiple domains, it can be used to understand how changes in one domain may impact the others. It can also be used to trace constraints and dependencies across domains. [65] An example of how different project domain elements are represented in PPLM is shown in the OPM model in Figure 4. Elements from each of the three project domains are included in a single model, and the interconnected nature of these domains is evident in the relationships between the elements. For example, agents which represent the teams of engineers, technicians, analysts, etc. (organization domain) handle task execution (process domain) which yields deliverables (product domain). [65] An example of the Object-Process Language which represents the model in paragraph form is shown below the diagram. The flexible and intuitive nature of PPLM and OPM makes it suitable for representing the unique characteristics of different domain-specific and discipline-specific project elements.



**Task Execution** exhibits **Duration**.  
**Duration** exhibits **Minimum Activation Time**, **Maximum Activation Time**, and **Units**.  
**Resources Set** exhibits **Utilization**.  
**Utilization** exhibits **Measurement Units**.  
**Agents Set** is physical.  
**Agents Set** is a **Resources Set**.  
**Agents Set** handles **Task Execution**.  
**Budget** is a **Resources Set**.  
**Instruments Set** is physical.  
**Instruments Set** is a **Resources Set**.  
**Task Execution** requires **Instruments Set**.  
**Task Execution** affects **Related Information Set**.  
**Task Execution** consumes **Budget** and **Consumables Set**.  
**Task Execution** yields **Deliverables Set**.  
**Deliverables Set** exhibits **Role**.  
**Document** is a **Role**.  
**Gate** is a **Role**.  
**Component** is a **Role**.  
**Requirements Set** is a **Related Information Set**.  
**Risks Set** is a **Related Information Set**.  
**Consumables Set** is a **Resources Set**.

Figure 4: OPM model of a generic project construct (reprinted from [65])

One demonstrated application of PPLM is the use of a view called the object work breakdown structure. [65] In this view, traditional work breakdown elements which represent tasks in the process domain are linked to the product breakdown structure elements which are inputs, outputs, or instruments to the tasks (see Figure 5 for an example). These relationships are again

modeled and managed in OPM from the beginning of the project to allow for management of both the process and product simultaneously. By managing the process and product together, the interactions between these domains can be evaluated when considering design choices, changes, risks, or unexpected events. This allows for a deeper understanding of dependencies and constraints which flow across domains and which have no method for expression in a standard work breakdown structure

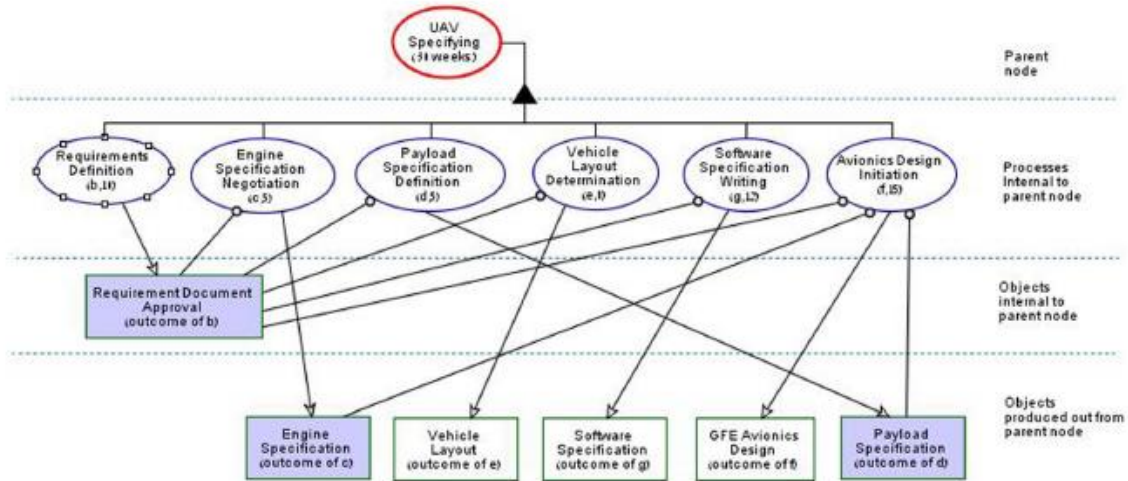


Figure 5: Object WBS Example (reproduced from [65])

A logical extension of this application would be to add organizational elements to the view which show the teams responsible and resources necessary to execute tasks and create products.

### 2.3.5 Project Design

Moser and Wood propose an approach called Project Design which formally treats a project as a complex sociotechnical system. [19] This approach focuses on representing the “essence rather than details of product, process, and organization” at a level of abstraction that allows for exploration of design options that result in differing levels of project feasibility and value. [19] The modeling approach relies on three project elements types: product elements, team elements (organizational domain), and phase-activity elements (process domain) (see Figure 6 for an example). These elements are grouped into a product breakdown structure, organizational breakdown structure, and work breakdown structure, respectively. [19] These elements are related in the modeling approach by parameters and functions meant to reproduce project phenomena. Complexity is presented as “the cost of uncertainty reduction,” dependence as “a continuous and mutual demand for coordination between teams,” coordination as an information sharing activity proportionate to the complexity of a task or product, exceptional activity as a result of dependency violation, and attention as how participants manage competing demands. [15] [19] The specific nature of these relationships is not obvious in the model diagram, but is captured in data structures underlying the symbols.



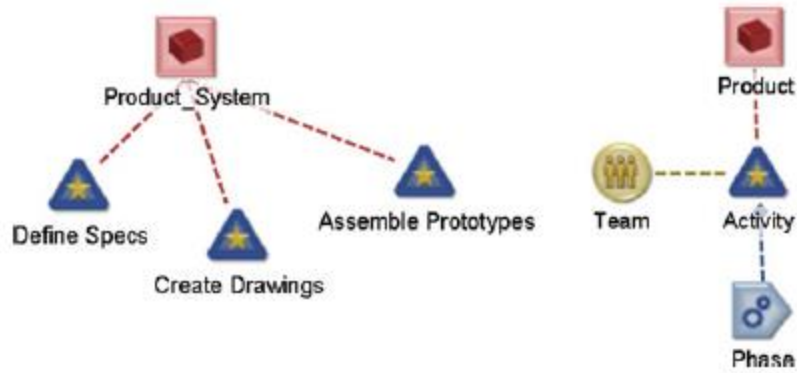


Figure 6: Project Design element relationships (reproduced from [19])

This method is differentiated from many other integrated modeling methods by the emphasis it places on the social aspects of the organizational domain with an aim towards capturing the complex psychological and behavioral factors relevant to organizations and teams. The modeling approach allows for an evaluation of how these factors impact process and product outcomes in a project. The approach relies on agent-based simulations to forecast project behavior and outcomes under uncertain conditions. [19]

## 2.4 Project Management Process

According to the Project Management Institute, “project management is the application of knowledge, skills, knowledge, skills, tools, and techniques to project activities to meet the project requirements.” [16] The Project Management Body of Knowledge defines the five project management process groups shown in Figure 7: Initiating, Planning, Executing, Monitoring and Controlling, and Closing which form a system for managing a project throughout its lifecycle. [16] The processes within these groups are defined in detail in the PMBOK guide along with the recommended ways to implement them in order to achieve desired project outcomes. The PMBOK guide calls for implementing these processes in a way that is tailored to the structure and characteristics of the particular project being managed.

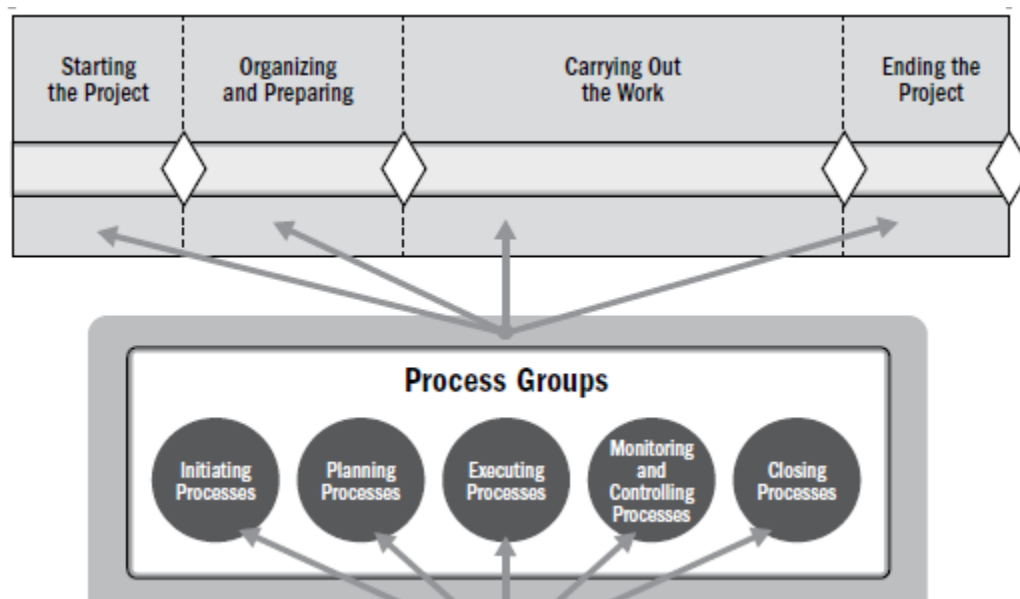


Figure 7: Project Management Process Groups supporting the Project Life Cycle (reproduced from [16])

In Table 2, Turner describes the project management life cycle consisting of five stages: concept, feasibility, design, execution, and close-out. [28] Vanhoucke models the project lifecycle according to six activities shown in Figure 8.

Table 2: The Basic Project Management Life Cycle (reproduced from [28])

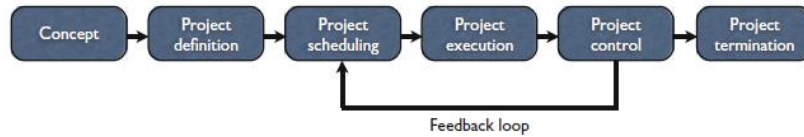


Figure 8: Project Life Cycle (reproduced from [66])

Several project life cycle models emphasize the product development process aspect of a project lifecycle vice the project management process – see Figure 9, Figure 10, and Figure 11.

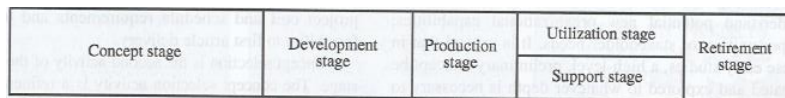


Figure 9: Generic project life cycle (ISO/IEC/IEEE 15288:2015) (reproduced from [26])

Name	Process
Concept	Identify opportunity for performance improvement Diagnose problem
Feasibility	Develop proposals Gather information Conduct feasibility Estimate design
Design	Develop design Estimate costs and returns Assess viability Obtain funding
Execution	Do detail design Baseline estimates Do work Control progress
Close-out	Finish work Commission facility Obtain benefit Disband team Review achievement

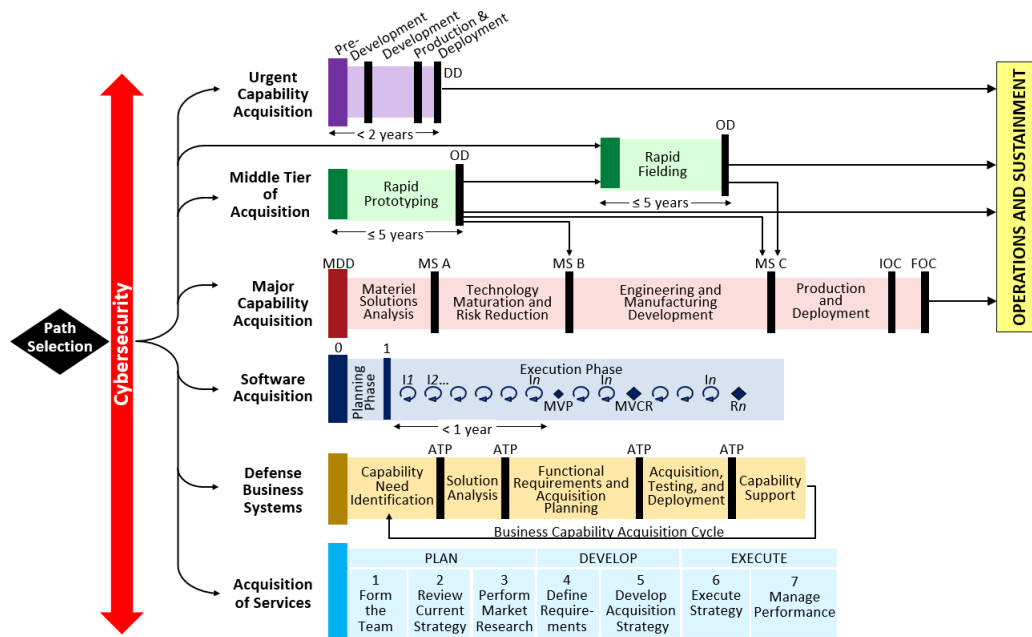


Figure 10: Department of Defense Adaptive Acquisition Framework Pathways (reproduced from [67])

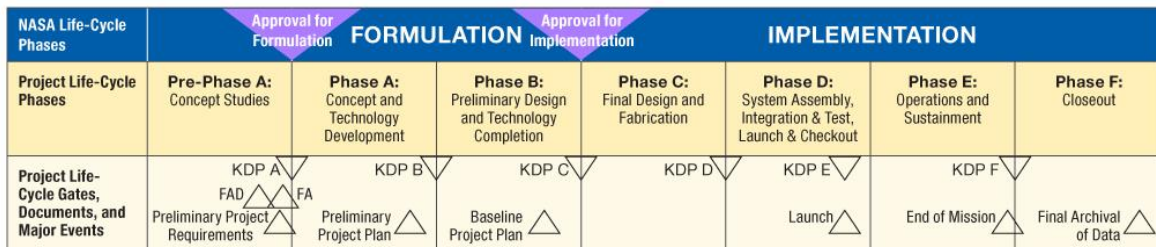


Figure 11: NASA Project Life-Cycle Phases (reproduced from [68])

The same essential project management concepts can be distilled from each of these models. Each model begins with an initiation process where the project targets are defined and negotiated based on feasibility followed by a planning process where the execution strategy is formulated. The strategy is then implemented, measured, and controlled, the project targets are realized, and the project is closed. While these processes must happen in sequence, that sequence may iterate and cycle, different processes may take greater or lesser significance, and the formality of the structure can vary greatly depending on the nature of the project. The goal of the project management process is thus to realize the project targets as efficiently and effectively as possible, and all project management activities should be directed towards this end. How targets are met will also be dependent on the specifics of a given project, and there will be no one-size-fits-all optimal approach to project management. Reflective of this principle, each project management guide and handbook emphasizes the need to tailor the methods used to fit the project. [16] [26] [28] [69]

It is worth acknowledging the ease of conflating the project management and systems engineering management disciplines. Both fields are concerned with managing engineering projects to create a satisfactory end product, and the responsibilities of each role overlap considerably. For example, both the systems engineering manager and project manager are concerned with the duration of project activities, success or failure of subsystem development and fabrication, and the status of funding for project tasks. Systems engineering managers are primarily concerned with coordinating the various specialized engineering disciplines and technical processes in order to realize scope targets and must be cognizant of cost and schedule constraints in order to perform their primary task, while project managers are concerned with the overall prioritized set of cost, time, and scope targets and concerned with coordinating the project processes and efforts to realize a project outcome that satisfied the combined project targets. [16] [26] [70] Regardless of the emphasis, the tools for each discipline contain oftentimes overlapping information. Each discipline plays a crucial role in achieving successful project outcomes with a strong influence on the other. [16] [26] [27] [70] The principles of each discipline are relevant when considering the overall process of managing a project.

## 2.5 Project Control Process

A key activity within the project management process is that of measuring the project status and making changes to correct for observed or anticipated variances. This sub-process is known by various names within different project management frameworks such as assessment and control, monitoring and control, project control, gate reviews, or milestone reviews. [16] [26] [27] [71] The risk management technical management process also overlaps with this activity in that its purpose is to anticipate and prevent or mitigate negative project outcomes. [26]

According to the INCOSE Systems Engineering Handbook, the purpose of the assessment and control process is to assess if the project plans are feasible, determine the status of project, technical, and process performance, and direct execution to satisfy technical objectives and ensure performance is according to plans, schedules, and budgets. [26] The Guide to the Project Management Body of Knowledge identifies Monitoring and Controlling as one of five project management process groups. [16] This group “consists of those processes required to track, review, and regulate the progress and performance of the project; identify any areas in which changes to the plan are required; and initiate the corresponding changes.” [16] The PMBOK asserts that the primary value of project control lies in identifying and correcting variances from the project management plan. [16] The 1995 NASA Systems Engineering Handbook discusses project control as a feedback loop which is conducted at each level of a project hierarchy and in the contexts of cost and schedule control measures, technical performance measures, and systems engineering process metrics. [27] In each context, the goal of the control process is to identify and address problems related to the development of the product and execution of the project.

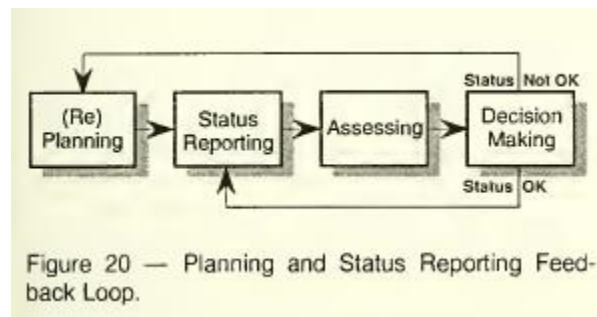


Figure 20 — Planning and Status Reporting Feedback Loop.

Figure 12: Project Control Feedback Loop (reproduced from [27])

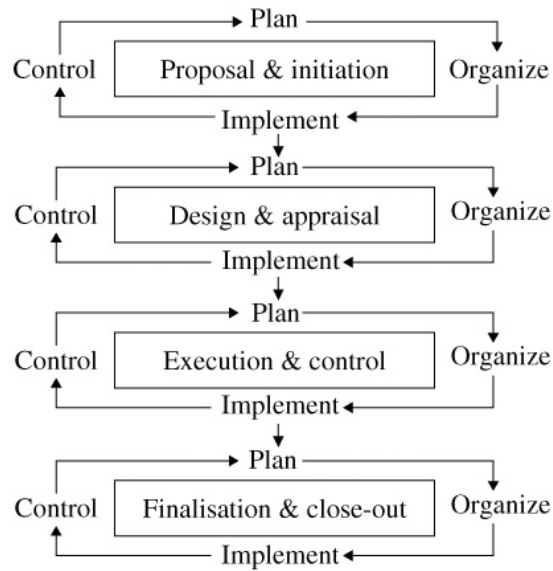


Figure 13: Control in project management phases (reproduced from [28])

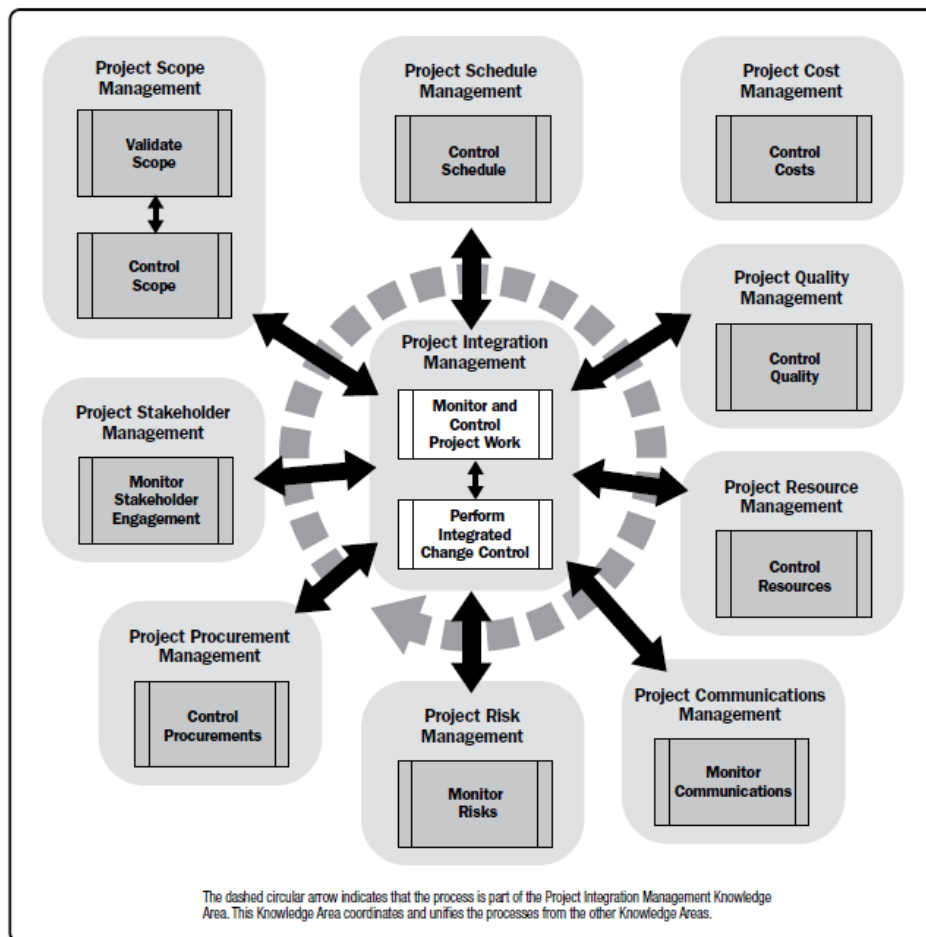


Figure 14: Monitoring and Controlling Process Group – detailed view (reproduced from [16])

In the *Mythical Man Month*, Brooks emphasizes the importance of rigorously built schedules with concrete, specific, measurable, clearly defined milestones. [34] He feels that project control is an activity which must be shared between higher- and lower-level managers in an organization, implying that some decisions are best made by those with a broad system perspective while others are best made at the local level. He also discusses the value of having a dispassionate team responsible for measuring and reporting on progress which both minimizes the reporting burden on the lower-level managers and ensures an objective accounting of minor delays which otherwise might be papered over.

Mihm et al found that decisions which account for effects upon an entire system as opposed to optimizing local performance lead to superior results. [55] Pich et al discuss the implications of uncertainty and ambiguity in projects and the need to develop strategies to address them. [72]

In *Total Project Control*, Deveaux claims that formal project control systems are unnecessary if a project is planned and managed well. [73] Instead, frequent and informal status reporting combined with a culture of early, rapid problem identification and resolution will ensure that projects self-control.

While there is considerable variation in the level of abstraction, decision making, and emphasis in the different frameworks, they each agree at a basic level on the need to measure project status, compare those measurements to the project plan, and implement changes to the project when those comparisons warrant. This thesis will use the term “project control” to refer to this process.

## 2.6 DoD Acquisition Process as System

Engineering projects performed by the Department of Defense take place within the auspices of the Defense Acquisition System. The Defense Acquisition System is governed by the Federal Acquisition Regulation (FAR), Defense Federal Acquisition Regulation Supplement (DFARS), and a collection of Department of Defense directives and instructions collectively referred to as the “5000 series” due to their nomenclature (each document is numbered with the same convention: 50XX.XX). The primary 5000-series document is DoD Directive 5000.01, “The Defense Acquisition System,” which establishes overarching acquisition policy and expresses foundational management principles.

From a systems thinking perspective, DoD Directive 5000.01 establishes the primary value-related process of the Defense Acquisition System as: “[supporting] the National Defense Strategy, through the development of a more lethal force based on U.S. technological innovation and a culture of performance that yields a decisive and sustained U.S. military advantage.” [69] It also states that “the acquisition system will be designed to acquire products and services that satisfy user needs with measurable and timely improvements to mission capability, material readiness, and operational support, at a fair and reasonable price.” [69] These statements can be combined to infer a system problem statement for the Defense Acquisition System:

*To: Develop a more lethal force which yields a decisive and sustained U.S. military advantage and supports the National Defense Strategy*

*By: Acquiring products and services that satisfy user needs with measurable and timely improvements to mission capability, material readiness, and operational support, at a fair and reasonable price*

*Using: The Defense Acquisition System [69]*

Figure 15 shows an OPM representation of this system problem statement.



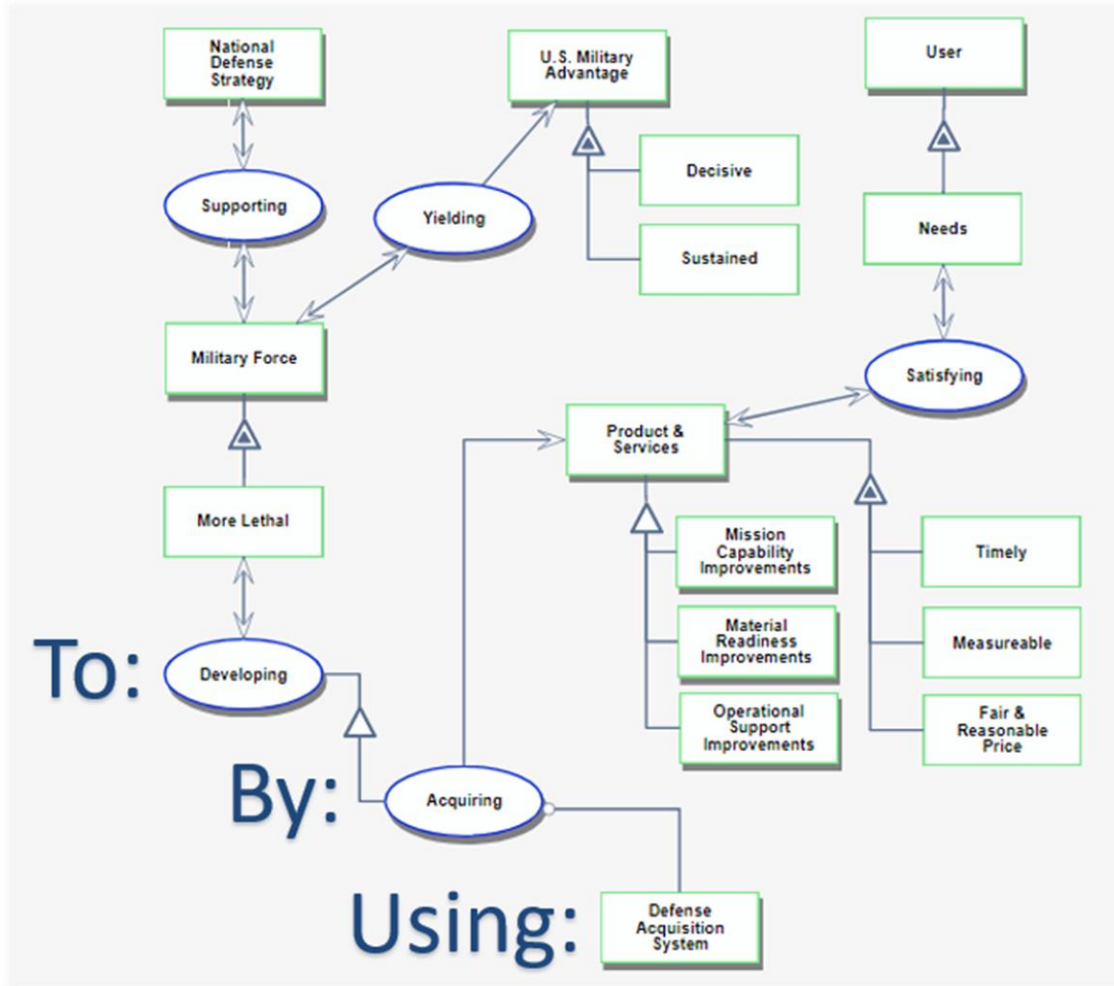


Figure 15: OPM diagram of Defense Acquisition System system problem statement

The “By” statement is consistent with the system view of an engineering project established earlier in this thesis. A defense acquisition project is a process executed by an organization to acquire a product. In contrast to the generic engineering project discussed in Section 2.1, a defense acquisition project acquires instead of creates the product. This implies a transfer of the process “creating” to a third-party organization from which the products will be acquired. This is the model the Department of Defense typically uses. In this sense, a defense acquisition is an engineering project where the organization can be considered a partnership between the Department of Defense “acquiring” and the engineering firm “creating” a product. The remaining verbiage in the “By” statement, “that satisfy user needs...”, speaks to the project outcomes that form the basis of value for the Department of Defense and are in terms of the project outcome variables of cost, time, and scope which have been established previously in this thesis. It follows that any specific defense acquisition project will have specific prioritized cost, time, and scope outcome targets. These targets define the project goals and satisfy the principles outlined in the system problem statement.

The Defense Acquisition System is in a period of significant change as the Department of Defense revises and reissues the 5000-series documents to improve process effectiveness and implement a new Adaptive Acquisition Framework. [74] The revised policy specifically calls for innovative and tailored approaches to the management of acquisition programs. [69] [74] These revisions, coupled with newly defined procedures to allow for more flexible project structures in the Adaptive Acquisition Framework (see Figure 10), create an environment conducive to implementing alternative project management methods with the potential to improve project outcomes. [67] [74] [75]

## 2.7 Project Control in DoD Acquisitions

Despite the significant policy changes taking place in the Defense acquisition community, the specific policies governing control of acquisition projects remain relatively unchanged. Project control policy for Defense acquisitions flows from the Capital Planning Guide within the Office of Management and Budget Circular No. A-11, “Preparation, Submission, and Execution of the Budget.” In the *Risk Management* section, it states: “cost, schedule, and performance goals are to be controlled and monitored by using an earned value management system.” [18] The instructions in the OMB Circular No. A-11 are decomposed and expanded upon in FAR section 34.2, DFARS section 234.2, and DoD Directive 5000.85. These documents prescribe the contract-specific requirements to use an Earned Value Management System (EVMS) shown in Table 3 and Table 4.

Table 3: DoD EVM Application Requirements (reproduced from [76])

Contract Value	Applicability	Notes	Source
< \$20M	EVMS not required; may be applied at PM discretion based on risk to the Government	Requires business case analysis and MDA approval.	Part 7 of Office of Management and Budget Circular A-11 FAR 52.234-4, FAR subpart, 34.2 DFARS 234.201 DoDI 5000.85, Para. 3C.3.c.(3)
≥ \$20M & < \$100M	EVMS Required; Contractor is required to have an EVMS that complies with the guidelines in EIA-748.*	The Government reserves the right to review a contractor’s EVMS when deemed necessary to verify compliance.	
≥ \$100M	EVMS Required; Contractor is required to have an EVMS that has been determined to be in compliance with the guidelines in EIA-748.*	The Contractor will provide access to all pertinent records and data requested by the Contracting Officer or duly authorized representative as necessary to permit initial and ongoing Government compliance reviews to ensure that the EVMS complies, and continues to comply, with the guidelines in EIA-748.*	

Table 4: DoD EVM Reporting Requirements (reproduced from [77])

Contract Value	Applicability	Notes	Source
< \$20M	Not required	IPMDAR may be used if cost and/or schedule reporting is requested by the program management office	Integrated Program Management Report DID DI-MGMT-81861B*
≥ \$20M & < \$100M	Required monthly when EVMS requirement is on contract	All IPMDAR datasets/ files must be included in the CDRL. Tailoring in accordance with DI-MGMT-81861B and Implementation Guide is allowed.	
≥ \$100M	Required monthly when EVMS requirement is on contract	IPMDAR is required. All files are required.	

In addition to meeting these criteria, a program must also include discrete, measurable scope and use a cost reimbursable or incentive contract with a period of performance greater than 18 months in order to require an EVMS. [76] [78] Most new large, complex engineering defense projects fall into this category and are thus subject to EVMS requirements. These criteria apply regardless of which acquisition pathway within the Adaptive Acquisition Framework a project uses. [79]

Consistent with the emphasis on tailoring and innovation in DoD Directive 5000.01 and DoD Instruction 5000.02, the policies governing EVMS allow for deviations from the baseline prescriptions in Table 3 and Table 4. [79] [80] Typically, the service (e.g. Air Force) or agency

(e.g. NRO) responsible for a given acquisition program has the authority to exempt it from including EVMS. Department of Defense policy discourages such an exemption, however, based on the assumption that earned value is the preferred program control technique for programs with discrete work using a cost reimbursable or incentive contract above the \$20M threshold with a duration longer than 18 months. [79] The cost of obtaining an exemption on an organization can be high as a result, and a resource-constrained organization will likely choose to apply earned value even if there may be better alternative project control methods available.

It is important to note that the use of an Earned Value Management System involves more than the application of the earned value method. The ANSI/EIA-748 standard governs private sector use of EVMS, while the Department of Defense Earned Value System Interpretation Guide governs the application of EVMS on defense programs. [81] The Department of Defense uses this interpretation guide to assess the compliance of a contractor’s EVMS with the 32 guidelines found in ANSI/EIA-748. The application of an EVMS is “based on the premise that the government cannot impose a singled integrated management solution for all contractors,” and the principle of letting contractors manage defense projects with their own internal management systems is core to EVMS. [79] [82] Indeed, reviewing the 32 guidelines in Table 5 shows that many of the criteria for a compliant EVMS are project management or accounting practices that are nothing more than prerequisites for applying the earned value method correctly.

*Table 5: ANSI/EIA-748 EVMS Guidelines [79]*

1. Define authorized work elements in a Work Breakdown Structure (WBS)	17. Summarize direct costs by WBS elements
2. Define Program Organization Structure	18. Summarize direct costs by OBS elements
3. Integrate subsidiary management processes	19. Record/allocate indirect costs
4. Identify overhead management	20. Identify unit and lot costs
5. Integrate WBS and Organization Breakdown Structure (OBS) to create control accounts	21. Track and report material cost/quantities
6. Sequential scheduling of work	22. Calculate schedule variance and cost variance
7. Identify products and milestones for progress assessment	23. Analyze significant variances
8. Establish the Performance Measurement Baseline (PMB)	24. Analyze indirect cost variances
9. Authorize and budget by cost elements	25. Summarize performance data and variances for management reporting
10. Determine discrete work and objective measures	26. Implement corrective actions
11. All work package budgets & planning packages sum to control account	27. Maintain estimates at completion
12. Plan and control Level of Effort (LOE) budgets	28. Incorporate changes in timely manner

13. Establish overhead budgets	29. Maintain baseline and reconcile budgets
14. Identify management reserve and undistributed budget	30. Control retroactive changes
15. Reconcile program target cost goal with sum of all internal budgets	31. Prevent unauthorized revisions
16. Record direct costs	32. Document changes to Performance Measurement Baseline (PMB)

The Guide to the Program Management Body of Knowledge, Chapter 7 details the mechanics of earned value analysis. [16] The Department of Defense Earned Value Management Implementation Guide and the Defense Acquisition Guidebook give specific techniques for applying the earned value method to Defense programs. [71] [79] In addition to traditional earned value analysis, this guidance also calls for the use of schedule network analysis methods to identify the critical path and conduct probabilistic schedule risk assessments for schedule control. [16] [81]

Whether using an EVMS or not, defense acquisition programs implement project control through a series of high-level milestone reviews and low-level management reviews. In these reviews, various program information is presented, discussed, and decided upon. [78] The information presented during these reviews includes earned value metrics; network schedule information; scope progress in the form of technology maturity, design progress, production metrics, or test results; resource status and projections; and project risks, issues, and opportunities. [71] [83]

Department of Defense policy documents give an extensive set of reasons for the use of an EVMS. The principal documents in which the Department of Defense expresses these reasons are DoD Instruction 5000.85, the Defense Acquisition Guidebook, and the Department of Defense Earned Value Management Implementation Guide. Table 6 lists the statements from these documents which state or directly imply a justification for using earned value management. The Defense Acquisition Guidebook acknowledges that several of these reasons do not specifically necessitate an EVMS, and instead they represent “best program management practice for development and delivery of both products and services.” [71]

Table 6: DoD Reasons for EVMS

5000.85 "Major Capability Acquisition": [78]	Defense Acquisition Guidebook: [71]
To "ensure sound planning and resourcing of all tasks required for contract performance"	To "provide Joint situational awareness of contract status"
<b>DoD Earned Value Management Implementation Guide: [79]</b>	To "assess the cost, schedule, and technical performance of contracts for proactive decision-making"
To "ensure that contractors have adequate management systems that integrate cost, schedule, and technical performance"	To "measure contract performance and progress in an objective manner"
To provide "better overall planning, control, and disciplined management of government contracts"	To ensure contractors: <ul style="list-style-type: none"> <li>- "plan all work scope" and "assign all work scope"</li> <li>- "integrate scope, schedule, and cost for measuring performance"</li> <li>- "record actual costs incurred"</li> <li>- "objectively assess accomplishments where work is being performed"</li> <li>- "assess variances"</li> <li>- "implement corrective actions"</li> <li>- "use performance information for decision making"</li> </ul>
To allow for "planning all work scope for the program from inception to completion"	
To allow for "assignment of authority and responsibility at the work performance level"	
To allow for "integration of the cost, schedule, and technical aspects of the work into a detailed baseline plan"	
To allow for "disciplined baseline maintenance and incorporation of baseline revisions in a timely manner"	
To provide "the [project manager] with an effective decision making tool"	To ensure the government receives performance information that: <ul style="list-style-type: none"> <li>o "Is exactly the same as what the contractor uses"</li> <li>o "Is timely, accurate, reliable, and auditable"</li> <li>o "Relates time-phased budget to scope of work"</li> <li>o "Measures progress as objectively as practicable"</li> <li>o "Enables independent government predictions of future cost and schedule conditions"</li> <li>o "Contributes to Acquisition Program Baseline (APB) trade-space decisions"</li> </ul>
To "facilitate risk identification and mitigation"	
To "ensure that valid cost, schedule, and technical performance information are generated"	
To allow for "objective measurement of progress at the work performance level with EVM metrics"	
To allow for "accumulation and assignment of actual direct and indirect costs"	
To allow for "analysis of variances or deviations from plans"	To ensure government contracts operate subject to "disciplined planning and control processes"
To allow for "summarization and reporting of performance data to higher levels of management for action"	To facilitate the government "develop[ing] relationships with their contractor counterparts to facilitate discussions regarding performance towards completion of the contractual scope of work"
To allow for "forecast[ing] of achievement of Milestones and completion of contract events"	To provide "actionable, trustworthy data"
To allow for "estimation of final contract costs"	To ensure that "contract performance data generated are consistently timely, accurate, reliable, and verifiable"

## 2.8 Literature Review: Earned Value Method

The earned value method has been researched extensively in academic literature. The roots of the technique extend as far back as 19<sup>th</sup> century industrial factories, but the formal method began in 1967 as part of the “Cost/Schedule Planning and Control Specification” (C/SPEC) as an outgrowth of the Program Evaluation and Review Technique with Cost (PERT/COST). [12] [84] PERT/COST was developed as part of the U.S. Navy’s POLARIS program. [84] Earned value was developed to address the distortions which result from managing project cost and schedule separately. [38] [39] The earned value method compares the time and cost parameters associated with planned and actual project work. This is similar to the accounting practice of a “flexible budget” which compares whether actual costs for incremental output (scope) are over or under budget. [85] Earned value adds a time dimension to this concept to show whether work is accomplished ahead of or behind schedule. [85] Most early applications of earned value have focused on controlling costs within a project. [12] [86] Later modifications and extensions shifted focus to scope and schedule management. [38] [39] [44] [87] [88] For information on traditional earned value calculations and metrics, see the Project Management Institute’s *Earned Value Management Practice Standard*. [89]

The combined analysis of the planned and actual cost for incremental scope and the planned and actual duration of the activity which creates that scope allows for identification of situations where the time-shifting of work creates the illusion of budget over- or under-runs. [39] Identifying these situations leads to more accurate assessment of the true project cost and schedule status. [39] The principles embodied in the traditional earned value method and the related EVMS criteria are reflective of sound project management and control principles. [90] [91] [92] [93] Their practice is effective if several assumptions hold true: the performance measurement baseline is accurate and includes all work necessary to complete the project, including coordination and rework; progress is measured quickly, accurately, and in a manner which represents its true value; the parties involved are open, honest, and realistic; each cost activity is independent; and the project is not subject to significant uncertainty. [90] [91] [92] [93] Earned value is most effective when employed by an organization with mature and disciplined project management and systems engineering processes, and earned value will not compensate for the lack of either. [39] [91] [94] Earned value has proven reliable in accurately forecasting project cost deviations and final project costs. [39] [49] [85] Earned value also drives robust project planning [85] and alignment between the operational and strategic elements of a project organization. [90]

Government and academic studies have documented the negative impacts of overly bureaucratic and burdensome implementations of EVMS policy. [44] [85] [95] Examples of such ineffective conduct include centralized monitoring and control at inappropriate levels of the project and excessive analysis of variance reports. [39] [85] [95] Earned value data is susceptible to inaccuracies due to human factors such as error, bias, or intentional manipulation. [85] An organizational culture that disincentivizes bad news will drive reporting of overly optimistic earned value data. [85] Earned value is also susceptible to moral hazard, especially when contract incentives are tied to earned value metrics. [79] [90] It may be tempting to solve this problem by abusing an EVMS and focusing on control of the contractor instead of the project.

Earned value is often touted as a method which integrates scope, cost, and schedule. [71] [85] [40] [90] In this context, “scope” refers to “work scope” and not “product scope.” Earned value measures work scope in terms of tasks complete instead of product scope in terms of task outcomes. Task progress is calculated based on the nature of the task and usually involves measuring effort. The basis for crediting progress could be the percentage of items produced or whether a task has started or finished. [89] Measures of how well technical specifications or quality measures are met can also be used as the basis of progress, but they are not required. If technical outcomes are not accounted for when measuring progress, then earned value will not account for these factors. In practice, most implementations of earned value do not include such measures when calculating task progress, and so traditional earned value does not typically account for technical performance measures or quality. [39] [38]

Perhaps the most well-documented shortfall in the traditional earned value method is its inability to accurately report or forecast schedule status. Classic earned value schedule metrics are unreliable, most remarkably for late projects approaching completion because the schedule performance index always converges to perfection. [39] [44] [49] [87] [90] Traditional earned value metrics also do not account for the flow of work or task dependencies and thus do not differentiate between tasks on, near, or far from a project’s critical path. [38] [90]

Several modifications and extensions to earned value have been proposed to address the shortcomings identified in research. Kim and Ballard proposed an extension under the term “Customer Earned Value,” which emphasizes work flow and demand for task output. [38] Solomon and Young proposed an “enhancement” to the earned value method in their book *Performance-Based Earned Value*, which emphasizes technical performance measures and product requirements instead of work. [39] Abba proposes a modification to earned value within a broader “Integrated Program Performance Management” method in order to integrate it more effectively with broader program management systems, focus on business or mission outcomes, and combine it with dynamic scheduling and control. [88]

Multiple modifications to earned value have been proposed to incorporate elements of uncertainty, risk, and imprecision. [47] [96] [97] [98] Seshadri proposed a modification to earned value, which incorporates risk estimates in the performance measurement baseline at the task level using ranges and Monte Carlo simulations to generate probabilistic earned value metrics and a risk-informed reserve. [96] Taha et al formulated an earned value modification using fuzzy number theory in an attempt to improve applicability under uncertain conditions. [97]

In order to address the schedule deficiencies of the classic earned value method, Lipke developed the Earned Schedule extension [87], while Khamooshi and Golafshani developed the Earned Duration extension. [44] Earned Schedule converts traditional earned value measures from units of cost to units of time in order to produce time forecasts and metrics. [87] Earned Duration decouples cost and time measures. [44] It creates a parallel set of time measures that are used to produce time forecasts. [44] Batselier and Vanhoucke conducted empirical research on forecasting project durations, which concluded that the Earned Schedule extension without any performance weighting was superior to the Earned Duration and classic Earned Value (referred



to as Planned Value) methods. [86] Batselier and Vanhoucke then developed a new extension based on Earned Schedule which incorporates the exponential smoothing forecasting approach and the reference class forecasting technique called the eXponential Smoothing-based Method. [99] Chang et al proposed the use of a multiple risk level model with dynamic scheduling in conjunction with earned value to improve schedule management. [47] It uses estimates of project risk in multiple categories to create several project plans and dynamically implements the most appropriate plan based on which risks are realized.

### **3 RESEARCH QUESTION AND APPROACH**

The Department of Defense prescribes the use of Earned Value Management for project control on large, complex engineering projects. Literature shows that earned value produces mixed results, however. It is well suited to cost control and forecasting but fails to address schedule and scope sufficiently. Its potential for success is also greatly influenced by the maturity of the organization's processes.

Project control and the broader discipline of project management to which it belongs are the subjects of active research. Innovative new tools, techniques, and methods are being proposed regularly. In particular, integrated computational project models have the potential to revolutionize project management and project control.

Given that earned value is widely recognized as an imperfect method, it is worth asking whether it is the ideal choice for controlling defense acquisition projects. It is possible that an integrated project model may enable a more suitable project control method. In order to explore these questions, however, it is first necessary to understand what the Department of Defense values in a project control method.

This basis of value will be developed by applying the systems thinking methods presented in Section 2.3. Such a basis of value has never been established for the Department of Defense and will be a significant contribution of this thesis. Establishing this basis of value then allows for the comparison of how well different project control methods meet Department of Defense needs in a manner which has also never been done before.

The research within this thesis involves the following three efforts:

- 1) Analysis of the Department of Defense needs for project control to create a basis for comparison of different methods
- 2) Development of an integrated project control model that can be used to run computational simulations for an experiment
- 3) Conducting an experiment to compare and evaluate selected project control methods using the integrated model and Department of Defense needs

#### **3.1 Project Control Needs**

Section 2.7 introduced the Department of Defense policy documents concerning the prescribed project control approach for large, complex Defense engineering projects: The Earned Value Management System (EVMS). Section 2.8 presented several of the main themes from academic research on the earned value method and its extensions. The first portion of research within this thesis involves analyzing the stated needs of the Department of Defense regarding EVMS. This analysis is placed in the context of the academic research. Applying a systems perspective yields a basis of primary value for EVMS as a project control system. The Department of Defense is the principal beneficiary of the system.

Table 6 lists the reasons given in Department of Defense policy documents for the use of an EVMS. These can be considered expressions of goals that the Department of Defense expects an EVMS to meet. These goals are assumed to meet the project control needs of the Department of

Defense because EVMS is the prescribed project control method. These goals can therefore be considered representative of the project control needs of the Department of Defense on large, complex engineering projects.

Some of the goals are intermediate processes necessary to make other goals possible. In systems terms, these are internal value-related processes necessary for the emergence of an earned value management system’s primary external value-delivery process. The value of this process is defined by the beneficiary, which is the Department of Defense, and it is implied in the set of goal statements taken collectively.

In order to understand the primary value delivered by an earned value management system within a defense project, it is necessary to first define the value-related operand which is transformed or created across a system boundary. [61] The system problem statement for the Defense Acquisition System depicted in Figure 15 of Section 2.6 establishes the context within which an EVMS operates. By decomposing the Defense Acquisition System’s primary value delivery function two additional layers in Figure 16, it is possible to view how the value delivered by an EVMS relates to the value delivered by the Defense Acquisition System.

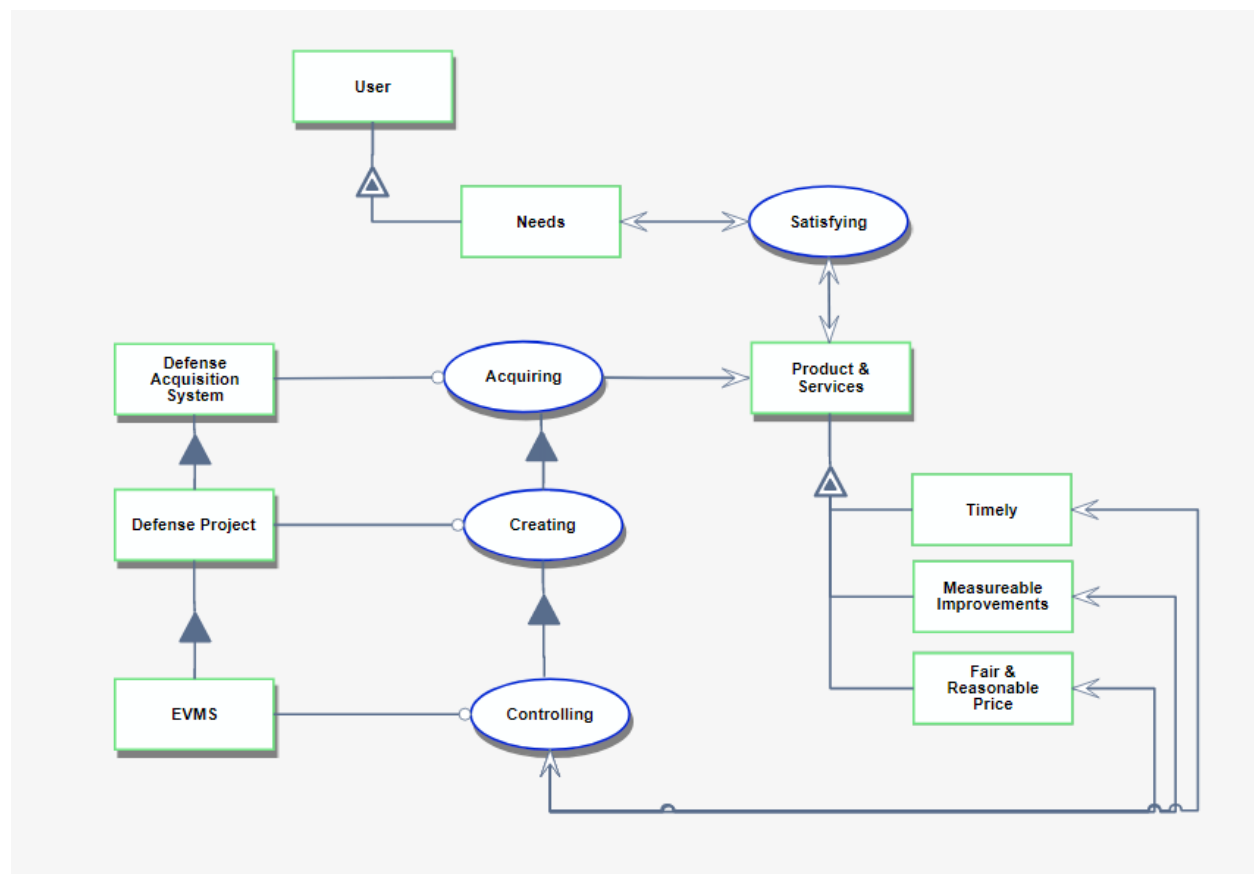


Figure 16: OPM diagram of EVMS within the Defense Acquisition System

The primary value delivery function for the Defense Acquisition System is “acquiring” the product or service. The value-related operand for the Defense Acquisition System is the product or service which is acquired to satisfy the needs of a user (e.g., a downed pilot in need of

communications capabilities). Decomposing a layer reveals a defense project creating a product or service (e.g., a communications satellite) that is timely, affordable, and provides the needed capability improvement. Decomposing yet another layer shows an EVMS controlling the project parameters which result in the timeliness, extent of improvement, and price fairness and reasonableness for the product or service. Those project parameters which ultimately manifest in cost, duration, and scope are the value-related operands of an EVMS, and the primary value function is the controlling of those parameters such that they satisfy the needs of the beneficiary of the project. The detailed internal processes by which the emergent controlling function occurs can be complex and are particular to each project. An abstract view of these processes is shown in Figure 17, constructed using a framework found in [61] and taught in the MIT System Design and Management curriculum.

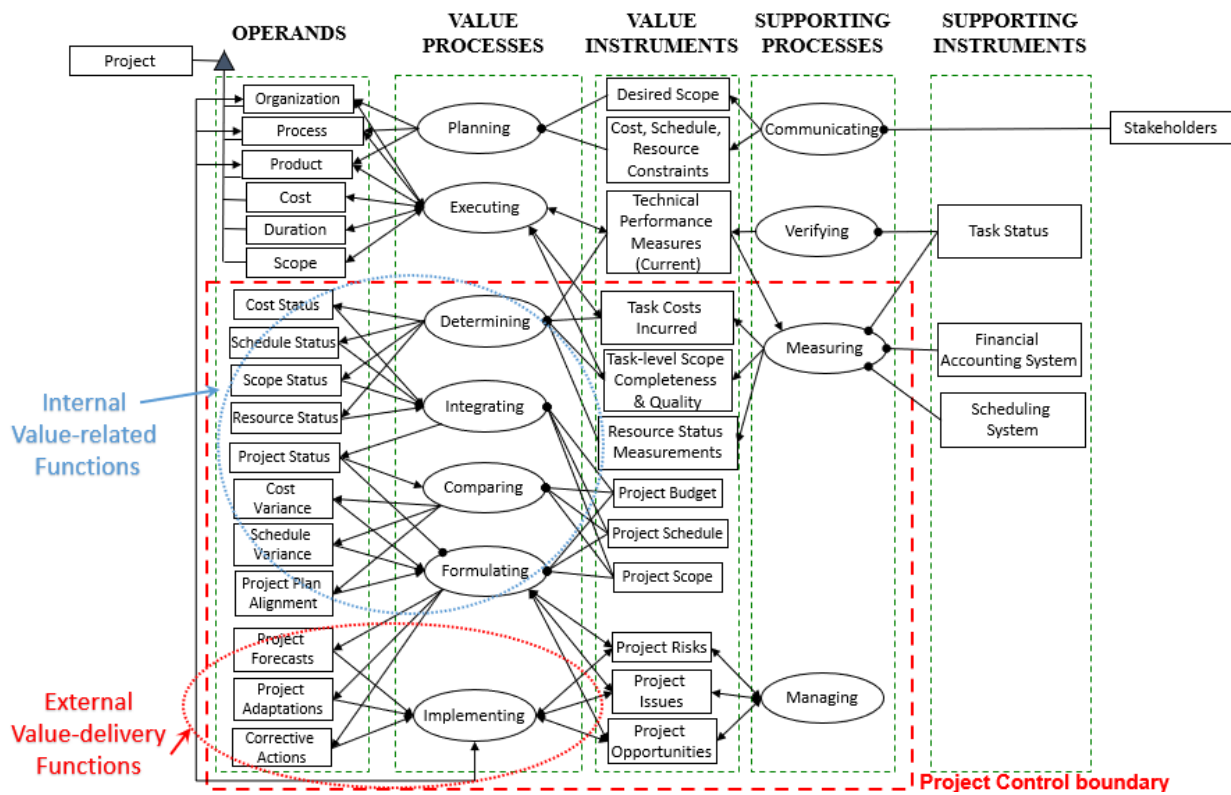


Figure 17: Abstract OPM diagram of a Project Control Process

With an understanding of the primary value-related function of an EVMS - controlling of project cost, duration, and scope - it is possible to analyze the stated goals for an EVMS listed in Table 6. Table 7 shows these goals organized under three broad categories based on the nature of the targeted outcome. These categories are: “Implementing Control Actions,” “Performing Disciplined Processes,” and “Generating Information and Measurements.”

Table 7: DoD Goals for EVMS grouped by value process

Implementing Control Actions	Generating Information & Measurements
To “assess the cost, schedule, and technical performance of contracts for proactive decision-making” [71]	To facilitate the government “develop[ing] relationships with their contractor counterparts to facilitate discussions regarding performance towards completion of the contractual scope of work” [71]
To ensure contractors “use performance information for decision making” [71]	To “provide Joint situational awareness of contract status” [71]
To ensure contractors “implement corrective actions” [71]	To “measure contract performance and progress in an objective manner” [71]
To provide “the [project manager] with an effective decision making tool” [79]	To ensure contractors “integrate scope, schedule, and cost for measuring performance” [71]
To “facilitate risk identification and mitigation” [79]	To ensure contractors “record actual costs incurred” [71]
To allow for “forecast[ing] of achievement of Milestones and completion of contract events” [79]	To ensure contractors “objectively assess accomplishments where work is being performed” [71]
To allow for “estimation of final contract costs” [79]	To ensure contractors “assess variances” [71]
To ensure the government receives performance information that: o "Enables independent government predictions of future cost and schedule conditions" [71]	To ensure that “contract performance data generated are consistently timely, accurate, reliable, and verifiable” [71]
<b>Performing Disciplined Processes</b>	To provide “actionable, trustworthy data” [71]
To “ensure sound planning and resourcing of all tasks required for contract performance” [78]	To “ensure that contractors have adequate management systems that integrate cost, schedule, and technical performance” [79]
To allow for “planning all work scope for the program from inception to completion” [79]	To “ensure that valid cost, schedule, and technical performance information are generated” [79]
To provide “better overall planning, control, and disciplined management of government contracts” [79]	To allow for “objective measurement of progress at the work performance level with EVM metrics” [79]
To ensure contractors “plan all work scope” and “assign all work scope” [71]	To allow for “accumulation and assignment of actual direct and indirect costs” [79]
To ensure government contracts operate subject to “disciplined planning and control processes” [71]	To allow for “analysis of variances or deviations from plans” [79]
To allow for “assignment of authority and responsibility at the work performance level” [71]	To allow for “summarization and reporting of performance data to higher levels of management for action” [79]

To allow for “integration of the cost, schedule, and technical aspects of the work into a detailed baseline plan” [79]	To ensure the government receives performance information that: <ul style="list-style-type: none"> <li>o "Is exactly the same as what the contractor uses"</li> <li>o "Is timely, accurate, reliable, and auditable"</li> <li>o "Relates time-phased budget to scope of work"</li> </ul>
To allow for “disciplined baseline maintenance and incorporation of baseline revisions in a timely manner” [79]	<ul style="list-style-type: none"> <li>o "Measures progress as objectively as practicable"</li> <li>o "Contributes to Acquisition Program Baseline (APB) trade-space decisions" [71]</li> </ul>

The goals under the categories of “Generating Information and Measurements” and “Performing Disciplined Processes” align with the internal value-related functions depicted in Figure 17. Those goals under the category of “Implementing Control Actions” align with the primary external value-delivery function in Figure 17. It can be assumed that the internal value-related functions necessary to perform the primary value-delivery function for a system are also necessary, but these functions do not satisfy beneficiary needs independently. Furthermore, it is possible to decompose internal value-related functions repeatedly to develop an ever-increasing list of valid but unhelpful intermediate processes and operands. It is most useful to focus on those functions which deliver external value to the beneficiary.

The goals listed within the “Implementing Control Actions” category of Table 7 represent the external value-delivery functions which are then supported by the internal value-related functions represented within the other two categories. The aggregation of these goal statements in this context can be summarized to create a project control system problem statement specific to the Department of Defense’s needs:

*To: Make proactive decisions for mitigating risks and implementing corrective actions*

*By: Measuring, assessing, and forecasting integrated cost, schedule, and technical performance information in a timely, accurate, and reliable manner*

*Using: An effective project control system*

The diagram in Figure 18 shows how the project control system contributes to satisfying user needs by creating a product with the value-related attributes of timeliness, measurable improvements, and fair and reasonable price through value-delivery functions that mitigate project risks and modify project parameters.

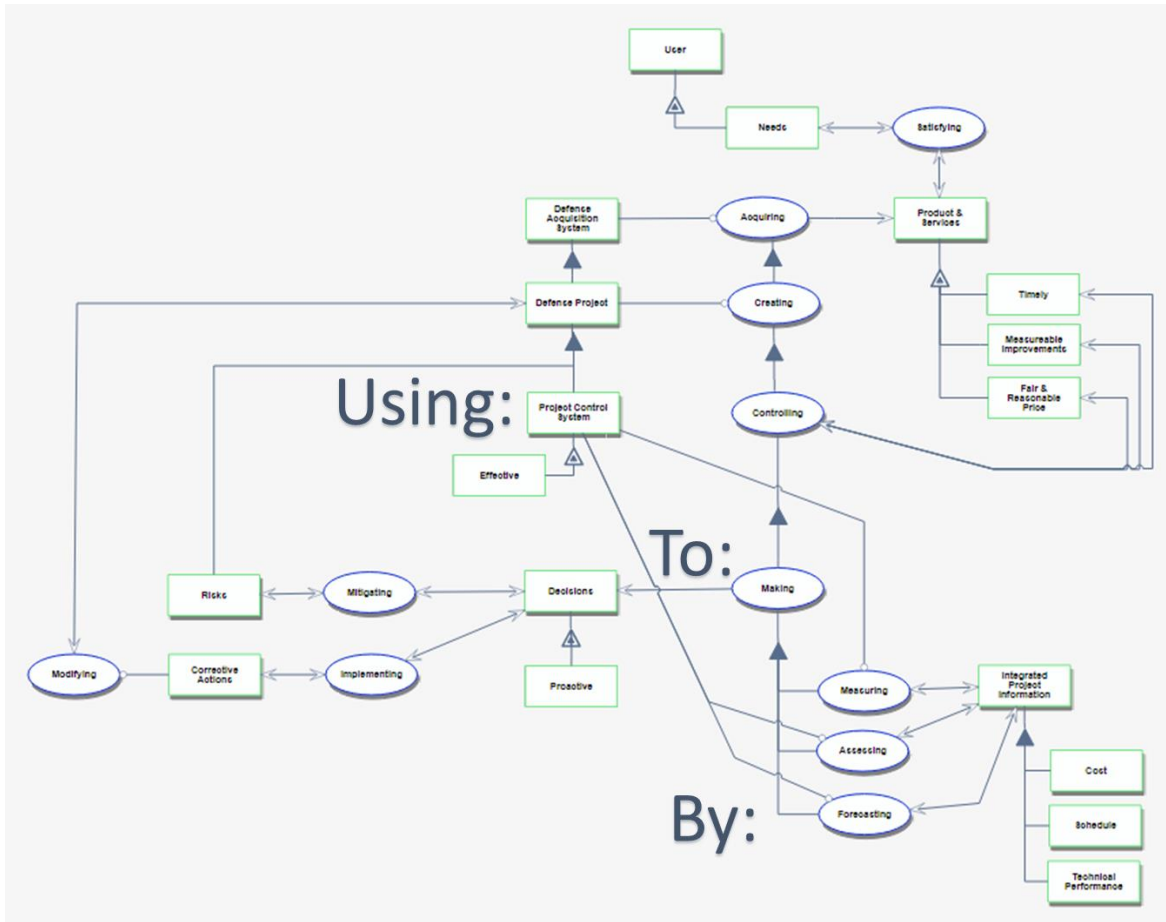


Figure 18: OPM Diagram of Proposed DoD Project Control System Problem Statement

The proposed system problem statement does not include several statements from Table 7 because they are related to the larger project management process instead of project control specifically. The project management process is the whole product system that a project control system operates within. These goals are relevant when discussing the broader whole product system, but are not appropriate to include when discussing the project control system specifically.

When placed in context of the academic literature on the subject of earned value, it becomes clear that there is not necessarily reason to expect an EVMS to meet the goals expressed by the Department of Defense. The Department of Defense goal statements can be divided into three categories: value which is hoped for but has no basis in the established record, value which is targeted and reasonable to expect under certain conditions, and value which can be expected under all conditions.

### 3.1.1 Hopes

*Value which is hoped for but has no basis in the established record*

- a) To allow for "planning all work scope for the program from inception to completion" [79]

- b) To “ensure sound planning and resourcing of all tasks required for contract performance” [78]
- c) To ensure contractors “plan all work scope” and “assign all work scope” [71]
- d) To ensure government contracts operate subject to “disciplined planning and control processes” [71]
- e) To provide “better overall planning, control, and disciplined management of government contracts” [79]
- f) To allow for “disciplined baseline maintenance and incorporation of baseline revisions in a timely manner” [79]
- g) To allow for “assignment of authority and responsibility at the work performance level” [79]
- h) To ensure contractors “use performance information for decision making” [71]
- i) To ensure contractors “implement corrective actions” [71]
- j) To provide “actionable, trustworthy data” [71]
- k) To ensure the government receives performance information that:
  - o "Is exactly the same as what the contractor uses"
  - o "Measures progress as objectively as practicable" [71]
- l) To “measure contract performance and progress in an objective manner” [71]
- m) To ensure contractors “objectively assess accomplishments where work is being performed” [71]
- n) To allow for “objective measurement of progress at the work performance level with EVM metrics” [79]
- o) To facilitate the government “develop[ing] relationships with their contractor counterparts to facilitate discussions regarding performance towards completion of the contractual scope of work” [71]

The statements which fall under the category of “hopes” tend to include superlatives or otherwise rosy adjectives which the study of earned value in practice does not support. Many of these value statements are pre-requisites-for versus results-of effective earned value management. Others are unrealistic when considering the conditions facing real-world projects. None of these statements are specific to the use of the earned value method in particular.

Statements *a) – c)* imply value in having a project plan which includes all of the work activities that will be completed as part of a project. Such a plan implies a deterministic environment and is not realistically achievable because of the uncertainty inherent in projects. If a project could be perfectly planned, there would be no need for a project control process. A better formulation of these goal statements would involve incentivizing a robust planning activity which anticipates risks and uncertainty instead of expecting to plan all work up front. The value in planning is derived as much from the knowledge and capability which is developed during the process itself as it is from plan which is developed. [72] Such a revised goal statement would still be inappropriate for a project control system, however, because the planning process is separate from the project control process. [16]



Statements *d) – i)* speak to the state of the systems engineering and project management processes of the project organization. As was discussed in Section 2.8, mature systems engineering and project management processes are a key determinant in whether an EVMS is effective. [39] [91] [94] Applying an EVMS to a project may provide incentive to improve these processes, but research has shown that earned value does not cause disciplined processes. On the contrary, it is more likely that a newly implemented EVMS will languish because of the poor processes that it may be intended to correct.

Statements *j) – n)* speak to the nature of data generated and measured as part of an EVMS. Each goal uses one or more adjectives to describe data that cannot be attributed to the use of an EVMS based on the existing earned value research. High-quality data is more likely to result from good organizational culture and process discipline than the use of an EVMS. If members of an organization are pressured to present information through “rose-colored glasses,” data distortions will occur while using EVMS. [100]

Statement *o)* speaks to the condition of the relationship between the principle and the agent in circumstances where the project sponsor is from a different organization than the project manager, as is the case in most defense acquisition programs. Nizam reviews principle-agent problems that can arise when using an EVMS which indicate that the EVMS in certain circumstances may actually be counterproductive when attempting to eliminate principle-agent problems. [90]

### 3.1.2 Targets

#### Value which is targeted and reasonable to expect under certain conditions

- p) To ensure the government receives performance information that "is timely, accurate, reliable, and auditable" [71]
- q) To ensure that “contract performance data generated are consistently timely, accurate, reliable, and verifiable” [71]
- r) To provide “the [project manager] with an effective decision making tool” [79]
- s) To “facilitate risk identification and mitigation” [79]
- t) To “assess the cost, schedule, and technical performance of contracts for proactive decision-making” [71]
- u) To “ensure that valid cost, schedule, and technical performance information are generated” [79]
- v) To “ensure that contractors have adequate management systems that integrate cost, schedule, and technical performance” [79]
- w) To ensure contractors “integrate scope, schedule, and cost for measuring performance” [71]
- x) To allow for “integration of the cost, schedule, and technical aspects of the work into a detailed baseline plan” [79]

The statements which fall in the category of “targets” are reasonably achievable goals partly or wholly as a result of using an EVMS in certain circumstances. These are goals which require specific conditions to be met before an EVMS is capable of achieving them. The required

conditions are not routinely observed in projects using EVMS and may require deliberate effort to create them.

Statements *p) – s)* are achievable outcomes which result from using EVMS, but only when the underlying human, organizational, and process factors support them. [39] [91] [94] Several instances have also been documented where an EVMS is used but the resulting data is not timely, accurate, reliable, and auditable. [95] [101] Christensen also illustrates the tension between the reliability and relevance of EVMS data, suggesting tradeoffs which will necessarily be made within the targets listed in *f)*. [85] Whether the use of an EVMS results in proactive decision-making depends on the awareness, ability, and willingness of the decision-makers to act on the information generated. Similarly, organizational and process factors determine whether the information generated by an EVMS can be leveraged for risk identification and mitigation. The guidance given in canonical earned value documentation provides little in the way of how to address problems revealed by earned value analysis beyond the allocation of reserves. [39] [89]

Statements *t) – x)* depend upon the specific methods by which progress is planned and measured within earned value tasks. If the progress measurement accounts for technical performance and risk, then those factors will be incorporated within that specific project's application of an EVMS. Most applications of earned value do not account for technical performance measures or risk when considering task progress, however. [38] [39] [90]

### 3.1.3 Truths

#### *Value which can be expected under all conditions*

- y) To allow for “analysis of variances or deviations from plans” [79]
- z) To “provide Joint situational awareness of contract status” [71]
- aa) To ensure contractors “assess variances” [71]
- bb) To ensure the government receives performance information that:
  - o "Relates time-phased budget to scope of work"
  - o "Contributes to Acquisition Program Baseline (APB) trade-space decisions" [71]
  - o "Enables independent government predictions of future cost and schedule conditions" [71]
- cc) To allow for “forecast[ing] of achievement of Milestones and completion of contract events” [79]
- dd) To allow for “estimation of final contract costs” [79]
- ee) To allow for “summarization and reporting of performance data to higher levels of management for action” [79]
- ff) To allow for “accumulation and assignment of actual direct and indirect costs” [79]
- gg) To ensure contractors “record actual costs incurred” [71]

The statements in the “truths” category are those goals which can be reasonably expected to be met directly as a result of using an EVMS under normal conditions. While there will be instances where these goals are not met by an EVMS, these are the exception instead of the rule.

Statements y) – gg) are reasonable expectations in every instance of EVMS implemented according to ANSI/EIA-748 which involves a government principle-contractor agent relationship. The basic earned value method requires the generation of data which is then used to calculate cost and schedule variances as well as completion forecasts for a project. The data includes planned and actual costs as well as direct and indirect cost collected a level of the project (control account) defined by management and then aggregated to the project level. [89] That data is also provided to the government principal under the terms of the contract which stipulates the use of an EVMS. [102] The government can use that data to create its own cost and schedule forecasts as well as maintain awareness of the project status. Contract status and project status are largely interchangeable terms in this context.

The analysis of Department of Defense project control needs in this section is used to develop a basis for evaluation of the project control methods compared in the third part of this thesis. Considering the proposed project control system problem statement and the previous discussion of the benefits that can be reasonably expected from the use of an EVMS, the following criteria will be used as the basis of evaluation:

- The method integrates cost, schedule, and scope information
- The method provides accurate forecasting information to enable proactive decisions
- The method enables effective risk mitigation

### **3.2 Research Question**

With an understanding of needs, it is possible to assess whether earned value meets those needs and compare the performance of earned value with alternative project control techniques. It is also possible to explore how an integrated project model can be leveraged to provide insights when used to evaluate project control methods.

This thesis attempts to answer the following research question:

*Does a model-based control method achieve better project control outcomes than the earned value management method?*

The basis of evaluation for project control established in the last section is used to measure which outcomes are “better”.

The measurement and control technique for each control method is analyzed to determine whether it integrates cost, schedule, and scope information.

The forecast prediction accuracy determined by experiment indicates whether a method provides accurate forecasting information. Whether those predictions enable proactive decision making is measured by calculating the sensitivity of the prediction results and how much time and money would have been saved if the project was cancelled when the forecast was made.

Each experimental trial produces a priority-weighted normalized project score that combines the cost, schedule, and scope outcomes for the project simulated in the trial. The composite project scores and percentage of trials which meet scope targets for each method indicate the extent to which a method enables effective risk mitigation.

The research Chang et al present in [47] describes a model-based project control technique, the “Multiple Risk-Level” model with dynamic scheduling, that achieved superior project control results relative to traditional earned value for construction projects. This method is a good model-based candidate to compare with an Earned Value Management control method.

The Earned Value Method evaluated in this experiment is the traditional earned value (also referred to as planned value) method for cost performance measures coupled with Lipke’s earned schedule extension for time performance measures. [87] It is not clear what forms of earned value are most commonly practiced within Department of Defense programs, and this should be researched in future work. A future experiment should also add traditional earned value without the earned schedule extension for comparison.

Two additional control methods are also be evaluated. It is good practice to explore a diverse range of alternatives within a solution space. A method where project control activity is only measured and applied at the local task level, referred to as “Simple Control”, serves as a contrast to both the multiple risk level and earned value methods. Additionally, a method that lacks any project control activity, or a “No Control” method, serves as a counter-example and provides a useful reference case.

Uncertainty, measurement accuracy, and process latency are three factors likely to drive a range of project control behaviors and outcomes. These factors are likely to impact the success or failure of any project control method. These three factors are used as variables within the experiment to test the project control performance of each method under varying conditions.

*Uncertainty:* Uncertainty is the fundamental element of risk and arguably the reason that project control is necessary. If projects were deterministic in nature, a perfect plan could be constructed and control would not be necessary. As a project becomes more uncertain, the likelihood of significant variances grows, and the more important identifying and responding to those variances becomes. At the same time, greater uncertainty makes forecasting and thus formulating the optimal control response more difficult.

*Measurement Accuracy:* The accuracy of measuring project status impacts the ability to control a project because the signal given by a measurement is the basis for whether and what type of control action will be taken.

*Control Latency:* Finally, the greater the delay between project events and control processes, the more likely negative effects will compound before being addressed.

### 3.3 Hypotheses

The underlying assumption guiding this research is that the better project control methods lead to better overall project outcomes. Based on this assumption, this section presents three hypotheses predicting the relative performance a control method under specific conditions.

In conditions of no uncertainty, the original project plan should require no control. This assumes the plan was developed with accurate estimates. Active control methods are also least likely to perform well when measurement accuracy is the worst. Since all control methods in the experiment employ the same underlying planning processes, the *No Control* method should theoretically outperform the other control methods under conditions of no uncertainty and low measurement accuracy.

*Hypothesis 1: The “No Control” method will produce superior results when compared to the remaining three methods under conditions of no uncertainty combined with low measurement accuracy.*

Brooks, Mihm et al, and Deveaux each argue for the importance of considering a systemic perspective in control actions. [34] [55] [73] Based on this research, it follows that the *Simple Control* method should produce inferior results when compared to the *Earned Value* and *Multiple Risk Level* methods.

*Hypothesis 2: The “Simple Control” method will produce inferior results when compared to the “Earned Value” and “Multiple Risk Level” methods under all conditions.*

Chang et al argue that a Multiple Risk Level model produces more realistic estimates under conditions of uncertainty when compared to methods that estimate project duration using traditional deterministic or probabilistic critical path calculations. [47] If true, the Multiple Risk Level method should produce superior results when compared to the other methods under conditions of high uncertainty.

*Hypothesis 3: The “Multiple Risk Level” method will produce superior results when compared to the remaining three methods under conditions of high uncertainty.*

### 3.4 Research Approach

The research approach used to test these hypotheses and answer the research question divides into three tasks:

- 1) Develop an integrated project model which can be used to simulate the effects of different project control methods on a project’s outcome
- 2) Conduct an experiment to test the hypotheses and elicit differences between selected project control methods
- 3) Evaluate the results of the experiment to answer the research question

#### 3.4.1 Integrated Project Model

The second portion of research within this thesis involves constructing a computational model which can be used to evaluate project control methods. This model is based on the project

theory, project management process theory, and project control process theory concepts introduced in Section 2. The model structure is designed using Object-Process Methodology and Design Structure Matrices. The model combines several principles and properties from the Project-Process Lifecycle Management framework and Project Design method. Finally, the model is written and implemented in the Python programming language using the Scientific Python Development Environment (Spyder). Appendix B lists details of the tools and libraries leveraged as part of the model. Appendix D contains a copy of the entire model code.

### **3.4.2 Experiment**

The third portion of research within this thesis involves designing and conducting an experiment to test the hypotheses. The experiment generates data which is used to respond to the research question.

The integrated project model discussed in the previous paragraph is used as the platform for conducting the experiment. The model allows for repeated project simulations while controlling or varying project parameters designated in the experimental design. A specific set of parameters are selected to serve as variables with the intent of eliciting a range of project outcomes and project control behaviors. A set of project control methods are incorporated into the model and serve as the subjects of the experiment. The Department of Defense basis of evaluation developed for project control in Section 3.1 is used to evaluate the results of the experiment and accept or reject the hypotheses.

This experiment compares four project control methods while varying three factors relevant to project control outcomes. The four project control methods are: No Control, Simple Control, Earned Value Management with Earned Schedule, and the Multiple Risk Level model. Results are generated via Monte Carlo simulation and compared using figures of merit derived from the basis of evaluation to establish the relative performance of each method.

Uncertainty, measurement accuracy, and process latency are the three factors treated as variables in the experiment. These factors are chosen because of their likelihood to impact the success or failure of any project control method.

### **3.4.3 Experimental Design**

The experiment consists of a series of Monte Carlo simulations. Four control methods are evaluated in the experiment: No Control, Simple Control, Earned Value Management with Earned Schedule, and the Multiple Risk Level model. A simulation is conducted for each combination of settings for the three experimental variables of latency, measurement accuracy, and uncertainty indicated in Table 8. Latency is not varied for the No Control method because there are no control activities upon which to impose a time lag. The experiment for the No Control method consists of the combinations of settings for measurement accuracy and uncertainty.

The experiment consists of 90 Monte Carlo simulations. Each Monte Carlo simulation executes the project model with the same input settings 1,300 times. Random variation in the parameters affected by uncertainty, measurement accuracy, and risk cause variation in the project outcomes within a given simulation. Section 4.1.2 and Appendix B contain more information on these

parameters and the type of random variation. Each simulation consists of 1,000 “outcome” runs that generate project cost, time, and scope results and 300 “forecast” runs that generate project forecast versus actual results.

Table 8: Design of Experiments

Project Control Variables		Control Method			
		<i>No Control</i>	<i>Simple Control</i>	<i>Earned Value</i>	<i>Multiple Risk Level</i>
Latency	None	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>
	Low		<b>X</b>	<b>X</b>	<b>X</b>
	High		<b>X</b>	<b>X</b>	<b>X</b>
Measurement Accuracy	Perfect	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>
	High	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>
	Low	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>
Uncertainty	None	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>
	Low	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>
	High	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>

## 4 MODELING AND SIMULATION

### 4.1 Development of an Integrated Project Model

The information presented in Section 2 was used to construct an integrated project model with the express goal of evaluating specific project control methods. The model was designed using OPM and DSM by combining principles from several modeling frameworks listed in Section 2.3, most notably the Project-Product Lifecycle Management framework and Project Design approach. [29] [19] The model emphasizes the product and process domains. It includes relatively few factors from the organizational domain. The model consists of teams working tasks to create scope items. The teams are agents making decisions within each process that ultimately determine the project outcome. The high-level architecture of the integrated project model is shown in Figure 19.

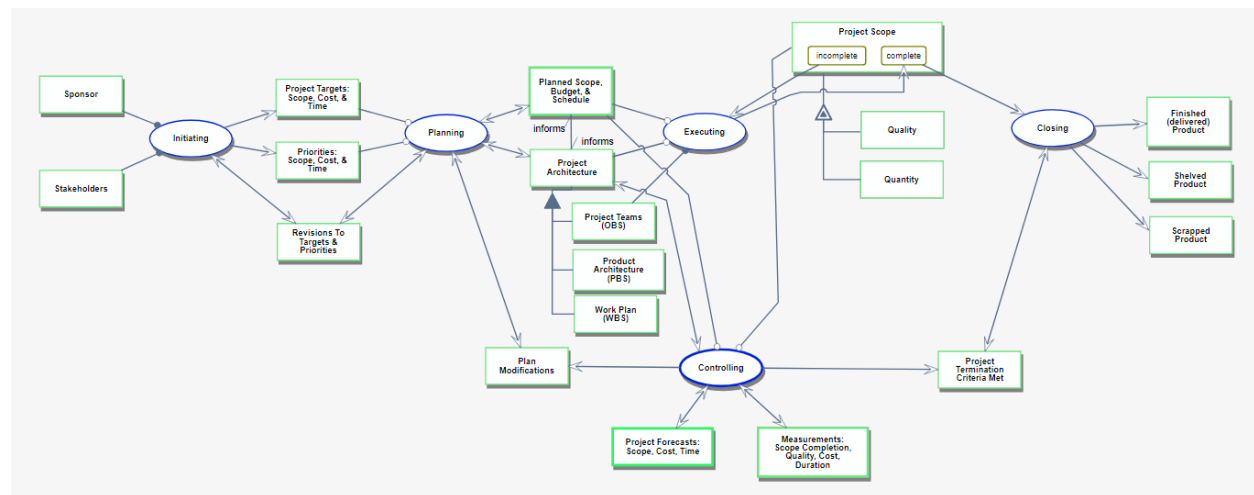


Figure 19: Integrated Project Model Architecture

#### 4.1.1 Model Scope and Concepts

A key challenge to modeling a project is deciding which factors are most relevant and worthy of inclusion in order to represent the behavior of the system. The number of possible factors and their range of values are enormous. Properly modeling them all and accurately portraying their behavior is an immense undertaking beyond the scope of this thesis. This thesis attempts to include enough of these factors to create a reasonable representation of project behavior.

Each element in Figure 19 is represented within the computational model as either a function, data structure, or operation. The five blue process ovals are represented as modules within the model. The model contains task vectors representing the process domain and scope item vectors representing the product domain. The organizational domain is represented by resource and human vectors. The task vectors also include organizational domain information in the form of the assigned team responsible for executing the task and making decisions. Both the task and scope item vectors include information on the resources consumed in scope realization and task execution. An OPM diagram representing the relationship between task, team, resource, and



scope item data structures within the model is shown in Figure 20. Tasks are primarily executed in the execution module to create scope.

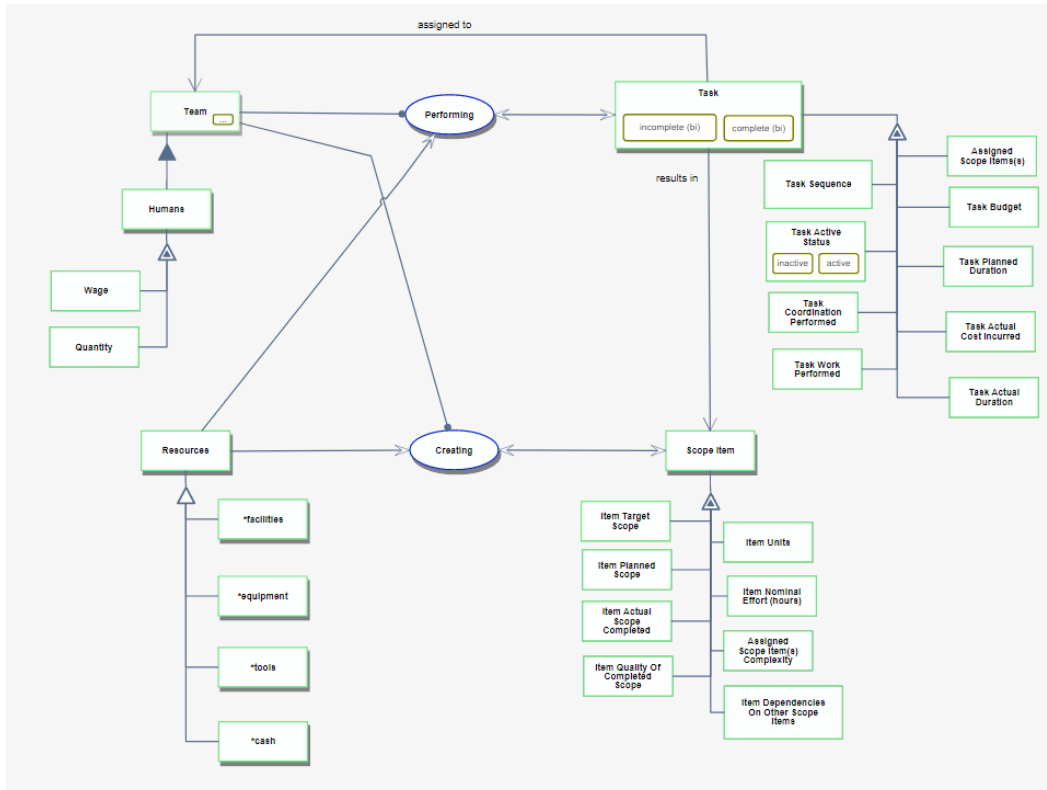


Figure 20: Task, Team, Resource, and Scope Item Vector Architecture

The project simulated by the model is artificially constructed for the purpose of the experiment. It is loosely based on the structure of a satellite engineering project from *Space Mission Engineering: The New SMAD*. [103] The project contains several elements intended to represent complexity found in real-world projects.

### Task Dependency

Task dependencies in the model are represented in two forms. The first type is a finish-to-start relationship, such that the successor requires the predecessor to finish before it may start. The second is a pacing relationship, such that the successor task cannot start until the predecessor begins and cannot finish until the predecessor is complete. These relationships are represented in the task DSM, **D** (see Appendix B).

It is worth noting that the traditional representations of task dependencies used in the model have several shortcomings. The model does not represent the nature of dependency as a demand for coordination. [19] The model also does not allow for violation of task dependencies which would cause exceptional activity. [19] Modifying the model to incorporate these concepts would be an improvement which is an area for future work.

### *Scope Dependency*

In addition to task dependencies, the model includes the concept of dependency for scope which imposes a constraint upon which scope items can be realized through work at a given point in time. If a scope item's predecessors are incomplete, the scope item cannot progress. The quality of predecessor scope items also impacts the quality of successors. These relationships are represented in the scope DSM, **G** (see Appendix B).

### *Scope and Task Relationships*

Several scope items are realized across multiple tasks. Each task is responsible for realizing a fraction of the planned intermediate and target scope quantity and quality. Additionally, two tasks within the project contribute to every scope item in a level-of-effort manner. The fraction of the planned scope item quantity and quality that a task is responsible for is represented in the task-scope Domain Mapping Matrix (DMM), **M** (see Appendix B).

### *Scope, Cost, and Duration as Emergent Outcomes*

Scope realization in the model is an emergent outcome resulting from teams consuming resources to execute tasks with the goal of realizing planned scope values. These planned scope values are the products of interaction between target values, available organizational resources, and stakeholder priorities. Project cost and duration in the model are emergent values resulting from the interaction of scope realization, task execution, teams, and resources.

### *Scope Characteristics: Target and Intermediate Scope, Nominal Effort, Units, Complexity, Quantity, Quality [15]*

Scope represented within the model consists of that which is valued by the beneficiary, or "target" scope, and that which is necessary to achieve the target scope but which the beneficiary does not directly value, or "intermediate" scope. [15] These values are combined within the model to form the total required scope. For example, if five assemblies of subsystem A are required to build the final product (target scope) and an additional three assemblies are required for prototyping and testing (intermediate scope), the total required scope value stored in the model is eight assemblies.

Each scope item is defined by a nominal effort measured in hours which represents the amount of time a qualified worker would take to complete one unit. The scope item units are defined based on the nature of the scope item. For example, units for a component design may be in drawings or files. [15] A complexity value is set for each scope item which defines the amount of coordination necessary to realize the scope in relation to nominal effort. [15] Scope quantity results from work and coordination combined with resources, and scope quality is a function of learning, team size, team exhaustion, resource availability, complexity, and predecessor scope item quality. See Section 4.1.2.3 for details on the quality function.

### *Coordination as Activity*

The model reflects the nature of coordination as an activity required for realization of scope which is proportionate to the complexity of the scope item. [19] Implementation within the model is not calibrated and only represents this concept abstractly.

### *Team Size Effect on Productivity and Quality*

The size of the team working a task affects the productivity of the team members as well as the quality of the scope items that the team is working on. Larger teams reduce the productivity of the individual members as greater amounts of coordination activity is necessary to function. This is represented as a sub-linear constant exponent in the model. This exponent is applied to the team size and the resultant is applied as a productivity factor which has the effect of reducing the amount of productive work and coordination completed.

Larger teams also produce scope at greater relative quality compared to smaller teams, but only to a point. [104] After that point, the benefit fades. This optimal team size for a “quality bonus” is set to 10 in the model, and the benefit disappears completely at team sizes of 1 and 19. The values selected for the model are not calibrated and represent the concept imprecisely. See Section 4.1.2.3 for details on the functions governing the effect of team size on productivity and quality.

### *Effect of Insufficient Resources and Team Exhaustion on Quality*

Conducting work without sufficient resources results in a linearly proportional reduction in scope quality based on a resource penalty constant,  $p_r$ . Implementing overtime (a workday greater than eight hours) also results in a linearly proportional reduction in quality based on an overtime penalty constant,  $p_o$ . The values of these constants are not calibrated and represent the concepts imprecisely.

### *Measured versus Actual Values*

The model represents the differences between the values measured through human activity and perception and the actual values that exist in the real world. Team decisions during project execution are based on measured values, while actual values determine whether the product is accepted or rejected by the customer upon delivery.

### *Measurement Accuracy*

Measured and actual values within the model differ based on normally distributed random variables. The degree to which measured and actual values differ is controlled by a parameter which is varied during the experiment. This parameter sets the magnitude of the variance of the normal random variables which transform actual values into measured values.

The concept of measurement accuracy reflects limitations in the ability to accurately measure progress as a result of several factors including latent defects, reluctance of

personnel to honestly report bad news, and the fact that measurement is an imprecise activity. It incorporates the concept of undiscovered rework in the project. Implementation within the model is not calibrated and only represents this concept abstractly. The model also does not include the ability to implement control activity aimed at increasing measurement accuracy or the detection of rework. The inclusion of an ability to adjust measurement activity would improve the model and is an area for future work.

### *Uncertainty*

Several variables and operations within the model include an element of random uncertainty which is controlled by a parameter that is varied during the experiment. Both normal and uniform distributions are used. A normally distributed random variable affects work and coordination complete, scope item quality realized, and costs incurred while a uniform random variable affects risk realization and the team-size quality bonus. Similar to measurement accuracy, the control parameter sets the level of variance for the random variables.

Uncertainty within the model is intended to represent the unpredictable nature of reality which faces projects as well as the limited ability to accurately estimate project variables. Implementation within the model is not calibrated and represents this concept imprecisely.

### *Risk*

The model includes the possibility of four separate independent risks occurring at some point during the project. These risks are summarized in Table 9. The realization of these risk events is determined by a uniform random variable within the model. These risks cause a diverse set of challenges within the project and provide the conditions for a variety of results as well as differentiation in project control outcomes depending on how well suited a method is to addressing risks.

*Table 9: Project Model Risk Descriptions*

<b>Risk Title</b>	<b>Description</b>
Customer Adds Requirements	15% chance that customer increases requirements (represented by 40% increase in target scope quality standard) after subsystem 1.1 integration has begun
Hiring Freeze	5% chance that no additional hiring will be possible for the duration of the project (Initial hire set to 75)
Late Undiscovered Rework	25% chance that undiscovered rework (errors) in the system design goes undetected until the component fabrication tasks begin
Defective parts	5% chance that all orders of resources contain 15% defective parts

### *Latency*

The concept of latency in the model changes how often project control activity is conducted, how much delay exists between when a value is measured and recorded, and how much delay exists between deciding upon a project control action and when it takes effect. The magnitude of the latency is set by a parameter which is varied in the experiment. Implementation within the model is not calibrated and represents this concept imprecisely.

### *Priorities*

Cost, schedule, and scope outcomes are not necessarily valued equally by project stakeholders. The model records these relative priorities during project initiation and uses them to guide project control activity. When variances occur, they often occur in more than one variable. Teams within the model use the relative priorities of cost, schedule, and scope priorities to decide which variable will be addressed by control activity. For example, if negative variances are observed for both cost and schedule, if the performance indices are equally poor, and if the cost priority is twice the schedule priority, then the control activity will favor cost over schedule performance.

### *Product Delivery and Project Cancellation*

The model establishes cancellation thresholds for cost and schedule during project initiation. It also establishes delivery thresholds for scope. If the cost or duration of the project exceeds the cost or schedule cancellation thresholds, the project organization decides to close the project. This represents a sponsor's level of willingness to accept cost overruns or schedule delays. If the project closes and the target scope does not meet the delivery threshold values for quantity and quality, then the product is not delivered to the beneficiary (i.e., the product is "scrapped"). This represents the customer rejecting the product for failing to meet minimum acceptable requirements.

Additionally, if a project control method forecasts that a project will exceed the cost and schedule cancellation thresholds to an unrecoverable degree, it signals cancellation of the project before that level is reached. Whether the project is recoverable is decided based on a heuristic that requires the project to be at least 25% complete and exceed the cancellation threshold by a margin beginning at 37% and decreasing linearly to zero as the project approaches completion. A cancellation signal is considered accurate if the project reaches the cost or duration cancellation threshold without delivering a minimally acceptable product.

The Multidomain Matrix (MDM) which is composed of the DSMs for the product, process, and organizational domains in addition to the DMMs for the three domain intersections is displayed in Table 10. The scope and task DSMs represent dependencies, with rows as the successor elements and columns as the predecessor elements. The human-resource DSM represents the number of humans and amount of resources available to the project on the diagonal. These are variables set within the model that represent constraints under certain conditions.

Each row and column within the MDM represent a vector in the model containing all the information needed to simulate the conduct of the represented object within the project. The set of scope item vectors,  $\mathbf{p}$ , forms the scope item matrix,  $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_m]$ , and the set of task vectors,  $\mathbf{v}$ , forms the task matrix,  $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n]$  where  $m = 14$  and  $n = 22$  for the chosen project architecture. Selecting a row in the scope item or task matrices gives a vector representing a particular variable within that domain. For example, the actual costs incurred for each task are represented by a row vector in  $\mathbf{V}$ ,  $\mathbf{c}_v$ , which contains the element  $c$  of each task vector  $i$  representing the current actual cost,  $v_{c,i}$ . See Appendix B for details on the contents of the scope item and task vectors.

Table 10: Project Multidomain Matrix

	Scope Items														Tasks														Org									
	1-Project Charter	2-Project Plan	3-Prelim Sys Design	4-Detailed Sys Design	5-Detailed Sub Design 1.1	6-Detailed Sub Design 1.2	7-Component 1.1.1	8-Component 1.1.2	9-Component 1.1.3	10-Component 1.2.1	11-Component 1.2.2	12-Subsystem 1.1	13-Subsystem 1.2	14-System	1-Project Initiation	2-Project Planning	3-Prelim Sys Design	4-Detailed Sys Design	5-Detailed Sub 1.1 Design	6-Detailed Sub 1.2 Design	7-Comp 1.1.1 Fab and Test	8-Comp 1.1.2 Fab and Test	9-Comp 1.1.3 Fab and Test	10-Comp 1.2.1 Fab and Test	11-Comp 1.2.2 Fab and Test	12-Sub 1.1 Integration	13-Sub 1.2 Integration	14-Sub 1.1 Verification	15-Sub 1.2 Verification	16-System Integration	17-System Verification	18-System Assembly	19-System Environmental Testing	20-Project Closeout	21-Program Management	22-System Engineering	Humans	Resources
Scope Items	1-Project Charter	1													90%																					10%		10
	2-Project Plan		1													80%																				15%	5%	20
	3-Prelim Sys Design			1													80%																			5%	15%	20
	4-Detailed Sys Design				1													80%																		5%	15%	30
	5-Detailed Sub Design 1.1					1													80%																5%	15%	15	
	6-Detailed Sub Design 1.2						1													80%															5%	15%	15	
	7-Component 1.1.1							1													80%														5%	15%	100	
	8-Component 1.1.2								1													80%													5%	15%	200	
	9-Component 1.1.3									1													80%												5%	15%	50	
	10-Component 1.2.1										1													80%											5%	15%	30	
	11-Component 1.2.2											1													80%									5%	15%	300		
	12-Subsystem 1.1												1														40%		30%						10%	20%	150	
	13-Subsystem 1.2													1																					10%	20%	250	
	14-System																																		10%	20%	500	
Tasks	1-Project Initiation																																					4
	2-Project Planning														L																							5
	3-Prelim Sys Design															P																						2
	4-Detailed Sys Design																L																					3
	5-Detailed Sub 1.1 Design																	L																			5	
	6-Detailed Sub 1.2 Design																		L	P																	3	
	7-Comp 1.1.1 Fab and Test																			P																	10	
	8-Comp 1.1.2 Fab and Test																				P																	20
	9-Comp 1.1.3 Fab and Test																					P																5
	10-Comp 1.2.1 Fab and Test																						P															10
	11-Comp 1.2.2 Fab and Test																							P														15
	12-Sub 1.1 Integration																																					20
	13-Sub 1.2 Integration																																					15
	14-Sub 1.1 Verification																																					20
	15-Sub 1.2 Verification																																					10
	16-System Integration																																					10
	17-System Verification																																					30
	18-System Assembly																																					15
	19-System Environmental Testing																																					50
	20-Project Closeout																																					5
	21-Program Management																																					5
	22-System Engineering																																					5
Org																																						
	Humans																																					X
	Resources																																					Y

Scope Item (Product) DSM	Scope-Task DMM
Task (Process) DSM	Scope-Human-Resource DMM
Human-Resource (Organizational) DSM	Task-Human-Resource DMM

## 4.1.2 Modules and Functions

It is beyond the scope of this thesis to describe the working of the computational model exhaustively. The original code for the model is contained in Appendix D, allowing for critical analysis, reproduction, or expanded work. More information can be obtained by contacting the author at carsonc@alum.mit.edu. A brief description of the processes that occur during a project simulation is below. More details on the model structure and operation can be found in Appendix B.

### 4.1.2.1 Initiation Module

The initiation process establishes the basic model parameters and data structures, sets the overall project priorities, and establishes the threshold and target values for cost, scope, and duration. Target and cancellation threshold values for cost ( $C_x$ ,  $C_t$ ) and duration ( $T_x$ ,  $T_t$ ) exist only at the project level, while target and delivery threshold values for scope quantity ( $s_x$ ,  $s_t$ ) and quality ( $q_x$ ,  $q_t$ ) exist at the scope item level for those scope items which are deliverable to the beneficiary of the project (strategic scope). Required intermediate scope quantities ( $s_r$ ) and quality thresholds ( $q_r$ ) are also established.

This module makes a quantity of personnel available for teams and defines the project architecture consisting of the task vector structure, scope item vector structure, and Multidomain Matrix. (Scope Item DSM:  $\mathbf{G}$ , Task DSM:  $\mathbf{D}$ , Scope-Task DMM:  $\mathbf{M}$ )

The module also sets several project constants which govern processes throughout the project such as wage level, overhead rates, maximum overtime, team productivity penalty and quality bonus, and model cycle time.

### 4.1.2.2 Planning Module

The planning module is executed to create an initial project plan as well as to replan the project if variances warrant. The module determines the planned scope item quantities ( $s_{p,j}$ ) by estimating the quantity necessary to achieve the target and intermediate scope quantities and qualities. It creates the estimates using a function that approximates the actual quality function. This process represents the project management and systems engineering teams estimating the total scope necessary to achieve the project targets based on experience and past project data. A limitation of the model is that it does not adjust the accuracy of this approximation based on the complexity or novelty of the project scope. Modifying the model to account for decreasing accuracy of estimates as scope complexity increases is an area for future work.

The work necessary to realize the planned scope ( $w_j$ ) is determined by multiplying the planned scope quantities by an estimated value of the nominal effort for each scope item  $j$  ( $e_{nj}$ ). The coordination ( $o_j$ ) necessary to realize each scope item is determined by multiplying the estimated complexity for each scope item ( $x_j$ ) by the necessary work. Adding work and coordination gives the total necessary effort to realize the planned scope ( $e_{s,j}$ ).

$$\mathbf{w} = \mathbf{s}_p \cdot \mathbf{e}_n , \quad \mathbf{o} = \mathbf{w} \cdot \mathbf{x} , \quad \mathbf{e}_s = \mathbf{w} + \mathbf{o}$$



Projecting the vector of necessary effort for all scope items onto an identity matrix and multiplying with the scope-task DMM gives an effort matrix (**E**) where each element ( $e_{jk}$ ) shows the amount of planned effort for each scope item  $j$  assigned to each task  $k$ :

$$\text{diag}(\mathbf{e}_s) * \mathbf{P} = \mathbf{E}.$$

Adding the values of each column gives the total planned effort for each task:

$$\sum_j e_{s,jk} = e_{s,k}.$$

In this way, information from the product domain is translated to the process domain by means of the DMM. The model makes use of the DMM to translate effects from one domain to another and simulate the interactions between them. This is done to translate work and coordination into scope quantity complete and quality state. It is also the mechanism by which tasks are reactivated and teams reassigned upon discovery of needed rework.

The process of planning and replanning scope and task values is shown in Figure 21.

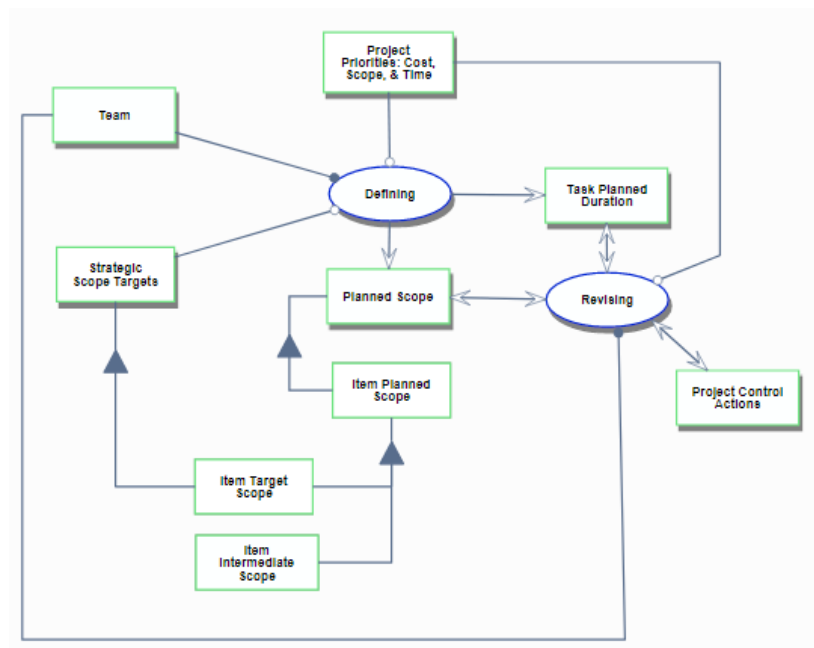


Figure 21: Scope and Task Planning OPM Diagram

Once the planned scope and effort is defined, the planning module uses an iterative heuristic to determine suitable team sizes, calculate task durations based on projected assigned teams, and then calculate total nominal project duration based on the critical path calculated by a precedence diagram algorithm. The heuristic settles on a preferred team distribution that obeys the constraint of total number of personnel hired for the project, calculates the anticipated amount of resources required for the project, and finally calculates a budget. The project plan consists of a project schedule and budget, planned scope quantities for each scope item, team assignments for each task, task budget, task planned resources, planned duration, critical path designation, and planned float for each task.

### 4.1.2.3 Project Execution Module

The project execution module contains the processes which execute the bulk of the project. The time variable iterates with each module call. During execution, teams decide whether a task should be activated based on the available personnel and resources and whether its dependency conditions are met. Once active, the team conducts work and coordination and consumes resources in order to realize scope. Teams conduct work and coordination on a task proportionate to the measured complexity value of the assigned scope items. Work and coordination are collectively treated as effort. Effort is measured in units of hours. Each team member working a task contributes hours of effort to the task they are actively assigned. No team member is assigned to more than one task at a time.

Scope is represented in the project model with state characteristics of quantity and quality. Scope quantity is completed as the productive effort for the tasks contributing to a scope item accumulates. Productive effort is total effort reduced by any exception handling activity and coordination inefficiency. The coordination inefficiency in the model is set to a constant value of 0.9. The quantity  $i_j$  of scope for scope item  $j$  with nominal effort per unit of  $e_j$  assigned to task  $k$  produced by a team of  $n_k$  individuals which have worked  $t_k$  hours on the task with an average rate of exceptional activity of  $z_k$  is calculated as:

$$i_j = \frac{n_k^{0.9} \cdot t_k \cdot (1 - z_k)}{e_j}$$

This assumes that scope item  $j$  is the only scope item assigned to task  $k$ . If multiple scope items are assigned, the team decides whether to split their effort or focus on one scope item based on whether the scope dependencies for each scope item are met. Furthermore, increasing resources beyond the quantity required increases productivity with diminishing marginal returns defined by a constant,  $V_m$ , between 0 and 1. An excess quantity of resources,  $r_e$ , increases productive effort by a factor of  $r_e^{V_m}$ .

The  $i^{th}$  unit of scope for scope item  $j$  assigned to task  $k$  is produced at a level of quality,  $q_{i,j}$ , defined by the function given in Equation 1.

*Equation 1: Scope Item Quality*

$$q_{i,j} = \sqrt{i-1} \cdot \min Q_{p,j} \cdot (1 - p(r_j)) \cdot (1 + b(h_k)) \cdot (1 - u(o_k))$$

Where:

$\min Q_{p,j}$  is the minimum quality of scope items with predecessor dependency relationships to scope item  $j$ ;

*Equation 2: Resource Penalty*

$p(r_j) = \frac{V_r \cdot r_j}{R_j}$  is the resource penalty applied if the available resources assigned to scope item  $j$  ( $r_j$ ) are less than the required amount for that scope item ( $R_j$ ), scaled by a resource penalty constant ( $V_r$ );

Equation 3: Team Coordination Bonus

$b(h_k) = V_q \cdot \left(3 - \frac{(h_k-10)^2}{27}\right)$  is the coordination quality bonus representing the increase in quality that has been shown to result from moderately sized teams [104] with a maximum bonus at a team size of ten which disappears at team sizes of one and nineteen;

Equation 4: Overtime Penalty

$u(o_k) = V_o \cdot o_k$  is the overtime quality penalty where  $o_k$  is the overtime worked (in hours per day) for task  $k$  and  $V_o$  is an overtime penalty constant.

This set of functions incorporates concepts of learning, rework, coordination, dependency constraint, and exhaustion in the realization of scope quality. Rework is addressed in the model by teams accomplishing additional scope quantity at a higher quality to replace previous scope of unacceptable quality. A scope item with poor quality limits the quality of scope items which depend on it. Increasing the overall quality of a scope item by creating additional quantity will reduce the quality constraint flowing to downstream work. Alternatively, it is necessary to accomplish greater quantities of downstream scope to accommodate for latent defects in predecessor items. Undiscovered rework appears in the model when poor quality work goes undetected and must be created when discovered later. This manifests through the concept of measurement accuracy which randomly prevents teams from discovering the true quality value of the scope items they create. This poor-quality scope which limits the quality of downstream scope, which is randomly detected by the teams working it.

Once the measured quantity and quality of the scope assigned to a task exceed the planned and target values, the team completes the task and returns to the pool of available personnel. They can then be assigned to a new task. A diagram of this process is shown in Figure 22.

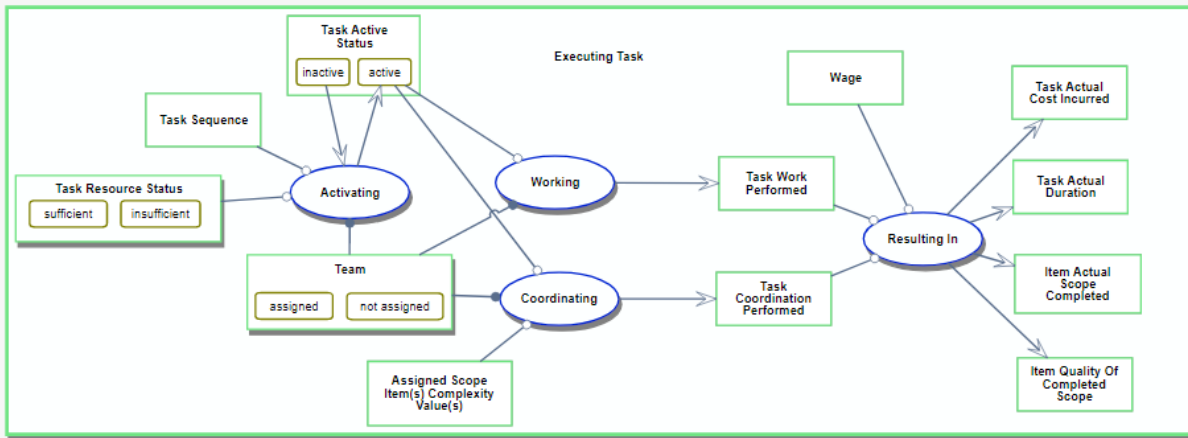


Figure 22: Task Executing OPM Diagram

Decisions are made regarding whether additional personnel should be hired for the project and whether additional resources need to be purchased. Absent a signal from the control module, these actions are only taken if there is a shortfall preventing a task from executing.

The execution module calls the control module on a frequency determined by the *Latency* parameter.

Finally, the execution module measures the project status to determine whether it is ready to close. The closing module is called if 1) all target and intermediate scope values are met, 2) the cost cancellation threshold is exceeded, or 3) the duration cancellation threshold is exceeded.

#### 4.1.2.4 Control Module

The control module consists of functions for the four control methods evaluated in the experiment. The applicable control method is specified at project initiation. Once the project enters the execution phase, the control module is called on a frequency determined by the *Latency* parameter. The control function measures project status, calculates a cost and duration forecast, and implements control actions. The control function also projects whether the project will exceed the cancellation threshold values for cost and duration and signals cancellation if it determines the project is unrecoverable.

Details on the specific implementation of the four project control methods contained within the control module are listed below:

*No Control:* The “No Control” method measures scope quantity and quality values at the task level only. No control activity is implemented. The project executes until the planned scope values are achieved for each task and then closes. The method does not integrate cost or schedule information with scope information. Forecasted values for cost, duration, and scope are the same as the project planned values.

*Simple Control:* Simple control conducts measurement and control activity at the local level only. Each task performs its own monitoring and control activities with the goal of optimizing task results. No system-level measurement or control activity is conducted. Measurements consist of scope quality and quantity, expenditures, and duration at the task level. This method integrates cost, scheduled, and scope information by comparing the task-level measured values for cost, duration, scope quantity, and scope quality to the task-level planned values and then implementing control actions to favor the variable which is performing most poorly compared to the plan. This method does not anticipate risks. Possible control activity consists of raising/lowering scope quality thresholds, increasing/decreasing team size, or increasing/decreasing hours worked per day at the task level. Forecasted values for cost, duration, and scope are projected at the task level and then aggregated to create project forecasts.

*Earned Value Management with Earned Schedule:* Measurement and control activity is conducted according to earned value methods. Control accounts which aggregate several tasks are established, and monitoring and control is conducted both at the control account level and at the project level. Cost metrics are derived based on the standard earned value method as defined by the PMI Earned Value Practice Standard. [89] Time metrics are derived based on the Earned Schedule extension developed by Lipke using the set of tasks on the critical path calculated by the precedence diagram method. [87] These EVM metrics integrate cost, schedule, and work scope (effort). Control activity is selected to

favor the variable which is performing most poorly compared to the plan. Risks and uncertainty are anticipated by applying reserves at the project and task level. Control activity is implemented at either the local control account level or the global project level. Possible control activity consists of raising/lowering scope quality thresholds, increasing/decreasing team size, or increasing/decreasing hours worked per day. The method can also increase or decrease the amount of resources applied at the project level only. Forecasted values for cost and duration are based on the project estimate at completion and project time estimate at completion, calculated according to the traditional earned value method and Earned Schedule extension respectively. Forecasted values for scope are the same as the project planned values. The project is replanned if project cost or duration forecasts exceed 150% of the planned cost and schedule.

*Multiple Risk Levels:* Measurement activity is conducted at the local and the project levels using earned value metrics. Control activity is implemented based on modified version of the multiple risk level model proposed by Chang et al which calculates a risk-based critical path and uses dynamic scheduling. [47] The most likely risks for the project are estimated during the planning phase and multiple plans are developed based on a “stacking” of risks resulting in a series of plans ranging from ideal conditions to worst-anticipated-case. The project plan placed into effect is dynamically selected based on which plan most closely matches the measured project values. Forecasted values for cost, duration, and scope are calculated in the same manner as for the earned value control method. Project-level controls are also implemented in the same manner as the earned value control method, but no control-account-level controls are implemented.

Table 11: Control Activity Taken by Control Methods

Method	Control "Levers"
No Control	- None
Simple	- Increase/Decrease Scope Quality Thresholds - Increase/Decrease Team Sizes - Increase/Decrease Hours Worked
EVM-ES	- Increase/Decrease Scope Quality Thresholds - Increase/Decrease Team Sizes - Increase/Decrease Hours Worked - Increase/Decrease Resources Applied
MRL	- Increase/Decrease Scope Quality Thresholds - Increase/Decrease Team Sizes - Increase/Decrease Hours Worked - Increase/Decrease Resources Applied

#### 4.1.2.5 Closing Module

In the closing module, the customer reviews the final product and accepts or rejects it based on whether it meets the minimum delivery scope criteria. The final project cost is the sum of labor,

overhead, and resource costs. A normalized score is calculated for each variable in the triple constraint based on the project target values. A composite project score which is an average of the individual normalized scores is also calculated.

### **4.1.3 Model Validation**

#### *4.1.3.1 Project Factors Omitted*

While evaluating a model's potential to represent complex engineering project outcomes, it is important to first consider what the model does not include. The relevant factors included in the model supporting this thesis were reviewed in the previous section. This section will summarize the factors acknowledged as omitted. It is important to note that many additional factors are surely unacknowledged as well. Engineering projects are complex sociotechnical systems, and an exhaustive list of relevant factors does not exist. Identifying and classifying these factors is the subject of ongoing project research. [25]

##### *Stakeholder Priority Negotiations*

All phases of real-world projects involve negotiations with stakeholders to establish and revise the priorities and targets which guide the decisions made throughout the project. These negotiations reflect an analysis of the project trade space in light of feasible outcomes defined by scope targets, project status, organizational resources and capabilities, and cognitive limitations. The model does not include any representation of these negotiations and assumes project priorities and targets are fixed.

##### *Scheduling, Cost Estimating, and Funding Processes*

The model simplifies the scheduling and cost estimating processes, which are extensive and complex systems unto themselves in most project organizations. Instead, scheduling and cost estimating within the model is abstracted to reflect the imprecise approximations that result from these complex processes. The concept of a funding process which ensures cash is available for project execution is omitted entirely.

##### *Social and Cultural Factors*

The model does not represent the complex social and cultural factors present in real-world projects. These factors include team dynamics, team cognition, the effect of leadership and management activity on process and product outcomes, political factors within organizations, organizational competencies and capabilities, and many others. These factors are highly relevant to project outcomes and are the subject of debate and ongoing research. [20] [21] [22] [23] [24] [25] It is beyond the scope of the model to incorporate these factors, but their inclusion would be valuable and is an area for future work.

##### *Human Factors*

The model does not incorporate human factors such as specialization (e.g., skills, abilities, domain knowledge), attitude, experience, interest, attention, incentive, motivation, or limits to cognition (with the exception of those represented in the concept

of measurement accuracy). These factors are relevant to project outcomes but are omitted from the model. It is beyond the scope of the model to incorporate these factors, but their inclusion would be valuable and is an area for future work.

#### *Resource Characterization*

The model represents resources generically and interchangeably as “resource units” which are required and consumed to complete tasks and realize scope. The model omits any special characterization of resources such as tools, vehicles, equipment, facilities, cash funds, supplies, parts, or materials. The model also does not incorporate a concept of durable resources which can be returned to the resource pool and reused. These concepts are relevant to project outcomes but are omitted from the model. Characterizing resources more specifically and including the related mechanics to represent the logistical considerations and effects of specialized resource requirements, use, and consumption would be valuable and is an area for future work.

#### *4.1.3.2 Model Settings*

The model includes several customizable parameters. The model treats some of these parameters as constants for all simulations, others as internal control variables, and yet others as controlled variables for the experiment. Appendix C includes a complete list of base project parameters and the values they take in the experimental trials.

The controlled experimental variables are uncertainty, measurement accuracy, and control latency. The values for these variables represent project conditions that range from ideal to poor when viewed from the perspective of the teams executing the project. Table 12 lists the possible values for these variables.

*Table 12: Experimental Variable Settings*

<b>Conditions</b>	<b>Uncertainty</b>	<b>Accuracy</b>	<b>Latency</b>	<b>Value</b>
Ideal (Base)	None	Perfect	None	0
Moderate	Low	High	Low	0.25
Poor	High	Low	High	1

When considering combinations of experimental settings, it is useful to refer to the sum of the combined values. For example, the combined settings of Low uncertainty, Perfect accuracy, and High latency would yield a combined setting value of 1.25. This approach will be used in the results section to evaluate performance under aggregate experimental conditions.

#### *4.1.3.3 Validation Trials*

The theory of the triple constraint is arguably the most basic concept in project management. [14] It states that the three triple constraint variables of cost, schedule, and scope exist in tension. All else being equal, an improvement in one variable necessarily must be accompanied by a worsening in one or both of the other two.

A series of validation trials explore the model’s behavior across a range of values for key model parameters. The goal is to confirm the model behavior’s consistency with the triple constraint. Table 13 lists the key parameter settings used for each validation trial. Appendix C includes complete list of validation parameters and their value settings for each trial.

Table 13: Validation Trial Settings

Validation Trial Name	Scope Quality Targets						Scope Quantities					Team Size					Resources Required					Structure			Uncertainty			Accuracy					
	XL	Low	Base	Mod	High	XH	XL	Low	Mod	High	XH	XL	Low	Mod	High	XH	XL	Low	Mod	High	XH	Flat	Mixd	Full	Base	Low	High	Base	Low	High			
Quality Validation		X		X	X				X					X					X						X			X			X		
Quantity Validation			X				X	X	X			X					X								X			X			X		
Team Validation			X					X				X					X								X			X			X		
Resource Validation			X					X				X					X	X	X						X			X			X		
Structure Validation			X					X				X					X					X	X	X	X			X			X		
Edge Case Validation	X				X	X					X	X				X	X							X	X		X	X		X			
Uncertainty Validation			X					X				X					X								X			X	X	X	X		
Accuracy Validation			X					X				X					X								X						X	X	
U-A Interaction Validation			X					X				X					X								X			X	X		X	X	

The base setting is the “moderate” value for those parameters where one is not specifically indicated. Table 14 describes the behavior expected during these validation trials and whether it was met.



Table 14: Validation Trials Summary

Trial Name	Model Settings	Expected Behavior	Met										
Quality Validation	Low, Moderate, High Quality Targets	Duration & cost increase with quality	Yes										
Quantity Validation	Low, Moderate, High Quantity Targets	Duration & cost increase with quantity	Yes										
Team Validation	Low, Moderate, High Team Sizes	Duration decreases, cost increases as team size increases	Yes										
Resource Validation	Low, Moderate, High Resource Allocations	Duration decreases, cost increases as resource increases	Partial										
Structure Validation	Flat, Mixed, Full-Network Dependency Structure	Duration decreases as structure becomes more networked	Yes										
Edge Case Validation	Low and High Extreme Cases	Results exist at extremes for cost and duration	Yes										
Uncertainty Validation	Low and High Uncertainty	Cost and duration increase with uncertainty	Yes										
Accuracy Validation	Low and High Accuracy	Cost and duration increase as accuracy decreases	Yes										
U-A Interaction Validation	<table border="1"> <thead> <tr> <th><u>Uncertainty</u></th> <th><u>Accuracy</u></th> </tr> </thead> <tbody> <tr> <td>Low</td> <td>Low</td> </tr> <tr> <td>Low</td> <td>High</td> </tr> <tr> <td>High</td> <td>Low</td> </tr> <tr> <td>High</td> <td>High</td> </tr> </tbody> </table>	<u>Uncertainty</u>	<u>Accuracy</u>	Low	Low	Low	High	High	Low	High	High	Highest duration and cost with high uncertainty and low accuracy	Partial
<u>Uncertainty</u>	<u>Accuracy</u>												
Low	Low												
Low	High												
High	Low												
High	High												

The expected behavior was observed for each trial except two. The resource validation trial resulted in higher costs when resources were increased, but there was no change in the project duration. Instead, the quality value of the target scope increased. This reveals that adding resources in the model improves quality outcomes but does not improve duration outcomes. The interaction validation trial evaluated the effects of various uncertainty and measurement accuracy settings in combination. Three of these combinations (Low/Low, High/Low, and High/High) resulted in a cancelled project without delivering the product. This indicates that either low accuracy or high uncertainty is likely to result in a poor project outcome. The combination of low accuracy and high uncertainty produced results similar to each independent setting. The validation results are plotted in Figure 23 and Figure 24. The size of each point on the plot represents the scope quality value. Figure 23 omits the extreme edge cases to allow a scale conducive to viewing the remaining trials. Figure 24 adds the edge cases on a logarithmic scale which highlights the extreme nature of the results.

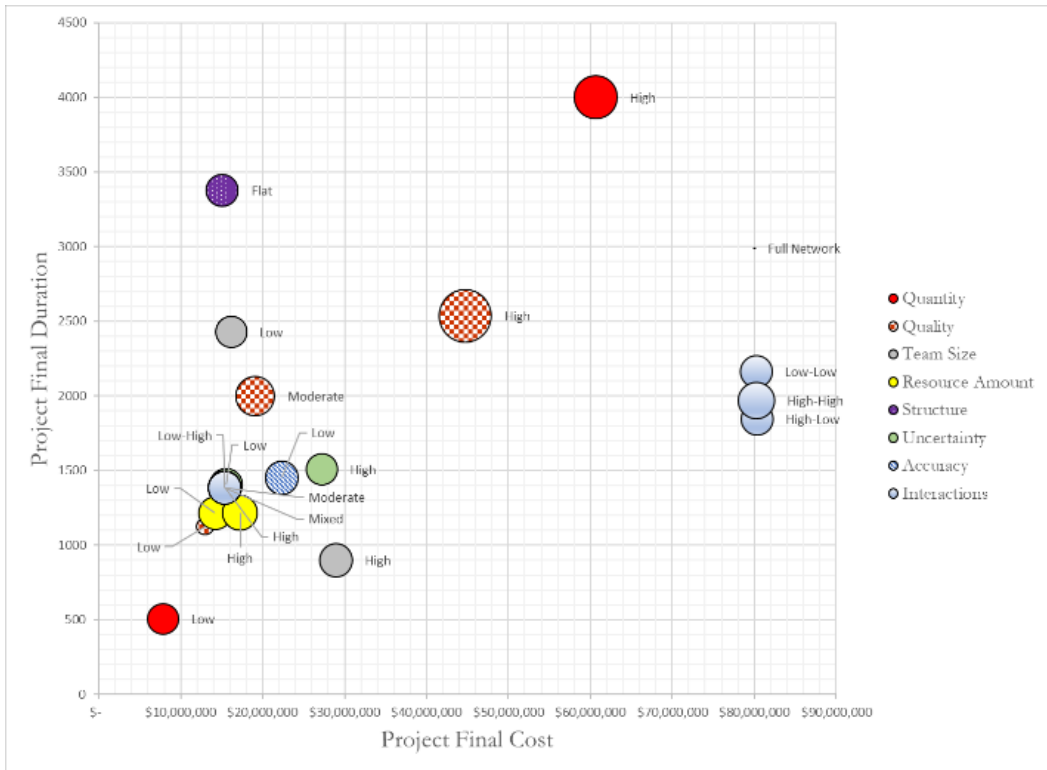


Figure 23: Validation Trial Results with No Edge Cases

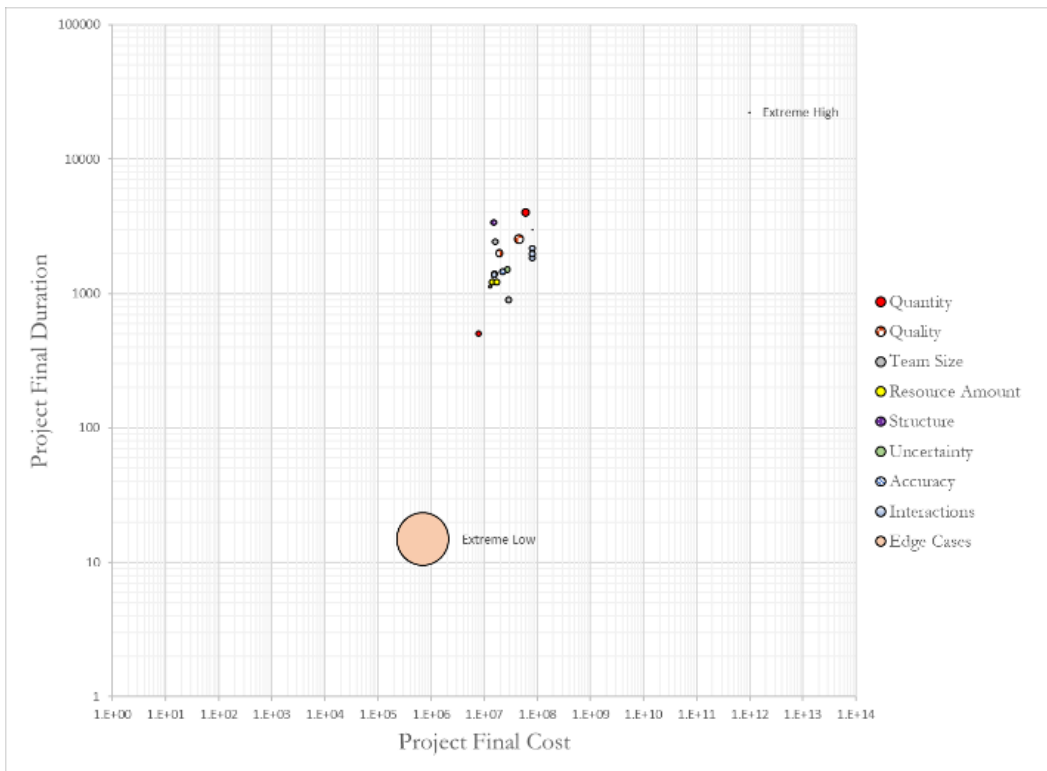


Figure 24: Validation Trial Results with Edge Cases

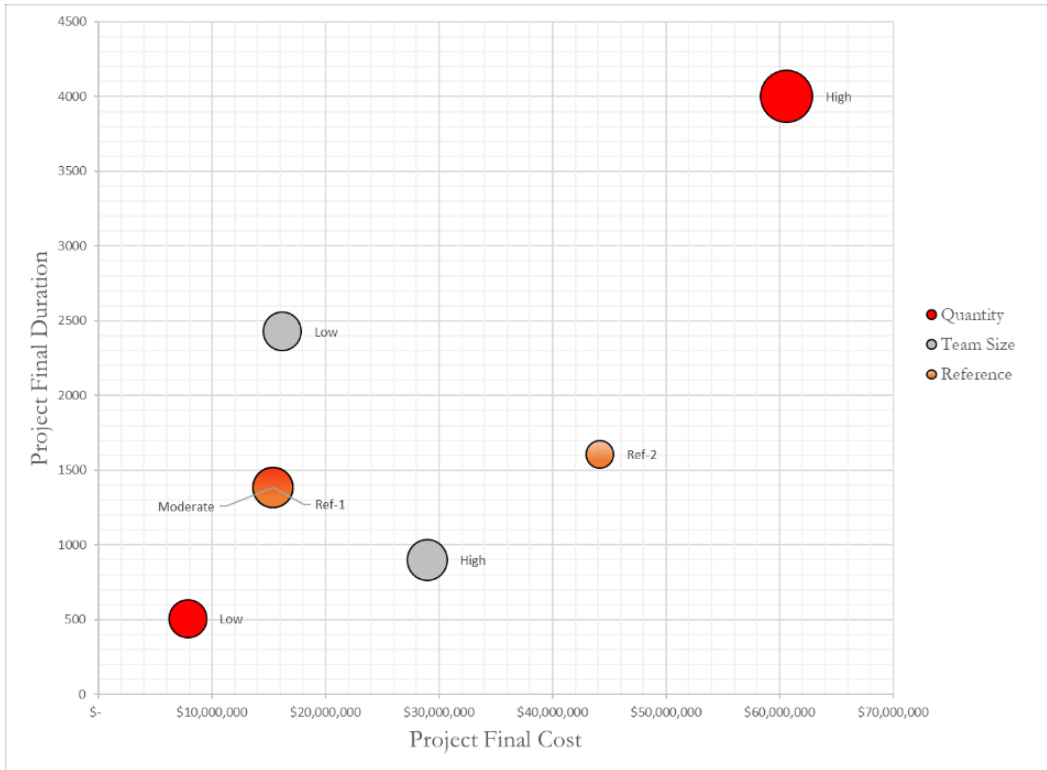


Figure 25: Reference Cases

Figure 25 adds two reference points which will be used to anchor plots in the results section. Reference Simulations 1 and 2 are simulations conducted using all of the base model settings for the experiment detailed in Appendix C. No control method is applied for either case, and each runs until the measured scope values meet or exceed the target values. The likelihood of all risks is set to 0% for the first reference simulation. The likelihood of all risks is set to 100% and all four risks within the model are realized for the second reference simulation. Table 15 shows the experimental variable settings for each simulation.

Table 15: Reference Case Variable Settings

Case	Variable	Value
Reference Case 1	Uncertainty	None
	Measurement Accuracy	Perfect
	Latency	None
Reference Case 2	Uncertainty	High
	Measurement Accuracy	Low
	Latency	High

## **4.2 Experimental Simulations**

The four project control methods were evaluated by executing 1,000 project outcome trials and 300 forecast trials for each simulation indicated in Table 16. The project outcome trials generated project cost, schedule, and scope data while the forecast trials generated predicted versus actual data concerning whether a project should be cancelled.

For each trial, the model runs until the project either delivers a product that meets the scope targets, exceeds the cost cancellation threshold, or exceeds the scheduled cancellation threshold. The project then closes and the model records the final results. The model sets the control method and values of the experimental variables according to the simulation settings. Each control method is represented by a specific set of functions in the model's control module. All other modules and parameters are treated the same for all trials.

Table 16: Experiment Simulations

Simulation	Variable Settings			Control Method			
	Latency	Accuracy	Uncertainty	<i>None</i>	<i>Simple Control</i>	<i>Earned Value</i>	<i>Multiple Risk Level</i>
1	None	Perf	None	X	X	X	X
2	None	Perf	None	X	X	X	X
3	None	Perf	None	X	X	X	X
4	None	High	Low	X	X	X	X
5	None	High	Low	X	X	X	X
6	None	High	Low	X	X	X	X
7	None	Low	High	X	X	X	X
8	None	Low	High	X	X	X	X
9	None	Low	High	X	X	X	X
10	Low	Perf	None		X	X	X
11	Low	Perf	None		X	X	X
12	Low	Perf	None		X	X	X
13	Low	High	Low		X	X	X
14	Low	High	Low		X	X	X
15	Low	High	Low		X	X	X
16	Low	Low	High		X	X	X
17	Low	Low	High		X	X	X
18	Low	Low	High		X	X	X
19	High	Perf	None		X	X	X
20	High	Perf	None		X	X	X
21	High	Perf	None		X	X	X
22	High	High	Low		X	X	X
23	High	High	Low		X	X	X
24	High	High	Low		X	X	X
25	High	Low	High		X	X	X
26	High	Low	High		X	X	X
27	High	Low	High		X	X	X

## 5 RESULTS

This section reviews the results of the experimental computer simulations. The experiment consisted of 90 Monte Carlo simulations of 1,000 project outcome trials and 300 project forecast trials. The project outcome trials provide the basis for evaluating the project cost, time, and scope outcomes. This data helps identify performance trends attributable to a given project control method. The project forecast trials measure the predictive power of each control method. Each control method predicts whether a project should be cancelled in a different way. If the project ultimately does not result in a delivered product before cancellation thresholds are reached, the project should have been cancelled. The difference between the final cost and time and the cost and time when the prediction was made are considered potential savings. This assumes the control method's prediction would have been used to make a decision to cancel the project, and the difference is what would have been saved. This section will present the simulation results for each method followed by the combined results for all methods.

Each trial returns a composite Project Score calculated by a multi-attribute utility function. The project score metric is calculated according to the following formula:

$$Project\ Score = \frac{1}{3} \left( p_c \left( 1 - \frac{C_f}{C_t} \right) + p_t \left( 1 - \frac{T_f}{T_t} \right) + p_s \left( \frac{q_{14,a} - q_{14,t}}{q_{14,t}} \right) \right)$$

$C_f$  and  $T_f$  are the final the final project cost and duration determined once the project has closed.  $C_t$  and  $T_t$  are the target project cost and duration set during project initiation.  $q_{14,a}$  is the overall actual system quality.  $q_{14,t}$  is the target system quality.  $p_x$  is the relative priority for cost, schedule, and scope set during project initiation and the three  $p_x$  values must add to 3.

This function combines the normalized cost, schedule, and quality outcomes for the project into a priority-weighted normalized composite score centered on zero. Positive values indicate better-than-target performance while negative values indicated worse-than-target performance. A project score of 1 implies the priority-adjusted equivalent of all three project outcome variables improve upon their targets by 100%. For cost and schedule, this implies unobtainable values of 0 while the scope quality value would be twice its target. Since the project score is a composite of the three variables, there are infinite possible individual values for each variable that result in the same composite score. The scenarios called out in Figure 26 are three possible cases.

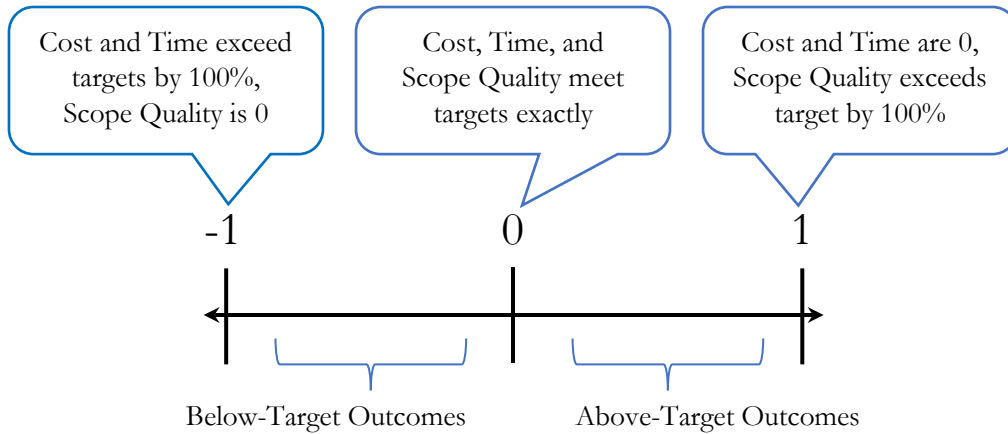


Figure 26: Composite Project Score Regions

Table 17 is an example of a “Confusion Matrix” which displays forecast results. The accompanying metrics give an indication of overall prediction accuracy, sensitivity (true positive rate), specificity (true negative rate), and the complementing positive and negative prediction error rates. The sensitivity measures how well a method predicts the need to cancel while the specificity measures how well a method predicts that a project should continue.

Table 17: Example Confusion Matrix

Confusion Matrix		Actual	
		0: Does Not Exceed Thresholds	1: Exceeds Cancellation Thresholds
Predicted	0: Do Not Cancel	True Negative Results	False Negative Results
	1: Cancel	False Positive Results	True Positive Results
Prediction Metrics			
Accuracy		True Results ÷ All Results	
Sensitivity (TPR)		True Positives ÷ Actual Positives	
Specificity (TNR)		True Negatives ÷ Actual Negatives	
False Negative Rate		False Negatives ÷ Actual Negatives	
False Positive Rate		False Positives ÷ Actual Positives	

## 5.1 No Control

The results for the No Control method indicate a close correlation between cost and duration values. The red markers in Figure 27 show which simulations did not meet the scope targets established for the project. The reference simulations from Figure 25 are labeled in this and subsequent charts.

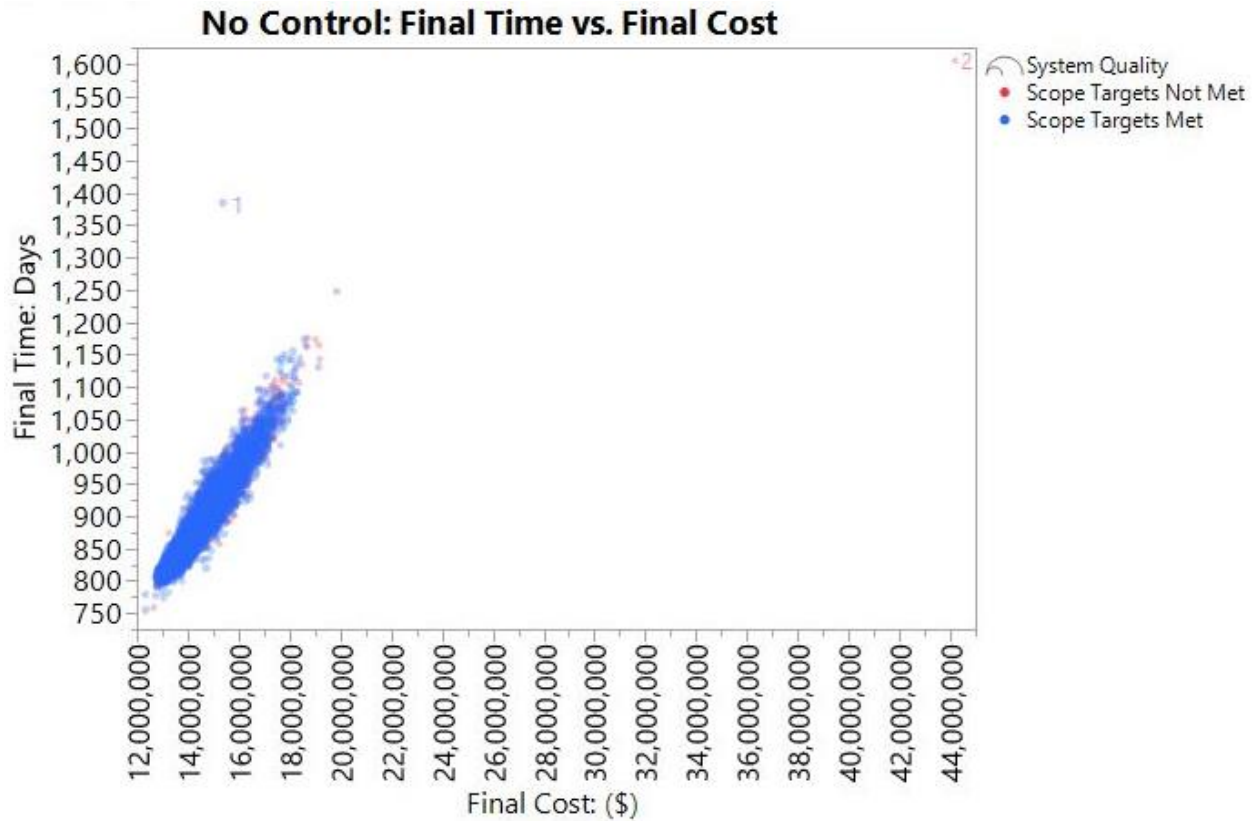


Figure 27: No Control Scatter Plot - Scope Targets Met by Cost and Time

Figure 29 shows the relative impact of uncertainty and accuracy upon the composite project score. The vertical orientation of the contours reveals a significantly greater sensitivity to changes in uncertainty settings compared to measurement accuracy settings.



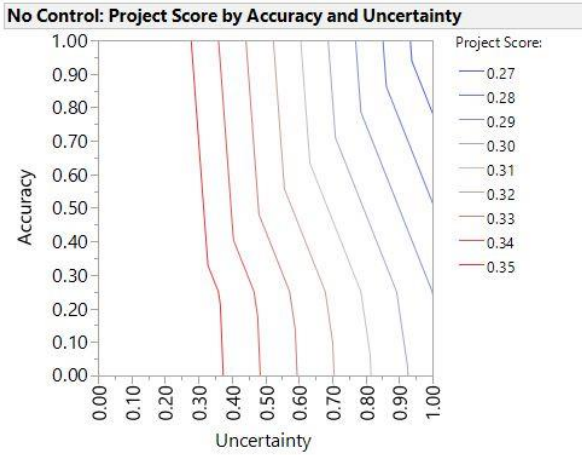


Figure 29: No Control Contour Plot – Project Score by Accuracy and Uncertainty

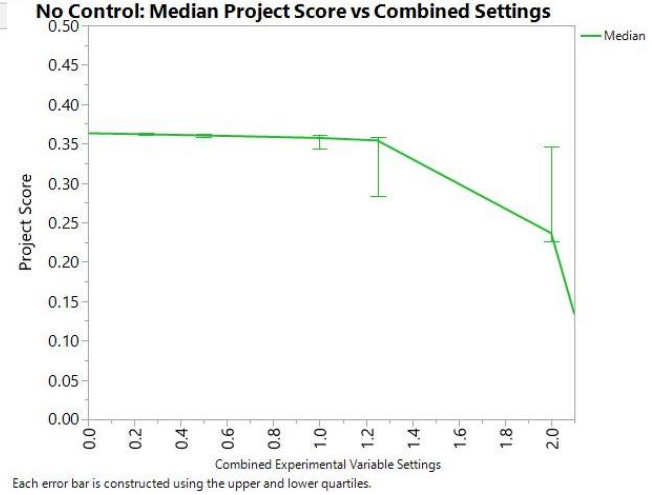


Figure 28: No Control - Median Project Score vs Combined Settings

Figure 28 shows a decline in median project score and a widening in outcomes as the combined experimental variable settings vary from ideal conditions to poor conditions.

Table 18 displays the forecasting results for the No Control method. The “Confusion Matrix” shows whether the method forecasted the need to cancel the project (1) or not (0) in the rows. The actual result of the simulation is shown in the columns. Since the No Control method simply executes according to the project plan, it does not anticipate the need to cancel the project. The No Control project trials did not result in any projects that exceeded the cost or duration cancellation thresholds

Table 18: No Control Confusion Matrix

Confusion Matrix		Actual	
		0	1
Predicted	0	2700	0
	1	0	0
Prediction Metrics			
Accuracy		100%	
Sensitivity (TPR)		N/A	
Specificity (TNR)		100%	
False Negative Rate		N/A	
False Positive Rate		0%	

## 5.2 Simple Control

The results for the Simple Control simulations show two strata which are not clearly explained by any of the experimental variables. It is possible the levels in the data are caused by the discrete realization of risk within the model. The model does not record whether risks are realized in the course of a simulation, and so it is impossible to determine with certainty.

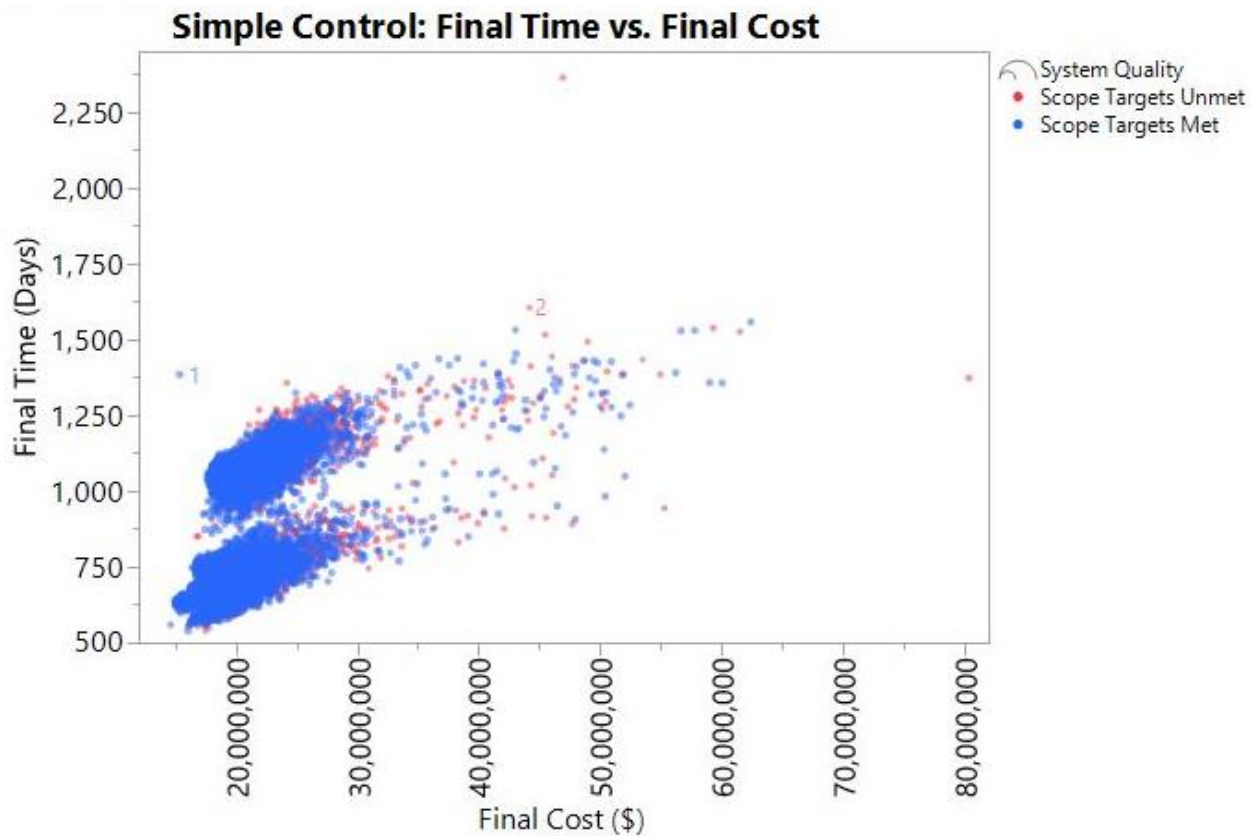


Figure 30: Simple Control Scatter Plot - Scope Targets Met by Cost and Time

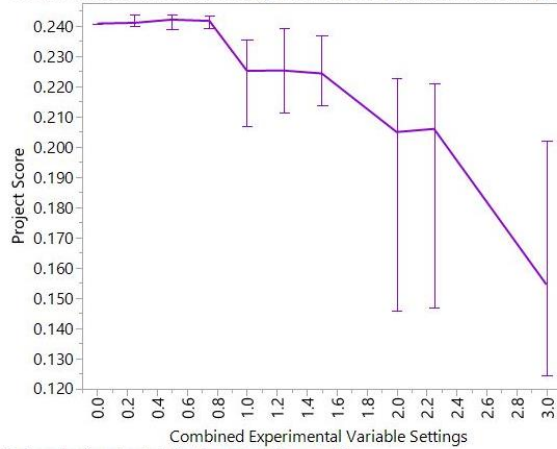
The Simple Control results display a similar pattern to the No Control results as the experimental variable settings vary. The Simple Control method achieves the best outcomes at a moderate level of control latency. This indicates that the method is likely prone to “over-controlling” by acting too quickly and too frequently on signals in project performance which do not warrant control activity. This is visible in Figure 32.

Table 19: Simple Control Confusion Matrix

Confusion Matrix		Actual	
		0	1
Predicted	0	720	0
	1	7380	0
Prediction Metrics			
Accuracy		8.9%	
Sensitivity (TPR)		N/A	
Specificity (TNR)		8.9%	
False Negative Rate		N/A	
False Positive Rate		91.1%	

The Simple Control prediction results show a strong propensity to signal cancellation in most circumstances which would have resulted in the loss of 7,380 products which met threshold delivery criteria.

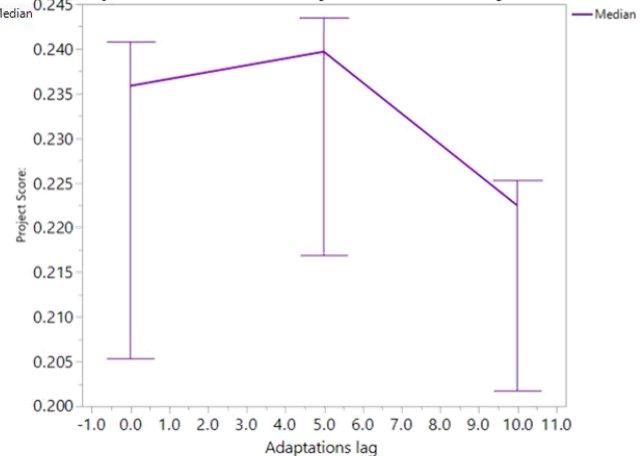
Simple Control: Median Project Score vs Combined Settings



Each error bar is constructed using the upper and lower quartiles.

Figure 31: Simple Control Plot - Median Project Score vs Combined Settings

Simple Control: Median Project Score vs Latency



Each error bar is constructed using the upper and lower quartiles.

Figure 32: Simple Control Plot - Median Project Score vs Latency Settings

### 5.3 Earned Value Management with Earned Schedule

The results from the Earned Value Management with Earned Schedule (EVM-ES) simulations display a greater variance than the previous two methods. The results which do not meet scope targets also tend to be more clearly segregated from those that do. This likely reflects the control activity taking place within the project to address variances.

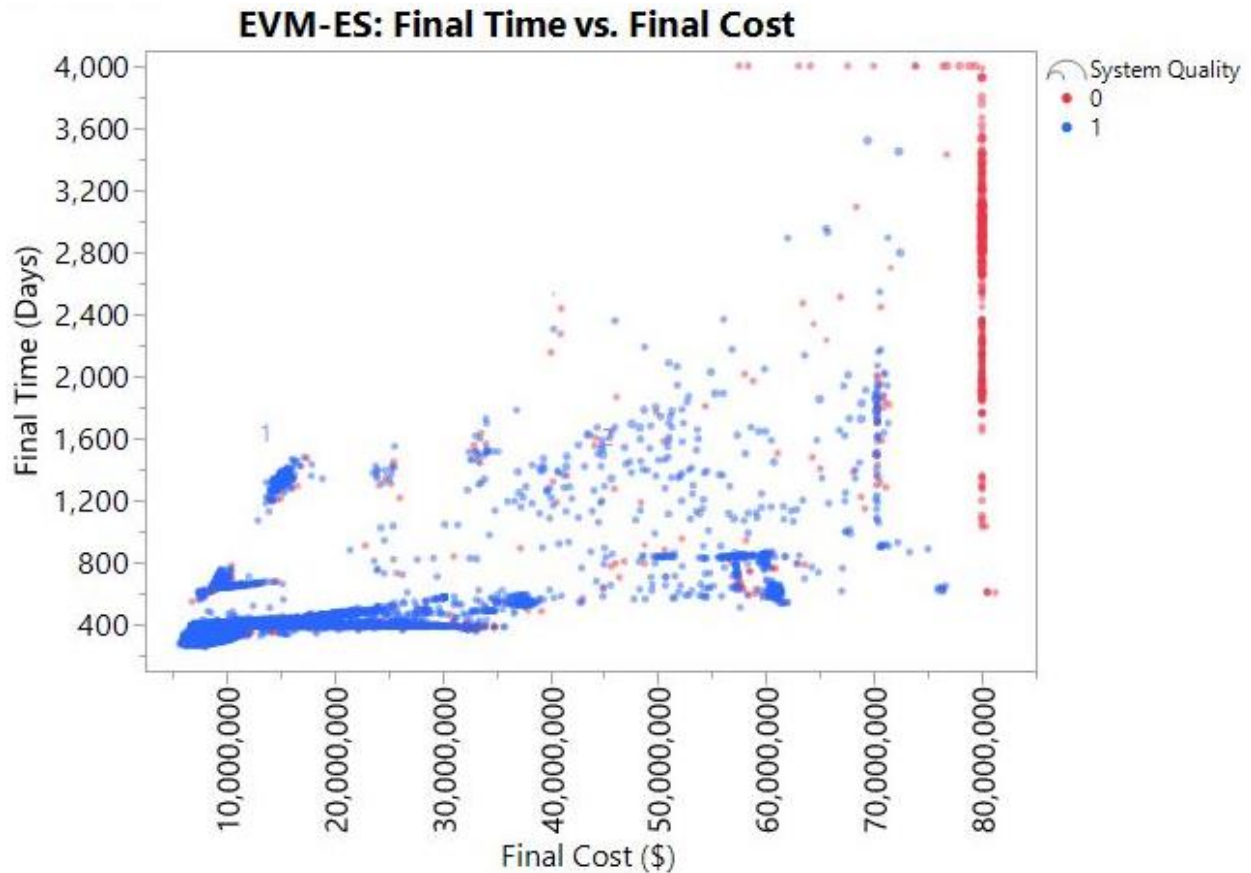


Figure 33: EVM-ES Control Scatter Plot - Scope Targets Met by Cost and Time

The Earned Value results also show an improvement as latency in the model increases. One possible explanation may be that the method is prone to “over-controlling” by acting too quickly and too frequently on signals in project performance which do not warrant control activity.

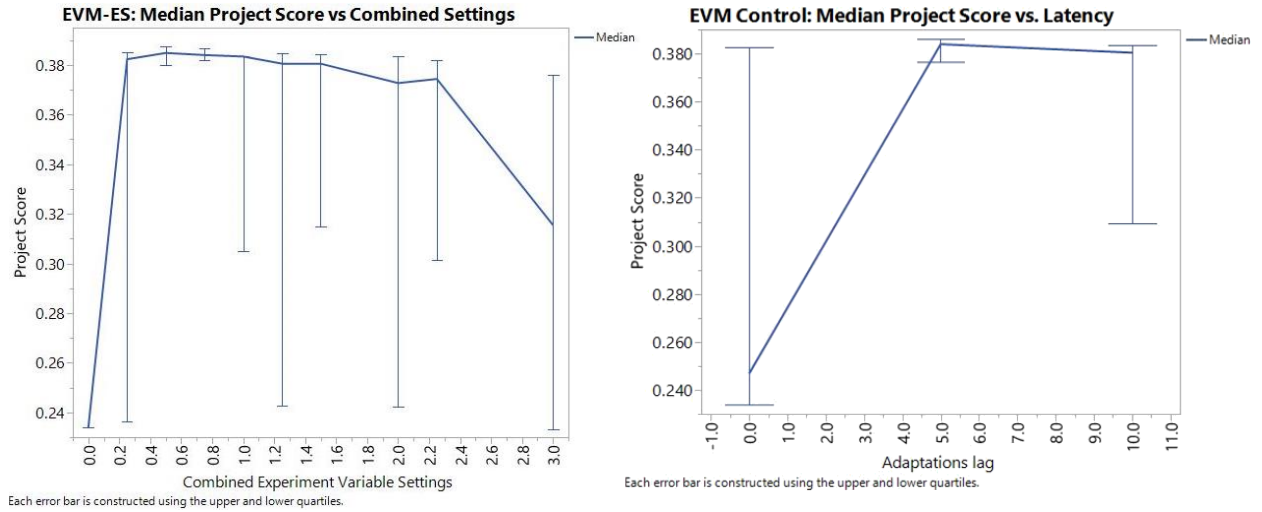


Figure 34: EVM-ES Control Plot - Median Project Score vs. Combined Settings

Figure 35: EVM-ES Control Plot - Median Project Score vs. Latency Settings

Table 20: EVM-ES Confusion Matrix

Confusion Matrix		Actual	
		0	1
Predicted	0	8072	12
	1	5	11
Prediction Metrics			
Accuracy		99.8%	
Sensitivity (TPR)		47.8%	
Specificity (TNR)		99.9%	
False Negative Rate		52.2%	
False Positive Rate		0.1%	

## 5.4 Multiple Risk Levels

The Multiple Risk Level (MRL) results have the greatest variance among all of the control methods. Results with unmet scope targets also tend to be more clearly segregated as the projects which do not meet the scope targets tend to reach the project cancellation cost or schedule thresholds.

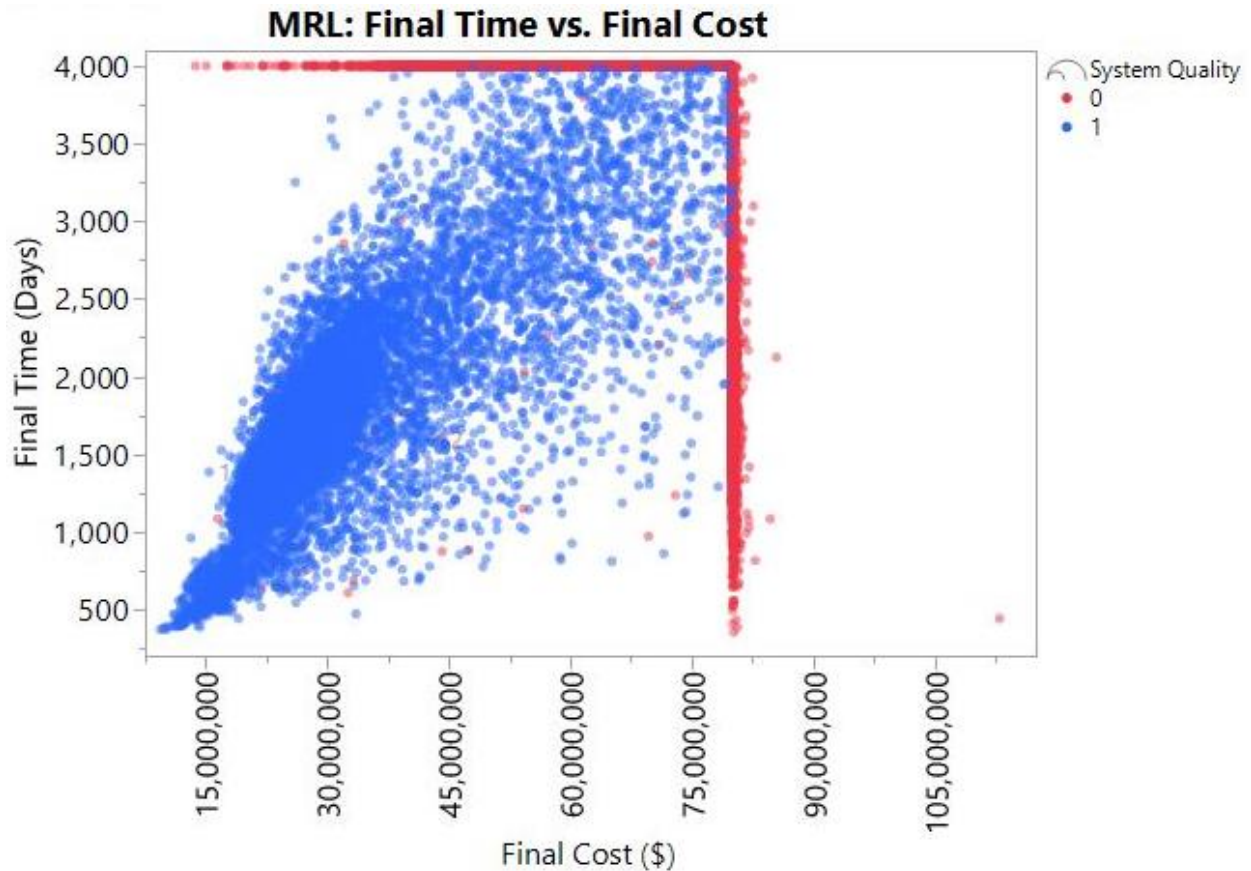


Figure 36: MRL Control Scatter Plot - Scope Targets Met by Cost and Time

The MRL results do not show any significant variation between latency settings. Figure 37 shows the accuracy settings have the greatest impact on project outcomes, while the uncertainty settings have only a minor effect. Figure 38 shows a significant increase in variance and decrease in project outcomes for the MRL results as the combined experimental settings increase.

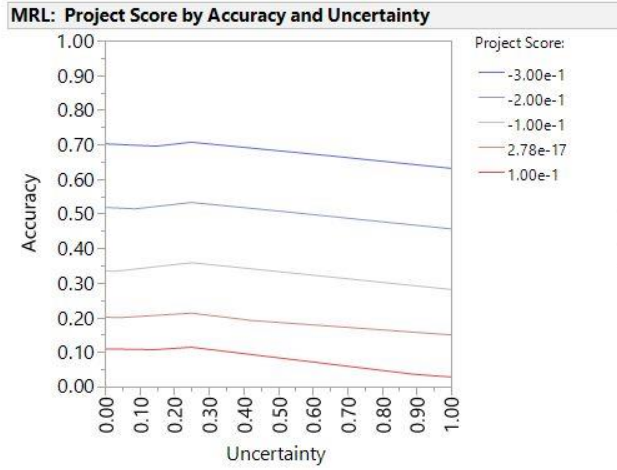


Figure 37: MRL Control Contour Plot – Project Score by Accuracy and Uncertainty

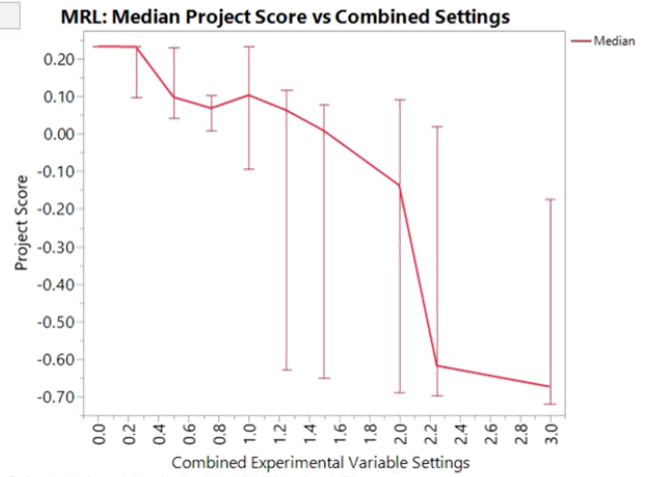


Figure 38: MRL Control Plot - Median Project Score vs. Combined Settings

Table 21 displays the forecast trial results for the MRL method.

Table 21: MRL Confusion Matrix

Confusion Matrix		Actual	
		0	1
Predicted	0	5318	280
	1	1622	880
Prediction Metrics			
Accuracy		76.5%	
Sensitivity (TPR)		75.9%	
Specificity (TNR)		76.6%	
False Negative Rate		24.1%	
False Positive Rate		23.4%	

## 5.5 Control Method Result Comparisons

Table 22: Summary of Outcome Trial Result Statistics by Method

Outcome Results	None	Simple	EVM	MRL
Project Score Mean	0.334	0.212	0.315	-0.117
Project Score Median	0.360	0.225	0.380	0.065
Project Score St Dev	0.054	0.045	0.139	0.378
% Met Targets	0.927	0.850	0.889	0.703
% Delivered	0.970	0.912	0.968	0.860

Table 22 shows the summary statistics for all project outcome simulations for each control method. Table 23 shows the summary statistics for all project forecast simulations for each control method.

Table 23: Summary of Forecast Trial Result Statistics by Method

Forecast Results	None	Simple	EVM	MRL
Accuracy	100.0%	8.9%	99.8%	76.5%
Sensitivity (TPR)	N/A	N/A	47.8%	75.9%
Specificity (TNR)	100.0%	8.9%	99.9%	76.6%
False Negative Rate	N/A	N/A	52.2%	24.1%
False Positive Rate	0.0%	91.1%	0.1%	23.4%
Average Cost Savings	N/A	N/A	\$14,053,991	\$34,620,314
Average Time Savings	N/A	N/A	355	1,398

The No Control method surprisingly produced some of the best outcomes, while the Multiple Risk Level method surprisingly produced some of the worst outcomes. The No Control method appears to deliver powerful prediction results, but this is an illusion due to the fact that no trials warranted cancellation. The No Control method is incapable of forecasting the need to cancel a project. The MRL method, while less accurate overall, performs the best at predicting the need to cancel a project, which is indicated by the sensitivity metric within the forecast results. It also performs best at predicting the need to cancel early.



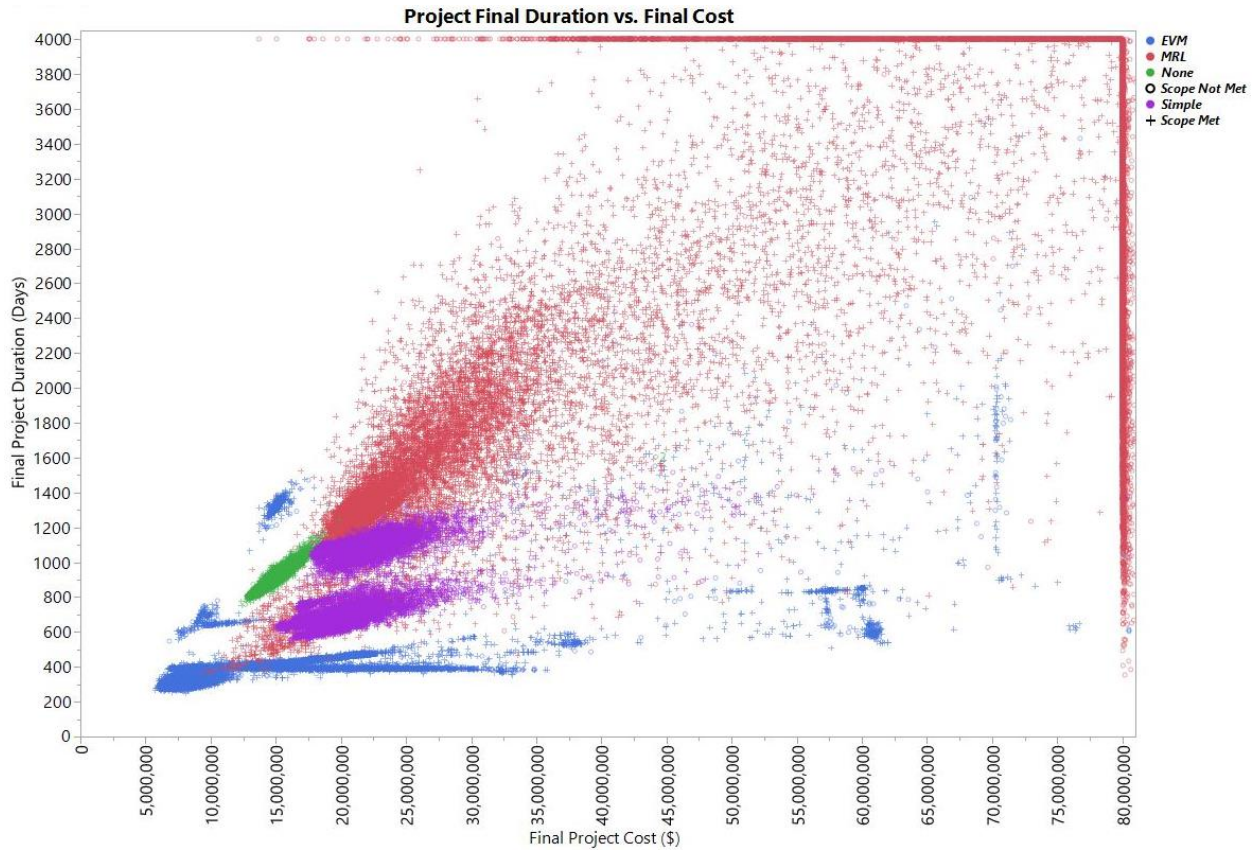


Figure 39: Combined Scatter Plot - Scope Targets by Cost and Duration

The results displayed in Table 22 and Figure 39 reveal the comparative performance of the four project control methods in the experiment. The results for the EVM method achieve the lowest combined cost and schedule outcomes while meeting the project scope targets as well as the best combined cost and schedule outcomes for those trials not meeting scope targets. The No Control method results are tightly distributed with a low variance. The MRL method produced the worst average and median outcomes for the experiment and met scope targets at the lowest rate.

Figure 40 highlights the distribution of outcomes for each control method. The MRL results have the highest variance, followed by the EVM results.

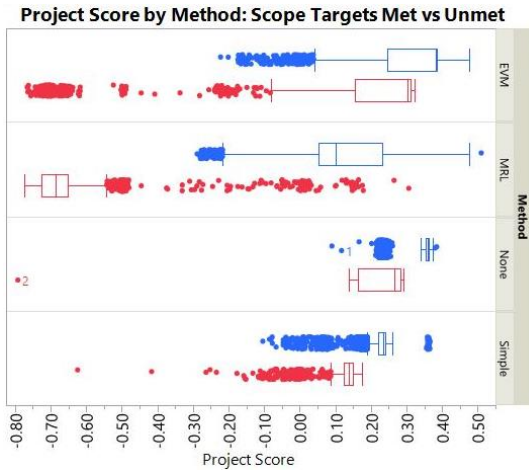


Figure 40: Box Plot - Project Score by Method

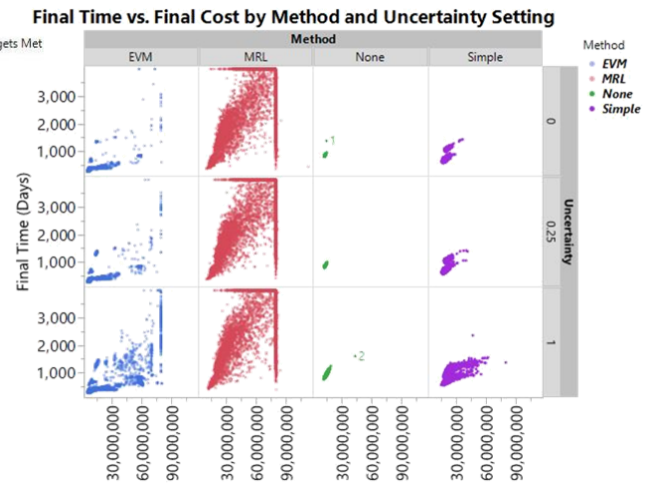


Figure 41: Scatter Plot: Time vs. Cost by Method and Uncertainty

Figure 41 shows the results for each method by uncertainty setting. Figure 42 shows the effect of the experimental variable settings on project outcomes. The first unmet targets do not appear until the combined setting value is 1.0. At the highest combined setting, nearly 50% of project simulations do not meet the scope targets. Figure 43 shows the overall rate of scope targets met for each method. The MRL method produces the highest rate of unmet scope targets, while the No Control method produces the lowest.

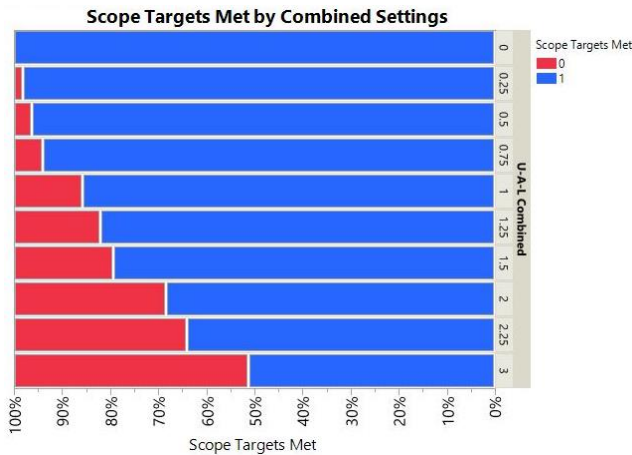


Figure 42: Bar Plot - Scope Targets Met by Combined Settings

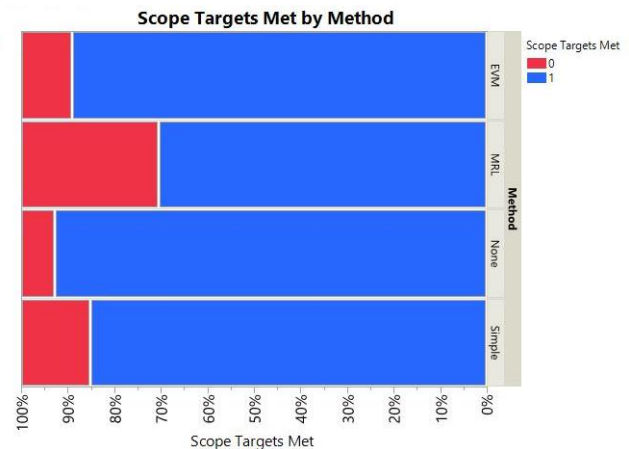


Figure 43: Bar Plot - Scope Targets Met by Method

Table 24: Number of Project Trials Resulting in Unmet Scope Targets by Cancellation Outcome

Unmet Scope Target Results	None	Simple	EVM	MRL
Cancelled		1	294	7927
Not Cancelled	660	4052	2694	91

Table 24 lists the number of project trials which did not achieve the scope targets for each control method. Those trials that closed because the cost or duration cancellation thresholds were exceeded are listed in the “Cancelled” row. The projects in the “Not Cancelled” row for the “No Control” method are those which reached the planned measured values but did not achieve the actual scope target values because of risk, uncertainty, or inaccurate measurements. Those projects in the “Not Cancelled” row for the remaining three methods closed as a result of uncertainty or inaccurate measurements.

## 6 ANALYSIS AND DISCUSSION

The experiment yields interesting and surprising results. This section will analyze the results of the experiment to uncover valuable insights. This section will also review the hypotheses formed prior to the experiment and examine whether each is supported by the results.

### 6.1 Evaluation of Methods: Basis of Value

Table 25: Control Method Ranking by Evaluation Criteria

Criteria	Metric	None	Simple	EVM-ES	MRL
1) The method integrates cost, schedule, and scope information	Control decisions based on integrated data?	No	Limited	Yes	Yes
2a) The method provides <b>accurate</b> forecasting information to enable proactive decisions	Accuracy	1st <sup>1</sup>	4th	2nd	3rd
2b) The method provides accurate forecasting information to <b>enable proactive decisions</b>	Sensitivity	N/A <sup>2</sup>	N/A <sup>2</sup>	2nd	1st
2b) The method provides accurate forecasting information to <b>enable proactive decisions</b>	Mean Cost Savings	N/A <sup>2</sup>	N/A <sup>2</sup>	2nd	1st
2b) The method provides accurate forecasting information to <b>enable proactive decisions</b>	Mean Duration Savings	N/A <sup>2</sup>	N/A <sup>2</sup>	2nd	1st
3) The method enables effective risk mitigation	Median Project Score	N/A <sup>3</sup>	2nd	1st	3rd
3) The method enables effective risk mitigation	Percentage Met Scope Targets	N/A <sup>3</sup>	2nd	1st	3rd

<sup>1</sup> The No Control method always predicts the same outcome

<sup>2</sup> The No Control and Simple Control simulations did not contain any trials which exceeded the cost or duration cancellation thresholds.

<sup>3</sup> The No Control method does not satisfy the criteria for risk mitigation because it does not implement control responses to risk events.

Table 25 shows the relative ranking of each control method. The EVM-ES and MRL methods combine system-level cost, schedule, and work-scope information to satisfy the first criteria. The Simple control method combines task-level cost, schedule, and product-scope information. The experiment did not evaluate a control method which considered system-level product scope, and this is a recommended area for future work.

The No Control method resulted in perfect accuracy because the No Control method did not contain any trials which exceeded the cost or schedule duration thresholds. The No Control method will never predict that a project should be cancelled, however, and so this result is a coincidence which is not indicative of the method's true predictive power. The EVM-ES method

resulted in near-perfect overall prediction accuracy based on actual project measurements and forecasts. The EVM-ES method is the best method in the experiment for prediction if it is equally important to continue projects that do not warrant cancellation as it is to cancel projects that do. The MRL method best enabled proactive decision making because it produced the highest sensitivity, mean cost saving, and mean duration saving values. MRL predictions were the most sensitive to predicting when a project warranted cancellation and forecasted the need to cancel the project earlier than all other methods.

The EVM-ES method produced the best median project score and met the scope target values at the highest rate; the EVM-ES method therefore best enabled effective risk mitigation. The project score is a normalized, priority-weighted value centered on zero produced by the integrated project model used in the experiment which combines the cost, duration, and scope outcomes for the simulated project. A project score of zero implies a priority-weighted project outcome that is equivalent to the project meeting its cost, duration, and scope targets. A project score of 1.0 implies a priority-weighted project outcome that is equivalent to the project exceeding its cost, schedule, and scope goals by 100%. A detailed explanation of the project score is given in Section 5.

## 6.2 Effect of Experimental Variables

The results in the Section 5 show that the three experimental variables of uncertainty, measurement accuracy, and control latency have a significant effect on project and project control outcomes. The effects varied depending on the project control method in use. This section will discuss the most surprising and interesting interactions witnessed in the experimental results.

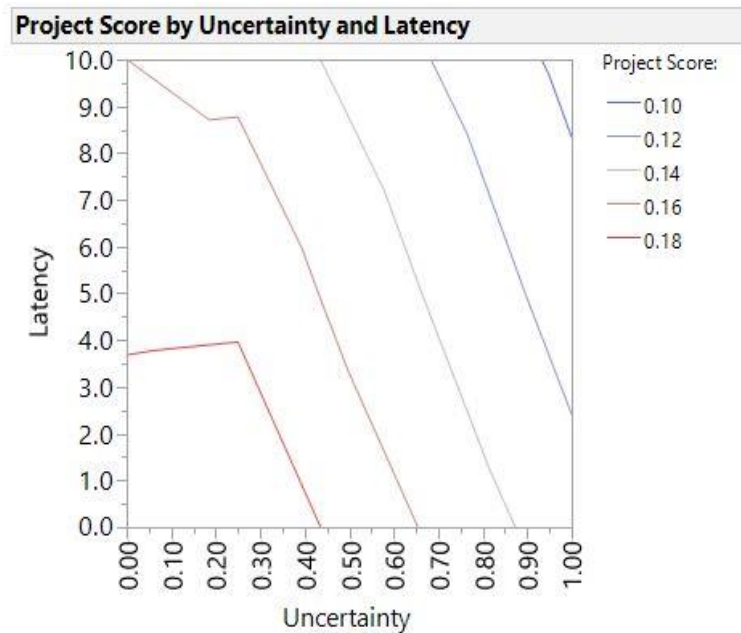


Figure 44: Contour Plot –Project Score by Uncertainty and Latency (All Methods)

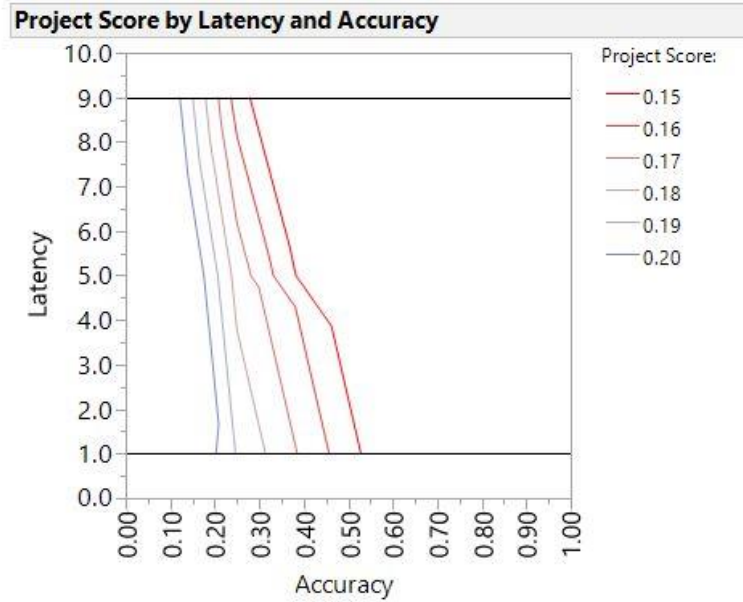


Figure 45: Contour Plot -- Project Score by Accuracy and Latency (All Methods)

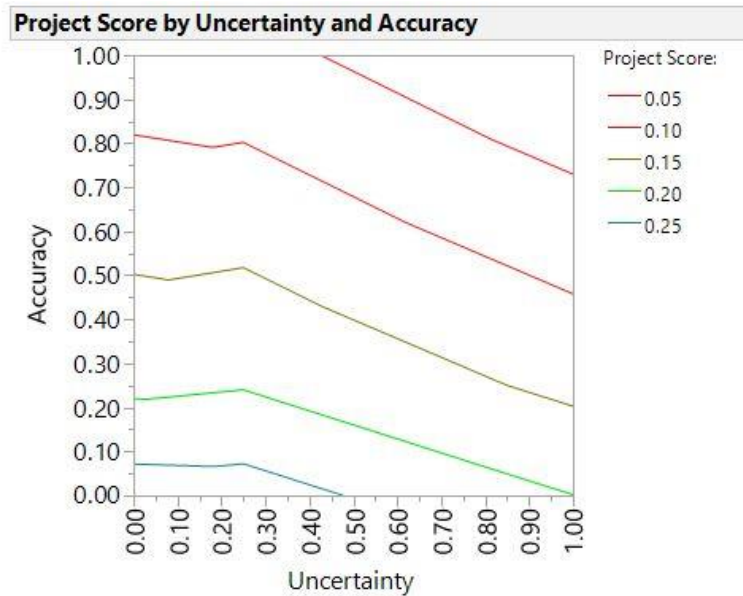


Figure 46: Contour Plot - Project Score by Accuracy and Latency (All Methods)

The contour plots depicted in Figure 44, Figure 45, and Figure 46 show the relative effect of the experimental variables on the overall experimental results. Each contour plot shows the relative effect of a pair of experimental variables. Accuracy had the greatest impact on the results relative to uncertainty and latency. Uncertainty had a lesser impact on the experimental results relative to accuracy and a greater impact on the experimental results relative to latency. Latency had a weaker impact on the experimental results relative to both accuracy and uncertainty.



The EVM-ES and Simple Control methods showed a benefit from higher settings of the latency variable, which is surprising because higher latency means that decisions are being made on older information. This effect can be seen in Figure 32 and Figure 35 in Sections 5.2 and 5.3. It is possible that the EVM-ES and Simple Control methods are prone to “over-controlling,” and that higher latency reduces this phenomenon. Exploring why higher latency benefits the EVM-ES and Simple Control methods is an area for future work.

The results from the subset of data where the uncertainty setting is “None” and the accuracy setting is “Low” produce the metrics in Table 26 and the box plot in Figure 47. No Control and EVM-ES produce the best experimental results under conditions of no uncertainty and low accuracy. The EVM-ES method produces a higher median project score under conditions of no

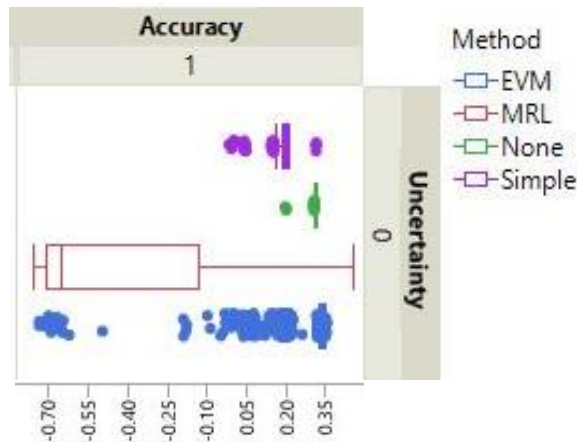


Figure 47: Box Plot - Project Score by Method, No Uncertainty, Low Accuracy.

uncertainty and low accuracy but also has a higher likelihood of producing negative outcomes as shown by the large number of outliers to the left of the distribution in Figure 47. The No Control results have a lower variance and a higher mean project score under conditions of no uncertainty and low accuracy. No Control is not clearly superior to EVM-ES under conditions of no uncertainty and low accuracy because the No Control method results in a lower median project score than EVM-ES under conditions of no uncertainty and low accuracy. The No Control method results could be preferred to the EVM-ES method results under conditions of no uncertainty and low accuracy if the project stakeholders are highly risk averse.

Table 26: Statistics for no uncertainty, low accuracy trials

Unc: 0, Acc: 1	No Control	Simple	EVM-ES	MRL
Project Score Mean	<b>0.360</b>	0.231	0.350	-0.462
Project Score Median	0.361	0.236	<b>0.387</b>	-0.656

*Hypothesis 1: The “No Control” method will produce superior results when compared to the remaining three methods under conditions of no uncertainty combined with low measurement accuracy. FALSE*

## **6.3 Method Comparisons**

### **6.3.1 No Control**

The No Control method produced surprisingly favorable project outcome results when compared with the other methods. The No Control method produced the highest mean project score and percentage of projects which met the scope targets with the second-lowest variability in project scores. This is surprising because the method does not measure the actual project status or respond to variances by implementing control measures. The No Control method limited both upside and downside performance by executing to a set plan regardless of the actual project developments. That the No Control method did not have as large a range of negative outcomes as other methods suggests that negative project outcomes are partly attributable to project control decisions. It also implies that control measures may be best applied with discretion.

The positive results for No Control are given due to an assumption that the baseline planning function within the model is highly efficient. The planning function within the model is very well calibrated to the project which is simulated. The resulting plan is well suited to achieving the project goals if executed without any control or feedback. For future research, this insight suggests that it would be a valuable exercise to add a variable to control the level of planning effectiveness which could represent the planning capability of the organization. Running the experiment with this included would give insight into which methods are robust or dependent on accurate planning.

Changes in the level of measurement accuracy did not significantly impact the No Control results because the No Control method only relies on measurements to determine when scope items are complete. The results declined modestly as uncertainty increased.

While the No Control method generated the appearance of ideal forecast results, this is coincidental due to no trials exceeding the cost of duration cancellation thresholds. The No Control method does not produce meaningful forecasts to predict project outcomes because it always assumes that the project will conform to the plan. To address the insufficient handling of the No Control method in the experiment, the planning mechanisms in the project model should be revised in future work.

### **6.3.2 Simple Control**

The Simple Control method produced a moderate set of project outcome results which underperformed the EVM-ES and No Control methods but generally outperformed the MRL results. It suffered significant declines in performance as the experimental variables increased. The results showed that this method as well as the EVM-ES method benefitted from moderate levels of control latency which likely reduced over-controlling behavior.



Simple Control produced extremely inaccurate predictions in the forecast results which would be of very little benefit to a project manager. It was the worst performing method for forecasting whether a project warrants cancellation.

While the Simple Control forecast results were inferior to the those for the EVM-ES and MRL methods, the Simple Control project outcome results were superior to the MRL results.

*Hypothesis 2: The “Simple Control” method will produce inferior results when compared to the “Earned Value” and “Multiple Risk Level” methods under all conditions. FALSE*

### **6.3.3 Earned Value Management with Earned Schedule**

The Earned Value Management with Earned Schedule (EVM-ES) control method produced results with a moderately high degree of variation. The results from the EVM-ES method were both the best and some of the worst in the range of experimental outcomes. The EVM-ES method demonstrated the ability to produce superior outcomes as a result of measuring variances and implementing control actions at both local and global levels in the project. Unfortunately, EVM-ES also demonstrated the ability to exacerbate problematic conditions through this behavior as well. The predictive value of the forecasts created from the EVM-ES method were the most accurate in the experiment. EVM-ES demonstrated only a moderate capability to predict the need to cancel projects early, however.

### 6.3.4 Multiple Risk Levels

The Multiple Risk Level control method produced generally poor results throughout the experiment. The results had a high degree of variability, likely due to the large differences in pre-planned options that were implemented as project variances changed over time. The MRL method ranked third in overall forecasting accuracy but ranked highest in sensitivity. The MRL method was good at predicting when a project should be cancelled but mediocre at predicting when a project should not be cancelled. The MRL method was least affected by uncertainty and most affected by measurement accuracy.

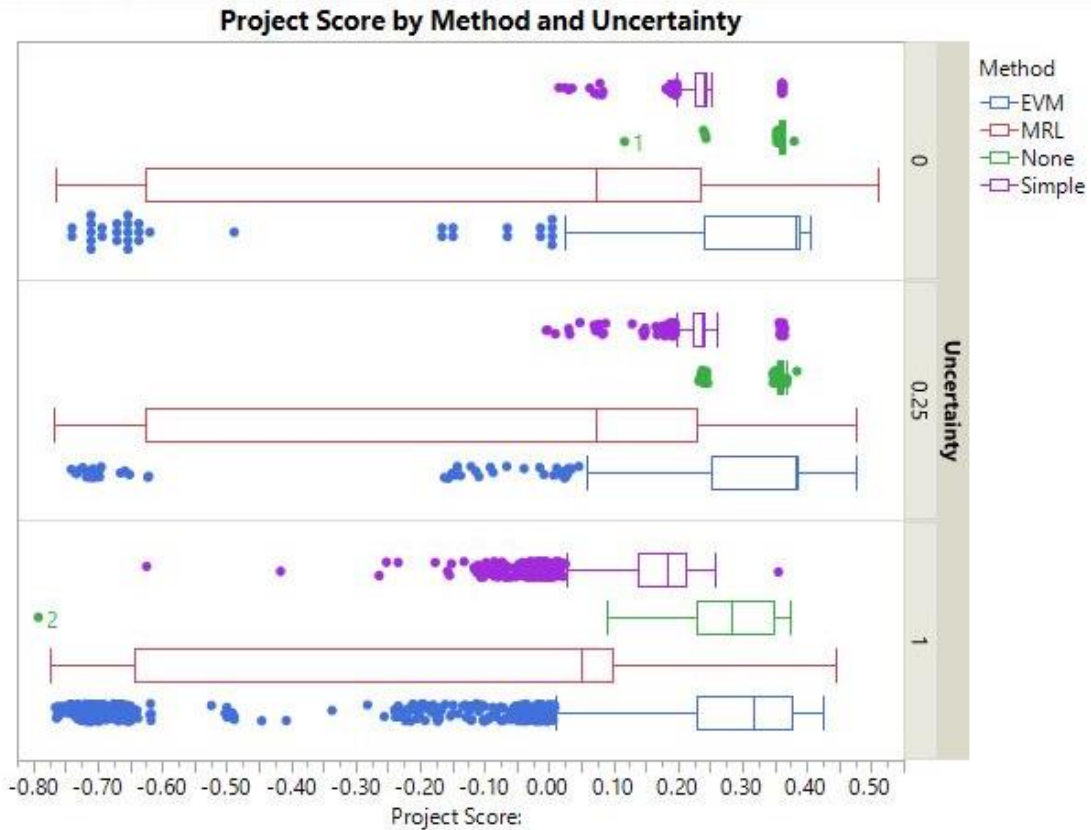


Figure 48: Box Plot - Project Score by Method and Uncertainty

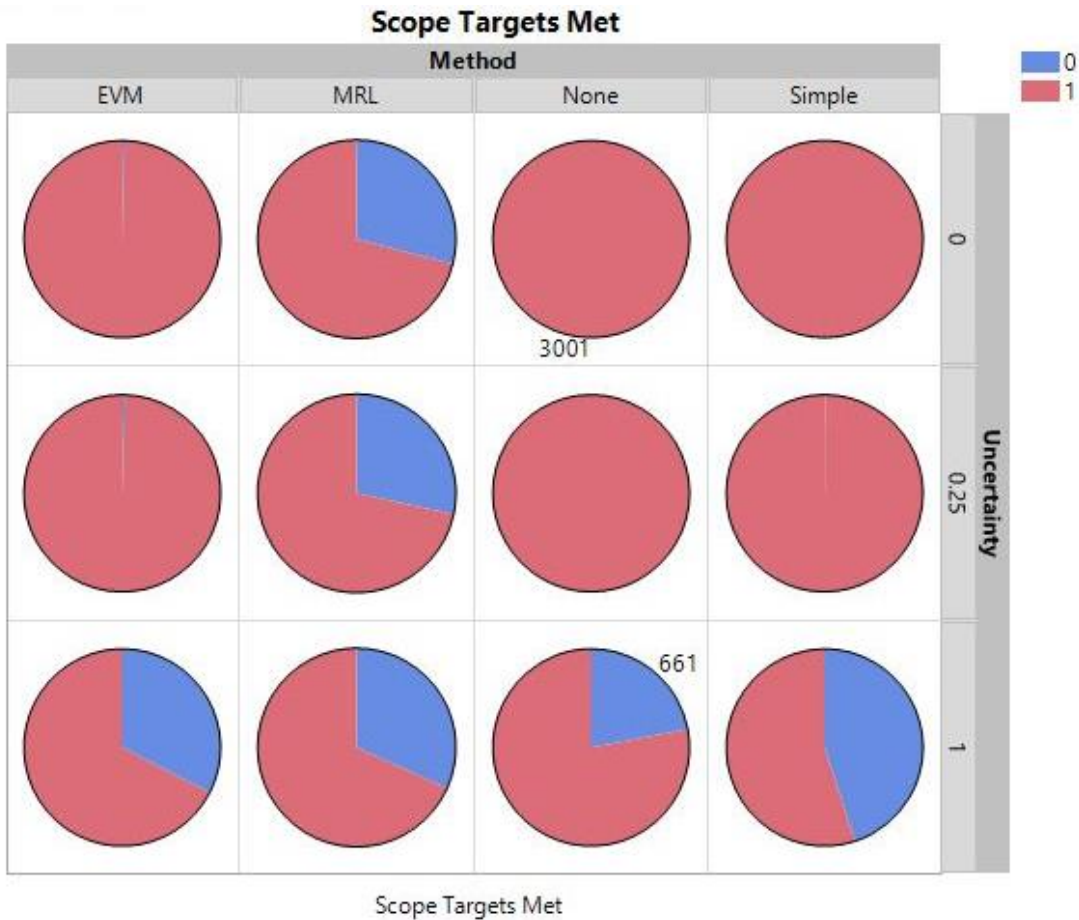


Figure 49: Pie Plots - Scope Targets Met by Method and Uncertainty

Figure 49 and Figure 48 show results for the four methods under varying levels of uncertainty. The MRL method underperforms the other three methods in terms of project score at all levels of uncertainty. The MRL method performs better than the Simple Control method, comparable to the EVM-ES method, and worse than the No Control method in terms of meeting scope targets under conditions of high uncertainty in the experiment. It is worth noting that the MRL method is robust to changes in the level of uncertainty in the experiment.

*Hypothesis 3: The “Multiple Risk Level” method will produce superior results when compared to the remaining three methods under conditions of high uncertainty. FALSE*

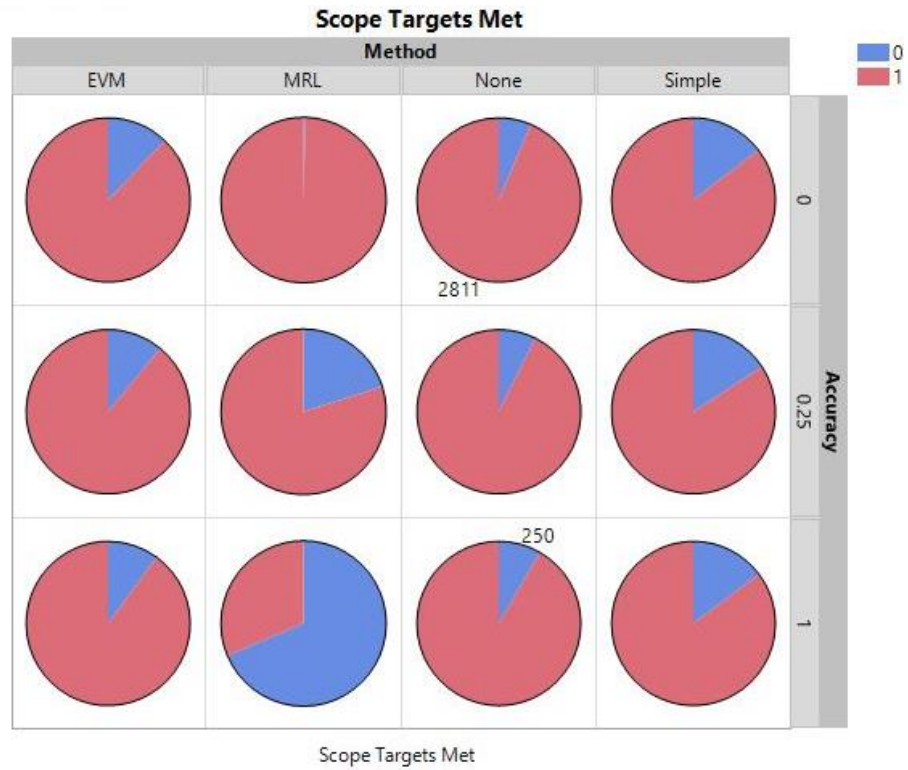


Figure 50: Pie Plots - Scope Targets Met by Method and Accuracy

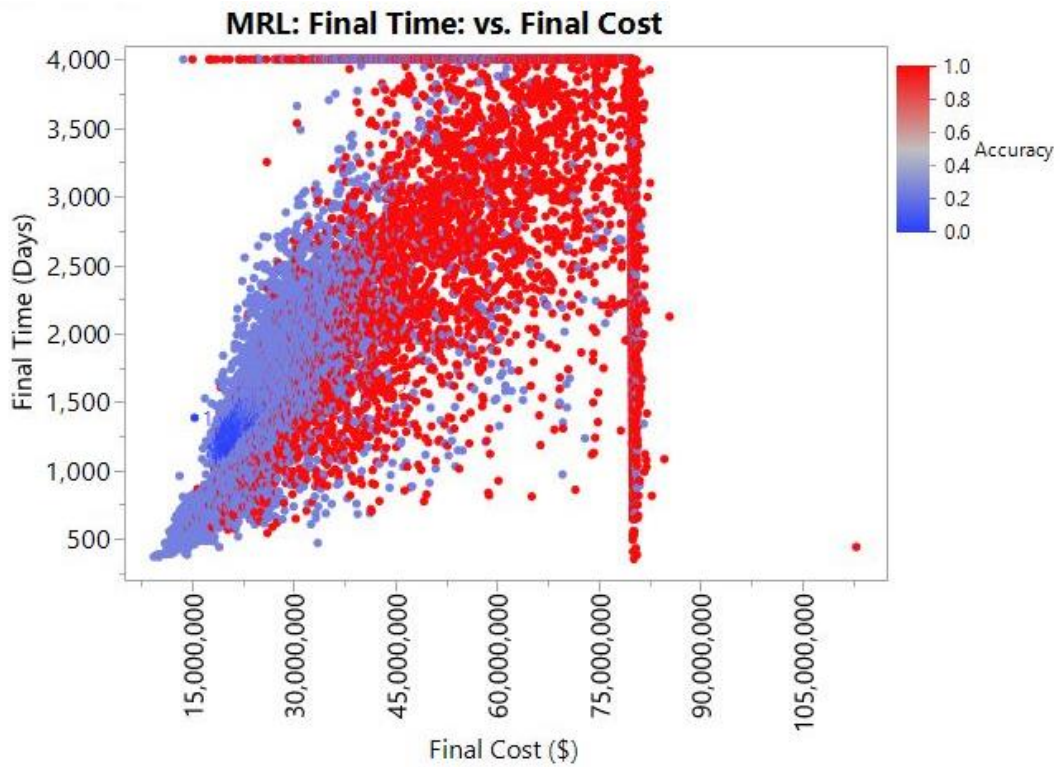


Figure 51: MRL Scatter Plot - Cost versus Time by Accuracy

Figure 50 shows results for the four methods under varying levels of accuracy. Figure 51 illustrates the effect of accuracy on the project cost and time outcomes for the MRL method. These charts reveals that the MRL method is greatly affected by changes in the level of measurement accuracy. The MRL method outperforms the other methods in terms of meeting scope targets under ideal conditions of measurement accuracy but underperforms in terms of cost and time outcomes in the experiment.

Beyond the outsized impact of the measurement accuracy variable, it is not clear why the MRL method underperformed the other methods in the experiment. One possible explanation is that the MRL method did not implement control activity as aggressively as the EVM-ES and Simple control methods in the project model. Control activity in the model is implemented when negative variances appear. Since the MRL method dynamically implements different plans as risks are realized, negative variances do not appear as often. The MRL method would therefore not implement control activity as often within the project model.

The less aggressive control activity in the project model would not explain the MRL underperformance when compared to the No Control method in the experiment. This likely resulted from the additional margin included in the multiple risk level plans which is meant to accommodate exceptional activity in the project model. Excess margin would have a negative effect on cost and time outcomes and a positive effect on scope outcomes in the experiment. Since the priorities for cost, time, and scope in the experiment were equal, there would be a negative net effect on the MRL project scores. It is possible that the margin included for the different risk levels is inappropriate or counterproductive given the conditions of the experiment.

## 7 CONCLUSION AND RECOMMENDATIONS

### 7.1 Conclusion

The research question motivating the work in this thesis is:

*Does a model-based control method achieve better project control outcomes than the earned value management method?*

The basis of value for a Department of Defense project control method are:

- The method integrates cost, schedule, and scope information
- The method provides accurate forecasting information to enable proactive decisions
- The method enables effective risk mitigation

The Multiple Risk Level (MRL) model is the model-based control method selected for comparison with the Earned Value Management with Earned Schedule (EVM-ES) method. Both the EVM-ES and MRL control methods integrate cost, schedule, and work scope information to a comparable degree. The results of the experiments show that the EVM-ES method provides more accurate forecasting information, while the MRL method enables more proactive decisions. The EVM-ES method enables more effective risk mitigation within the conditions of the experiment.

*Table 27: Comparison of EVM-ES and MRL results*

Criteria	EVM-ES	MRL
1) The method integrates cost, schedule, and scope information	✓	✓
2a) The method provides <b>accurate forecasting information</b> to enable proactive decisions	✓	
2b) The method provides accurate forecasting information to <b>enable proactive decisions</b>		✓
3) The method enables effective risk mitigation	✓	

**The results of the experiment do not support affirming the hypotheses.** The model-based control method as implemented in the model does not achieve better project control outcomes than the earned value method. While it does provide more proactive predictions than the earned value method, it does not predict as accurately or mitigate risk as well.

The surprising results for the MRL method may be due to the way that control activity is implemented in the model based on variances. They may also be partly explained by the implementation of the risk contingencies in the different levels of the MRL planning model. It would be interesting to conduct the experiment using alternate contingencies and a different method for applying control. This is a topic for future work listed in Section 8.2.

## **7.2 Recommendation 1: Explore Innovative Project Control Methods**

This thesis demonstrated the power of using an integrated project model to assess project control methods. Three of the four project control methods studied displayed at least one project control characteristic which was superior to the others. By isolating the properties of the method that lead to the superior outcome and experimenting with ways to combine or extend methods to capitalize on such strengths, it may be possible to project control methods that are clearly dominant.

Section 2.8 lists several emerging project control concepts with the promise to improve upon or replace traditional methods. Only one method was tested as part of this thesis. Integrated project modeling allows for testing and comparing several methods at relatively low expense. Additional effort should be applied to evaluating an exhaustive range of project control methods. The findings from this effort should be used in the same manner suggested in the previous paragraph to develop innovative project control methods.

## **7.3 Recommendation 2: Eliminate Prescriptive EVMS Policy**

The experiment conducted as part of this thesis demonstrated that the Earned Value Management with Earned Schedule (EVM-ES) control method can produce both excellent and poor project outcomes. Similarly, alternative methods perform better than EVM-ES under certain experimental conditions and stakeholder priorities. The policy of the Department of Defense is to require the use of an EVMS for large, complex engineering projects unless a waiver is justified and obtained. The reality facing early project teams typically consists of more work than workers, and so these teams may not have the capacity to pursue such a waiver. People respond to incentives, and the incentive in the situation is to follow the policy as written because of the effort required to waive it.

Instead, the policy should be scaled back to ensure the basic project management principles and best practices which are enshrined in the EVMS guidelines can be included in a contract without the prescription of EVM as a project control technique. The policy should allow for flexibility as the default position so that innovative project control techniques can be developed and matured using integrated project models on Defense programs. This is not to say that EVM should not be used. Indeed, if the nature of the project is such that EVM is a superior option, it should be implemented. The choice should be available at the beginning of the project for the project management team to make along with so many early choices about the structure of a project.

## **7.4 Recommendation 3: Implement Integrated Project Models on DoD Programs**

This thesis demonstrated the value of employing multi-domain integrated product models for evaluating project control methods. This value extends across the range of project management and systems engineering management disciplines. The Department of Defense should formally implement integrated project models to realize this value.

By creating an integrated project model, the project team must engage in a rigorous planning and validation process which exceeds what is necessary to create an integrated master schedule and earned value management system. A digital integrated project model can be used to explore the

project trade space during initiation. It can be used to explore the emergent outcomes of possible decisions that must be committed to early. Such a model also encourages cross-discipline communication and collaboration, especially if it is linked to a digital systems engineering model.

During project execution, the model facilitates a disciplined data strategy. Project data can be monitored, collected, groomed, and managed in a disciplined, consistent manner. Quality data enables the effective use of analytics and machine learning models which can inform and improve project decisions. Machine learning can significantly improve project planning by enhancing cost and schedule estimation, work planning, and risk planning. Machine learning models can also be employed to generate more accurate forecasts. Accurate predictions will give project managers the confidence to take action or refrain from action and realize more desirable project outcomes. The experiment showed that the best outcomes result from a combination implementing control actions that effect desirable change and refraining from control actions that effect undesirable change.

Analytics can also be used to maintain a project solution space from which to generate alternative project plans. The experiment showed that the use of multiple project plans insulated against the negative effects of uncertainty. If this benefit can be isolated from the undesirable project behavior seen in the experiment for the Multiple Risk Level method, it could be used to mitigate the risk posed by uncertain project environments.

An integrated project model can also be used to evaluate project management and systems engineering management decisions via simulation. The user can model different scenarios to understand what project outcomes may flow from the decision. The model can also be used to calculate the likelihood that a project action may be helpful or harmful.

The project management team can use the model to evaluate a multitude of factors taken for granted in a project. The team can automate workflows to measure the effect of changes in factors like tools and equipment, team composition, incentives, quality control activities, and much more. The team can then analyze the results to find the factors to which the project is most sensitive and which can yield the greatest benefits.

The ability to model and understand cross-domain effects is a major benefit of an integrated project model. Project managers are often left to guess what impacts a funding or scheduling issue may have on the product. A model eliminates the need to guess and provides the basis for an informed decision. As a model is used as part of a project, it becomes more accurate and useful because real-world project experiences are used to refine and calibrate the model.

## **7.5 Recommendation 4: Create a Project Control Selection Tool**

Defense Acquisition University should publish a software tool to assist project managers in selecting an appropriate project control approach for their program. This will allow project managers to take full advantage of the flexibility to tailor a control strategy which has the best chance of success. The software should employ prescriptive algorithms fed by data from past projects and project control method modeling. The software would recommend suitable control principles, tools, and methods based on the characteristics of the new project.



## **7.6 Recommendation 5: Publish a Digital Project Management Strategy**

Following the implementation of integrated project models within select programs to demonstrate value, the Department of Defense should publish a Digital Program Management Strategy advocating for their widespread use. This document would communicate a vision for the use of digital integrated project management tools and techniques and provide a framework for organizations within the Department to take the necessary steps to realize that vision. This policy would complement the existing Digital Engineering Strategy published in 2018.

The widespread use of integrated project models is likely to increase the chances of project success throughout Defense acquisition. Successful Defense projects yield improvements to mission capabilities that satisfy user needs at an affordable cost on a relevant timeline. In this way, integrated project models and digital project management techniques support “the development of a more lethal force which yields a decisive and sustained U.S. military advantage and supports the National Defense Strategy.” [69]

## **8 FUTURE WORK**

Many topics within this thesis warrant further exploration, revision, calibration, or validation. This section will list some of the topics most in need of future work but will surely overlook several more.

The Department of Defense acquisition community typically speaks in terms of programs rather than projects. Programs are a set of “related projects, subsidiary programs, and program activities managed in a coordinated manner to obtain benefits not available from managing them individually.” [13] In systems thinking terms, programs are sets of related projects managed together in a way that produces emergent benefits. This thesis focuses on the application of project control methods at the project level, considering the project as a system. It would be valuable to conduct an analysis of the control of programs similarly modeled as integrated systems as a topic for future work.

### **8.1 The Value of Integrated Models**

Research into real-world case studies of systems developed using integrated project models and systems developed using separate product, process, and organizational models would provide data to support or refute the value of integrated project models. It would be valuable to design and conduct an experiment which tested the use of integrated and separate domain models in a controlled manner to identify the effect of each approach on product and project outcomes. Such an experiment could be used to explore the factors influencing the benefits and costs of each approach.

### **8.2 Limitations**

There is much work that should be done to improve, calibrate, and validate the model used within this thesis.

Data from real-world engineering projects should be used to calibrate the model. Several functions within the model are overly simplified and could be modified to reflect the behavior witnessed in actual projects. Examples of such functions are: the scope quality function, the effects of coordination on quality and productivity, the effects of resource excess or shortfall on productivity, the prioritization of tasks and scope items, and the accomplishment rework, to name a few.

The representation of component and system scope quality within the model should be reviewed and revised. A complex range of factors influences the overall system quality including the quality of the original design, quality of component parts, process quality in assembly and integration, and emergent quality of the system as a whole. The function governing quality and rework in the model is simplistic and can be improved. Similarly, the representation of final scope quality in the composite project score should be updated to represent the same complex set of factors.

A planning effectiveness variable should be included in a future experiment. This variable would be used to control the accuracy of the baseline plans produced in the planning module.

The results of this experiment would give insight into which methods are most reliant on accurate planning and which are most robust to poor planning.

The forecasting function within each control method could be revised to include a more sophisticated statistical or machine learning technique which leverages a dataset of “historical data” to create cost and schedule forecasts for the trial data. This feature would allow for experiments which evaluate the benefits of combining an integrated project model with machine learning models.

The organizational, human, social, and cultural factors within the model are insufficient. The model should be revised to include organizational, social, and cultural factors such as team dynamics, personality, individual and team cognition, leadership and management, political dynamics, organizational competencies, and organizational behavior. It should also include human factors such as specialization, attitude, experience, interest, attention, incentive, and motivation. Including these factors would allow for experiments to explore the rich interaction between the product, process, and organizational domains.

The representation of dependency and coordination within the model is oversimplified. The model should be revised to represent dependency as a demand for coordination and coordination as a need for transferring information across interfaces. [19] This would also be an opportunity to incorporate a more complex representation of how concurrent work is executed based on that information transfer, how exceptions occur when insufficient information exists or when dependencies are violated, and how such activity affects quality and rework.

The model should be expanded to include more complex planning and measurement methods which are separate from the project control methods. The planning and measurement methods within the model are simplified and do not reflect the real-world behavior of these processes within projects. Furthermore, different organizations and projects use different techniques. The accuracy of cost and schedule estimates should decrease as scope or task complexity increases, as organizational expertise decreases, and as technological maturity decreases. Measurement activity should also drive cost within the task where it takes place. Incorporating these details into the model will allow for more realistic planning scenarios and exploration of the interaction effects between different planning, measurement, and control methods.

The implementation of measurement accuracy within the model should be adjusted to reflect the human aversion to “share bad news.” This could be done by creating a bias which would make measuring negative project information less accurate than positive information. Explicit testing activity could be built into the model as a way of increasing the chance of detecting latent defects that are otherwise randomly hidden.

The process of stakeholder priority negotiation was omitted from the model. This activity should be added and invoked when small marginal changes in project thresholds may make a large difference in the project outcome in the form of enabling product completion. Negotiation should also take place when project forecasts suggest the project should be cancelled. The concept of hidden stakeholder preferences could also be included which may manifest as requirements changes during a project or the changing of priorities and threshold values.

### 8.3 Additional Experiments

The experiment conducted within this thesis allowed for broad comparisons of project control methods in terms of project outcomes and an analysis of how the experimental variables affected those control methods. The experiment did not explore what differences between the control methods are important or how they influence project outcomes. An experiment that systematically controls for differences between project control methods would be valuable in understanding the factors that contribute or detract from project performance. For example, adding one risk level to the Earned Value Management method and examining the difference in outcomes under deterministic conditions.

There are several parameters in the model which were treated as constants for the sake of the experiment but which could be treated as variables in future experiments. These include the stakeholder priorities, human resource pool, material resource batch size, risk likelihoods and impacts, and the parameters governing control decisions and actions. A broad experiment varying each of these parameters to understand their impact on project outcomes as well as interaction effects between them would allow for better understanding of the project dynamics.

The experiment in this thesis revealed that both the EVM-ES and Simple Control methods performed best with a moderate amount of latency. A follow-on experiment should be conducted to explore why this is. It is hypothesized that these methods over-control at low latency levels, and so it would be interesting to adjust the control action parameters to explore whether this phenomenon persists and what causes it.

The model employs an extended version of earned value management which includes earned schedule calculations. It would be interesting to evaluate the performance of the traditional earned value method without earned schedule. Conducting the experiment on the traditional earned value method should be considered for future work.

The experiment in this thesis did not evaluate a control method which integrates product scope information at the system level. Conducting the experiment on a method which integrates system-level product scope information should be considered for future work.

It would also be valuable to study the actual implementation of earned value on Defense acquisition project to understand how earned value is most commonly applied. Are earned value extensions typically used on these projects, or is traditional earned value practiced? This information would give insight on what can be improved upon.

Finally, there are several more project control methods listed in Section 2.8 which would be interesting to evaluate. It would also be valuable to modify or combine control methods in an attempt to improve their performance.

THIS PAGE INTENTIONALLY LEFT BLANK

## **APPENDIX A: Acronyms**

ANSI/EIA: American National Standards Institute / Electronic Industries Association  
APB: Acquisition Program Baseline  
BIM: Business Information Modeling  
CAD: Computer Aided Design  
COVID-SARS: Coronavirus Disease  
C/SPEC: Cost/Schedule Planning and Control Specification  
DFARS: Defense Federal Acquisition Regulation Supplement  
DMM: Domain Mapping Matrix  
DoD: Department of Defense  
DoDAF: Department of Defense Architectural Framework  
DSM: Design Structure Matrix  
EVM: Earned Value Management  
EVM-ES: Earned Value Management with Earned Schedule  
EVMS: Earned Value Management System  
FAR: Federal Acquisition Regulation  
GERT: Graphical Evaluation and Review Technique  
INCOSE: International Council on Systems Engineering  
ISO: International Organization for Standardization  
LOE: Level of Effort  
MDM: Multidomain Matrix  
MRL: Multiple Risk Level  
NASA: National Aeronautics and Space Administration  
NRO: National Reconnaissance Office  
OBS: Organizational Breakdown Structure  
OMB: Office of Management and Budget  
OPM: Object-Process Methodology  
PERT: Program Evaluation and Review Technique  
PMB: Performance Measurement Baseline  
PMI: Project Management Institute  
PMBOK: Program Management Body of Knowledge  
PPML: Project-Product Lifecycle Management  
SAF/AQ: Assistant Secretary of the Air Force (Acquisition, Technology, & Logistics)  
TPR: True Positive Rate  
TNR: True Negative Rate  
WBS: Work Breakdown Structure

## APPENDIX B: Computational Model Documentation

### Programming Language and Development Environment:

Python Version: 3.8.2

Spyder Version: 4.1.3 and 4.1.5

### Libraries and Packages:

NumPy version 1.19.2

*An open source project aiming to enable numerical computing with Python.*

<https://numpy.org/>

Pandas version 1.0.5

*A fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language*

<https://pandas.pydata.org/>

Python Standard Library version 3.8

<https://docs.python.org/3/library/index.html>

The complete code for the model used for this thesis research can be obtained by contacting the author at [carsonc@alum.mit.edu](mailto:carsonc@alum.mit.edu). Appendix D contains summary code for the functions and arguments along with detailed code for the major modules.

### Model Structure

The product, process, and organizational domains of the project model are represented by tables, vectors, and matrices which contain information about the project scope items, tasks, humans, and resources. The information in this section details the base structure and constant values used for every experimental trial.

## Scope Items

Table 28: Base Scope Item List

ID	Title	Units	Nominal Effort (e <sub>a</sub> )	Complexity Value (x <sub>a</sub> )	Required Resources	PBS	Tier	Threshold (s <sub>s</sub> )	Target (s <sub>t</sub> )	Inter-Mediate (s <sub>r</sub> )	Plan (s <sub>p</sub> )
1	Project Charter	Documents	150	1.2	10	1.0.1	2	0	0	2	4
2	Project Plan	Documents	300	1.5	20	1.0.2	2	0	0	2	4
3	Preliminary System Design 1	Schematics	750	0.7	20	1.0.3.1	3	0	0	1	2
4	Detailed System Design 1	Schematics	1500	1.7	30	1.0.3.2	3	0	0	1	4
5	Detailed Subsystem Design 1.1	Schematics	1000	1.3	15	1.1.0	2	0	0	2	3
6	Detailed Subsystem Design 1.2	Schematics	750	1.2	15	1.2.0	2	0	0	2	3
7	Component 1.1.1	Components	625	0.7	100	1.1.1	2	3	4	8	10
8	Component 1.1.2	Components	750	0.8	200	1.1.2	2	3	4	8	10
9	Component 1.1.3	Components	1250	1.1	50	1.1.3	2	3	4	8	12
10	Component 1.2.1	Components	250	0.5	30	1.2.1	2	3	4	8	10
11	Component 1.2.2	Components	1000	1.4	300	1.2.2	2	3	4	8	12
12	Subsystem 1.1	Assemblies	500	1	150	1.1	1	3	4	6	8
13	Subsystem 1.2	Assemblies	1500	1.2	250	1.2	1	3	4	6	8
14	System 1	Systems	2500	1.3	500	1	0	3	4	5	6

Table 29: Base Product Breakdown Structure

PBS Number	Description
1	System 1
1.0.1	System 1 Design
1.1	Subsystem 1.1
1.1.0	Subsystem 1.1 Design
1.1.1	Component 1.1.1
1.1.2	Component 1.1.2
1.1.3	Component 1.1.3
1.2	Subsystem 1.2
1.2.0	Subsystem 1.2 Design
1.2.1	Component 1.2.1
1.2.2	Component 1.2.2



## Tasks

Table 30: Base Task List

ID	Title	WBS	Nominal Resource Ratio
1	Project Initiation	1.1	4
2	Project Planning	1.2	5
3	Preliminary System Design Task 1	3.0.1	2
4	Detailed System Design Task 1	3.0.2	3
5	Detailed Subsystem Design Task 1.1	3.1.0.1	5
6	Detailed Subsystem Design Task 1.2	3.2.0.1	3
7	Component 1.1.1 Fabrication and Test	3.1.1.1	10
8	Component 1.1.2 Fabrication and Test	3.1.2.1	20
9	Component 1.1.3 Fabrication and Test	3.1.3.1	5
10	Component 1.2.1 Fabrication and Test	3.2.1.1	10
11	Component 1.2.2 Fabrication and Test	3.2.2.1	15
12	Subsystem 1.1 Integration	3.1.0.2	20
13	Subsystem 1.2 Integration	3.2.0.2	15
14	Subsystem 1.1 Verification	3.1.0.3	20
15	Subsystem 1.2 Verification	3.2.0.3	10
16	System 1 Integration	3.0.3	10
17	System 1 Verification	3.0.4	30
18	System 1 Assembly	3.0.5	15
19	System 1 Environmental Testing	3.0.6	50
20	Project Closeout	1.3	5
21	Program Management	1.0	5
22	System Engineering	2.0	5

Table 31: Base Work Breakdown Structure

WBS Number	Title	Task ID	Control Account
0.0	Project		0.0
1.0	Program Management	21	1.0
1.1	Project Initiation	1	1.0
1.2	Project Planning	2	1.0
1.3	Project Closeout	20	1.0
2.0	System Engineering	22	2.0
3.0	System 1		3.0
3.0.1	Preliminary System Design Task 1	3	3.0
3.0.2	Detailed System Design Task 1	4	3.0
3.0.3	System 1 Integration	16	3.0
3.0.4	System 1 Verification	17	3.0
3.0.5	System 1 Assembly	18	3.0
3.0.6	System 1 Environmental Testing	19	3.0
3.1	Subsystem 1.1		3.1
3.1.0.1	Detailed Subsystem 1.1 Design Task	5	3.1
3.1.0.2	Subsystem 1.1 Integration	12	3.1
3.1.0.3	Subsystem 1.1 Verification	14	3.1
3.1.1	Component 1.1.1		3.1
3.1.1.1	Component 1.1.1 Fabrication and Test	7	3.1
3.1.2	Component 1.1.2		3.1
3.1.2.1	Component 1.1.2 Fabrication and Test	8	3.1
3.1.3	Component 1.1.3		3.1
3.1.3.1	Component 1.1.3 Fabrication and Test	9	3.1
3.2	Subsystem 1.2		3.2
3.2.0.1	Detailed Subsystem 1.2 Design Task	6	3.2
3.2.0.2	Subsystem 1.2 Integration	13	3.2
3.2.0.3	Subsystem 1.2 Verification	15	3.2
3.2.1	Component 1.2.1		3.2
3.2.1.1	Component 1.2.1 Fabrication and Test	10	3.2
3.2.2	Component 1.2.2		3.2
3.2.2.1	Component 1.2.2 Fabrication and Test	11	3.2

## Scope DSM (G):

Table 32: Base Scope Dependency Structure Matrix

		Scope Items													
Column scope item precedes row scope item		1-Project Charter	2-Project Plan	3-Prelim Sys Design	4-Detailed Sys Design	5-Detailed Sub Design 1.1	6-Detailed Sub Design 1.2	7-Component 1.1.1	8-Component 1.1.2	9-Component 1.1.3	10-Component 1.2.1	11-Component 1.2.2	12-Subsystem 1.1	13-Subsystem 1.2	14-System
Scope Items	1-Project Charter														
	2-Project Plan	1													
	3-Prelim Sys Design	1	1												
	4-Detailed Sys Design			1											
	5-Detailed Sub Design 1.1				1										
	6-Detailed Sub Design 1.2				1	1									
	7-Component 1.1.1					1									
	8-Component 1.1.2					1		1							
	9-Component 1.1.3					1		1	1						
	10-Component 1.2.1						1			1					
	11-Component 1.2.2						1				1				
	12-Subsystem 1.1					1		1	1	1					
	13-Subsystem 1.2						1				1	1			
	14-System												1	1	

### Task DSM (D) (Base):

Table 33: Base Task Dependency Structure Matrix

		Tasks																						
L - Leads: Finish-to-Start Relationship P - Pacing: Preceding task starts before succeeding task starts and finishes before finishes Column task precedes row task		1-Project Initiation	2-Project Planning	3-Prelim Sys Design	4-Detailed Sys Design	5-Detailed Sub 1.1 Design	6-Detailed Sub 1.2 Design	7-Comp 1.1.1 Fab and Test	8-Comp 1.1.2 Fab and Test	9-Comp 1.1.3 Fab and Test	10-Comp 1.2.1 Fab and Test	11-Comp 1.2.2 Fab and Test	12-Sub 1.1 Integration	13-Sub 1.2 Integration	14-Sub 1.1 Verification	15-Sub 1.2 Verification	16-System Integration	17-System Verification	18-System Assembly	19-System Environmental Testing	20-Project Closeout	21-Program Management	22-System Engineering	
Tasks	1-Project Initiation																							
	2-Project Planning	L																						
	3-Prelim Sys Design		P																					
	4-Detailed Sys Design			L																				
	5-Detailed Sub 1.1 Design				L																			
	6-Detailed Sub 1.2 Design				L	P																		
	7-Comp 1.1.1 Fab and Test					P																		
	8-Comp 1.1.2 Fab and Test					P		P																
	9-Comp 1.1.3 Fab and Test					P		L	L															
	10-Comp 1.2.1 Fab and Test						P			P														
	11-Comp 1.2.2 Fab and Test						P				P													
	12-Sub 1.1 Integration							L	L	L														
	13-Sub 1.2 Integration										L	L												
	14-Sub 1.1 Verification												L											
	15-Sub 1.2 Verification													L										
	16-System Integration													L	L									
	17-System Verification														L	L								
	18-System Assembly																L	L						
	19-System Environmental Testing																		L					
	20-Project Closeout																L	L	L	L			P	P
	21-Program Management		P																					
	22-System Engineering		P																					

### Scope to Task DMM (M):

Table 34: Base Scope to Task Domain Mapping Matrix

		Tasks																						
Element value represents fraction of scope in row assigned to each task in column		1-Project Initiation	2-Project Planning	3-Prelim Sys Design	4-Detailed Sys Design	5-Detailed Sub 1.1 Design	6-Detailed Sub 1.2 Design	7-Comp 1.1.1 Fab and Test	8-Comp 1.1.2 Fab and Test	9-Comp 1.1.3 Fab and Test	10-Comp 1.2.1 Fab and Test	11-Comp 1.2.2 Fab and Test	12-Sub 1.1 Integration	13-Sub 1.2 Integration	14-Sub 1.1 Verification	15-Sub 1.2 Verification	16-System Integration	17-System Verification	18-System Assembly	19-System Environmental Testing	20-Project Closeout	21-Program Management	22-System Engineering	
Scope Items	1-Project Charter	0.90																				0.10		
	2-Project Plan		0.80																				0.15	0.05
	3-Prelim Sys Design			0.80																			0.05	0.15
	4-Detailed Sys Design				0.80																		0.05	0.15
	5-Detailed Sub Design 1.1					0.80																	0.05	0.15
	6-Detailed Sub Design 1.2						0.80																0.05	0.15
	7-Component 1.1.1							0.80															0.05	0.15
	8-Component 1.1.2								0.80														0.05	0.15
	9-Component 1.1.3									0.80													0.05	0.15
	10-Component 1.2.1										0.80												0.05	0.15
	11-Component 1.2.2											0.80											0.05	0.15
	12-Subsystem 1.1												0.40		0.30								0.10	0.20
	13-Subsystem 1.2													0.40		0.30							0.10	0.20
	14-System																0.25	0.15	0.10	0.10	0.05	0.15	0.20	

## Scope Item Vector Structure

The scope item matrix (**P**) contains the set of 14 scope item column vectors for the project. The structure of each vector  $p_j$  for each scope item  $j$  is shown in Table 35.

Table 35: Scope Item Vector Structure

Row	Description
0	Target Scope Quantity ( $s_t$ ) (scope item units)
1	Required Scope (Intermediate + Target Scope) ( $s_r$ ) (scope item units)
2	Planned Scope Quantity ( $s_p$ ) (scope item units)
3	Minimum acceptable quantity for delivery ( $s_x$ ) (scope item units)
4	Actual Quantity Complete ( $s_a$ ) (scope item units)
5	Measured Quantity Complete ( $s_m$ ) (scope item units)
6	Actual Quality of Total Completed Scope ( $q_{a,n}$ )
7	Actual Quality of Completed Intermediate Scope ( $q_{a,i}$ )
8	Actual Quality of Completed Target Scope ( $q_{a,t}$ )
9	Measured Quality of Completed Intermediate Scope ( $q_{m,i}$ )
10	Measured Quality of Completed Target Scope ( $q_{m,t}$ )
11	Actual Nominal Effort ( $e_a$ ) (hours per scope item unit)
12	Estimated Nominal Effort ( $e_m$ ) (hours per scope item unit)
13	Actual Complexity Value ( $x_a$ )
14	Estimated Complexity Value ( $x_m$ )
15-28	Dependency upon scope items 1-14 (from row in Scope DSM <b>G</b> )
29	Quality target for intermediate scope ( $q_i$ )
30	Quality target for target scope ( $q_t$ )
31	Minimum acceptable quality of delivered scope ( $q_x$ )
32	Required resources (resource units)
33	Resource buffer (resource units)
34	Last unit of completed scope (scope item units)

## Task Vector Structure

The task matrix ( $\mathbf{V}$ ) contains the set of 22 task column vectors for the project. The structure of each vector  $\mathbf{v}_i$  for each task  $i$  is shown in

Table 36: Task Vector Structure

Row	Description
0	Task Active Status (true/false)
1-14	Quantity of each scope item $j$ assigned ( $s_{p,j}$ times $M_{ji}$ ) ( $s_{p,ji}$ )
15	Task budget ( $c_p$ ) (dollars)
16-29	Nominal effort for each scope item $j$ assigned ( $s_{p,j}$ times $e_{m,j}$ )
30-43	Planned nominal duration (hours) for each scope item assigned to task
44	Total planned duration (hours)
45	Length of planned work day (hours)
46	Planned duration ( $t_p$ ) (days)
47	Critical path status (calculated by PDM) (true/false)
48	Task Float (calculated by PDM) (days)
49	Required resources due to assigned scope (resource units)
50	Required resources due to team size (resource units)
51	Total required resources (resource units)
52	Task ready to begin status (true/false)
53	Humans assigned to task (team size)
54	Resources allocated to task (resource units)
55	Effort modifier due to resources (effect of non-nominal resource allocation)
56	Actual duration (hours) (increments in real time)
57	Length of actual work day
58	Actual duration (days) (increments in real time) ( $t_a$ )
59	Coordination performed ( $o$ ) (hours)
60	Work performed ( $w$ ) (hours)
61	Cost incurred ( $c_a$ ) (dollars)
62	Task complete status (true/false)
63	Estimated exception handling rate (control variable)

## Human Vector Structure

The team size for each task is accounted for within the task matrix  $\mathbf{V}$ . The structure for the human vector,  $\mathbf{h}$ , for the project is shown in Table 37.

Table 37: Human vector structure

Row	Description
1	Quantity of humans hired
2	Quantity of humans actively assigned to tasks
3	Quantity of humans available to be assigned
4	Status of hiring freeze (true/false)

## Resource Vector Structure

The team size for each task is accounted for within the task matrix  $\mathbf{V}$ . The structure for the human vector,  $\mathbf{h}$ , for the project is shown in Table 38.

Table 38: Resource vector structure

Row	Description
1	Amount of resources purchased (resource units)
2	Amount of resources consumed (resource units)
3	Amount of resources available (resource units)

## Model Constants and Variables

Symbol	Category	Title	Description	Setting	Value given
$u$	Variable	Uncertainty	Perfectly deterministic results	None	0
$u$	Variable	Uncertainty	Slightly uncertain results	Low	0.25
$u$	Variable	Uncertainty	Very uncertain results	High	1
$a$	Variable	Measurement Accuracy	Measurements match actual values	Perfect	0
$a$	Variable	Measurement Accuracy	Slight deviation between measured and actual values	High	0.25
$a$	Variable	Measurement Accuracy	Significant deviation between measured and actual values	Low	1
$l$	Variable	Control Latency	Constant Control Activity; No lag between measurements and control actions	None	0
$l$	Variable	Control Latency	Frequent Control Activity; Slight lag between measurements and control actions	Low	0.25
$l$	Variable	Control Latency	Infrequent Control Activity; Significant lag between measurements and control actions	High	1
$q_x$	Constant	Cancellation Quality	Minimum acceptable quality value for delivery of finished product	Base	0.9
$q_r$	Constant	Intermediate Quality	Threshold quality value for intermediate units of scope items	Base	0.85
$q_t$	Constant	Target Quality	Threshold quality value for target units of scope items	Base	1



$T_x$	Constant	Time Cancellation Threshold	Maximum acceptable duration for the project (set by stakeholders)	Base	4001
$C_x$	Constant	Cost Cancellation Threshold	Maximum acceptable cost for the project (set by stakeholders)	Base	\$ 80,000,000
$T_t$	Constant	Target Duration	Target duration for the project (set by stakeholders)	Base	1216
$C_t$	Constant	Target Cost	Target cost for the project (set by stakeholders)		\$ 15,364,723
$c_r$	Constant	Resource Cost	Cost per unit of resources in dollars	Base	
$c_w$	Constant	Wage	The cost of labor in dollars per hour	Base	40
	Constant	Overtime Factor	The factor applied to labor wage for overtime work	Base	1.5
	Constant	Fixed Overhead Rate	Factor applied to wage to determine fixed overhead. Represents fixed overhead costs.	Base	0.77
	Constant	Variable Overhead Rate	Factor applied to wage to determine variable overhead. Represents variable overhead costs.	Base	0.77
	Constant	Coordination Inefficiency	Applied as an exponent to team size resulting in sub-linear increase in productivity. Represents coordination inefficiencies as team size grows.	Base	0.9
$V_q$	Constant	Coordination Quality Bonus	Applied as a factor within the function defining how the optimal team size (set to 10) gives a relative increase to scope quality compared to smaller or larger team sizes	Base	0.1
$V_m$	Constant	Task Resource Effort Modifier	Applied as an exponent to the quantity of excess resources applied to a task. Results in sublinear growth in productivity as additional resources are applied.	Base	0.7
$V_r$	Constant	Resource penalty	Quality penalty for performing work without the required resource quantity	Base	0.5
$V_o$	Constant	Overtime penalty	Quality penalty per hour of overtime worked. Represents effect of exhaustion.	Base	0.025
	Constant	Exception Handling	Average exception handling for all tasks.	Base	0.2
	Constant	Tick	Number of days represented by an interaction of the model	Base	1

	Variable	Control Frequency	Ticks between execution of the control module	Base	$5 + 4*/I$
	Variable	Measurement Lag	Delay between measurement values and when those values are known to teams in ticks	Base	$2 + 2*/I$
	Variable	Adaptations Lag	Delay between the decision to administer a control action and when it takes effect in ticks	Base	$5 + 5*/I$
	Constant	Resource Batch	Resources are ordered in multiples of this quantity	Base	500
	Constant	Personnel Reserve	Threshold value used to signal need to hire more personnel	Base	20
	Constant	Initial Hire	Number of personnel hired at beginning of the project	Base	100
	Constant	Resource Defect Rate	Average ratio of resources which are unusable in an order	Base	0.02

# APPENDIX C: Validation Trial & Base Model Settings

## Scope Quantity Validation & Base Settings

### Low

Table 39: Validation Settings - Low Scope Quantities

ID	Title	Units	Threshold	Target	Intermediate
1	Project Charter	Documents	0	0	1
2	Project Plan	Documents	0	0	1
3	Preliminary System Design 1	Schematics	0	0	1
4	Detailed System Design 1	Schematics	0	0	1
5	Detailed Subsystem Design 1.1	Schematics	0	0	1
6	Detailed Subsystem Design 1.2	Schematics	0	0	1
7	Component 1.1.1	Components	1	1	1
8	Component 1.1.2	Components	1	1	1
9	Component 1.1.3	Components	1	1	1
10	Component 1.2.1	Components	1	1	1
11	Component 1.2.2	Components	1	1	1
12	Subsystem 1.1	Assemblies	1	1	1
13	Subsystem 1.2	Assemblies	1	1	1
14	System 1	Systems	1	1	1

### Moderate (Base)

Table 40: Validation Settings - Moderate Scope Quantities

ID	Title	Units	Threshold	Target	Intermediate
1	Project Charter	Documents	0	0	2
2	Project Plan	Documents	0	0	2
3	Preliminary System Design 1	Schematics	0	0	1
4	Detailed System Design 1	Schematics	0	0	1
5	Detailed Subsystem Design 1.1	Schematics	0	0	2
6	Detailed Subsystem Design 1.2	Schematics	0	0	2
7	Component 1.1.1	Components	3	4	8
8	Component 1.1.2	Components	3	4	8
9	Component 1.1.3	Components	3	4	8
10	Component 1.2.1	Components	3	4	8
11	Component 1.2.2	Components	3	4	8
12	Subsystem 1.1	Assemblies	3	4	6
13	Subsystem 1.2	Assemblies	3	4	6
14	System 1	Systems	3	4	5

## High

Table 41: Validation Settings - High Scope Quantities

ID	Title	Units	Threshold	Target	Intermediate
1	Project Charter	Documents	0	0	20
2	Project Plan	Documents	0	0	20
3	Preliminary System Design 1	Schematics	0	0	20
4	Detailed System Design 1	Schematics	0	0	20
5	Detailed Subsystem Design 1.1	Schematics	0	0	20
6	Detailed Subsystem Design 1.2	Schematics	0	0	20
7	Component 1.1.1	Components	10	15	20
8	Component 1.1.2	Components	10	15	20
9	Component 1.1.3	Components	10	15	20
10	Component 1.2.1	Components	10	15	20
11	Component 1.2.2	Components	10	15	20
12	Subsystem 1.1	Assemblies	10	15	20
13	Subsystem 1.2	Assemblies	10	15	20
14	System 1	Systems	10	15	20

## Team Size Validation & Base Settings

Table 42: Validation Settings - Humans per Task

Team Size	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Low	5	7	10	15	12	10	7	7	12	3	10	7	7	7	7	10	7	5	5	3	7	7
Moderate (Base)	10	15	20	30	25	20	15	15	25	5	20	15	15	15	15	20	15	10	10	5	15	15
High	20	30	40	60	50	40	30	30	50	10	40	30	30	30	30	40	30	20	20	10	30	30

## Resource Amount Validation & Base Settings

Table 43: Validation Settings - Resource Requirements per Task

Task Required Resources	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Low	2	3	1	2	3	2	5	10	3	5	8	10	8	10	5	5	15	8	25	3	3	3
Moderate (Base)	4	5	2	3	5	3	10	20	5	10	15	20	15	20	10	10	30	15	50	5	5	5
High	8	10	4	6	10	6	20	40	10	20	30	40	30	40	20	20	60	30	100	10	10	10

Table 44: Validation Settings - Resource Requirements per Scope Item

Scope Required Resources	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Low	5	10	10	15	8	8	50	100	25	15	150	75	125	250
Moderate (Base)	10	20	20	30	15	15	100	200	50	30	300	150	250	500
High	20	40	40	60	30	30	200	400	100	60	600	300	500	1000

# Structure Validation & Base Settings

## Flat

Table 45: Validation Settings - Flat Scope DSM

		Scope Items													
		1-Project Charter	2-Project Plan	3-Prelim Sys Design	4-Detailed Sys Design	5-Detailed Sub Design 1.1	6-Detailed Sub Design 1.2	7-Component 1.1.1	8-Component 1.1.2	9-Component 1.1.3	10-Component 1.2.1	11-Component 1.2.2	12-Subsystem 1.1	13-Subsystem 1.2	14-System
Scope Items	1-Project Charter														
	2-Project Plan	1													
	3-Prelim Sys Design	1	1												
	4-Detailed Sys Design	1	1	1											
	5-Detailed Sub Design 1.1	1	1	1	1										
	6-Detailed Sub Design 1.2	1	1	1	1	1									
	7-Component 1.1.1	1	1	1	1	1	1								
	8-Component 1.1.2	1	1	1	1	1	1	1							
	9-Component 1.1.3	1	1	1	1	1	1	1	1						
	10-Component 1.2.1	1	1	1	1	1	1	1	1	1					
	11-Component 1.2.2	1	1	1	1	1	1	1	1	1	1				
	12-Subsystem 1.1	1	1	1	1	1	1	1	1	1	1	1			
	13-Subsystem 1.2	1	1	1	1	1	1	1	1	1	1	1	1		
	14-System	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 46: Validation Settings - Flat Task DSM

		Tasks																					
		1-Project Initiation	2-Project Planning	3-Prelim Sys Design	4-Detailed Sys Design	5-Detailed Sub 1.1 Design	6-Detailed Sub 1.2 Design	7-Comp 1.1.1 Fab and Test	8-Comp 1.1.2 Fab and Test	9-Comp 1.1.3 Fab and Test	10-Comp 1.2.1 Fab and Test	11-Comp 1.2.2 Fab and Test	12-Sub 1.1 Integration	13-Sub 1.2 Integration	14-Sub 1.1 Verification	15-Sub 1.2 Verification	16-System Integration	17-System Verification	18-System Assembly	19-System Environmental Testing	20-Project Closeout	21-Program Management	22-System Engineering
Tasks	1-Project Initiation																						
	2-Project Planning	L																					
	3-Prelim Sys Design	L	L																				
	4-Detailed Sys Design	L	L	L																			
	5-Detailed Sub 1.1 Design	L	L	L	L																		
	6-Detailed Sub 1.2 Design	L	L	L	L	L																	
	7-Comp 1.1.1 Fab and Test	L	L	L	L	L	L																
	8-Comp 1.1.2 Fab and Test	L	L	L	L	L	L	L															
	9-Comp 1.1.3 Fab and Test	L	L	L	L	L	L	L	L														
	10-Comp 1.2.1 Fab and Test	L	L	L	L	L	L	L	L	L													
	11-Comp 1.2.2 Fab and Test	L	L	L	L	L	L	L	L	L	L												
	12-Sub 1.1 Integration	L	L	L	L	L	L	L	L	L	L	L											
	13-Sub 1.2 Integration	L	L	L	L	L	L	L	L	L	L	L	L										
	14-Sub 1.1 Verification	L	L	L	L	L	L	L	L	L	L	L	L	L									
	15-Sub 1.2 Verification	L	L	L	L	L	L	L	L	L	L	L	L	L	L								
	16-System Integration	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L							
	17-System Verification	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L						
	18-System Assembly	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L					
	19-System Environmental Testing	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L				
	20-Project Closeout	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L			
	21-Program Management	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L		
	22-System Engineering	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	

L - Leads: Finish-to-Start Relationship  
P - Pacing: Preceding task starts before succeeding task starts and finishes before finishes  
Column task precedes row task

### Mixed

The mixed validation trial used the base project structure shown in the **D** and **G** matrices earlier in this appendix.

## Full Network

The Scope DSM for the Full Network structure validation trial is an empty matrix to allow for simultaneous working of all scope items.

Table 47: Validation Settings - Full Network Scope DSM

		Scope Items													
		1-Project Charter	2-Project Plan	3-Prelim Sys Design	4-Detailed Sys Design	5-Detailed Sub Design 1.1	6-Detailed Sub Design 1.2	7-Component 1.1.1	8-Component 1.1.2	9-Component 1.1.3	10-Component 1.2.1	11-Component 1.2.2	12-Subsystem 1.1	13-Subsystem 1.2	14-System
Scope Items	1-Project Charter	■													
	2-Project Plan		■												
	3-Prelim Sys Design			■											
	4-Detailed Sys Design				■										
	5-Detailed Sub Design 1.1					■									
	6-Detailed Sub Design 1.2						■								
	7-Component 1.1.1							■							
	8-Component 1.1.2								■						
	9-Component 1.1.3									■					
	10-Component 1.2.1										■				
	11-Component 1.2.2											■			
	12-Subsystem 1.1												■		
	13-Subsystem 1.2													■	
	14-System														■

Table 48: Validation Settings - Full Network Task DSM

		Tasks																					
		1-Project Initiation	2-Project Planning	3-Prelim Sys Design	4-Detailed Sys Design	5-Detailed Sub 1.1 Design	6-Detailed Sub 1.2 Design	7-Comp 1.1.1 Fab and Test	8-Comp 1.1.2 Fab and Test	9-Comp 1.1.3 Fab and Test	10-Comp 1.2.1 Fab and Test	11-Comp 1.2.2 Fab and Test	12-Sub 1.1 Integration	13-Sub 1.2 Integration	14-Sub 1.1 Verification	15-Sub 1.2 Verification	16-System Integration	17-System Verification	18-System Assembly	19-System Environmental Testing	20-Project Closeout	21-Program Management	22-System Engineering
Tasks	1-Project Initiation		P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
	2-Project Planning	P		P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
	3-Prelim Sys Design	P	P		P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
	4-Detailed Sys Design	P	P	P		P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
	5-Detailed Sub 1.1 Design	P	P	P	P		P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
	6-Detailed Sub 1.2 Design	P	P	P	P	P		P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
	7-Comp 1.1.1 Fab and Test	P	P	P	P	P	P		P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
	8-Comp 1.1.2 Fab and Test	P	P	P	P	P	P	P		P	P	P	P	P	P	P	P	P	P	P	P	P	P
	9-Comp 1.1.3 Fab and Test	P	P	P	P	P	P	P	P		P	P	P	P	P	P	P	P	P	P	P	P	P
	10-Comp 1.2.1 Fab and Test	P	P	P	P	P	P	P	P	P		P	P	P	P	P	P	P	P	P	P	P	P
	11-Comp 1.2.2 Fab and Test	P	P	P	P	P	P	P	P	P	P		P	P	P	P	P	P	P	P	P	P	P
	12-Sub 1.1 Integration	P	P	P	P	P	P	P	P	P	P	P		P	P	P	P	P	P	P	P	P	P
	13-Sub 1.2 Integration	P	P	P	P	P	P	P	P	P	P	P	P		P	P	P	P	P	P	P	P	P
	14-Sub 1.1 Verification	P	P	P	P	P	P	P	P	P	P	P	P	P		P	P	P	P	P	P	P	P
	15-Sub 1.2 Verification	P	P	P	P	P	P	P	P	P	P	P	P	P	P		P	P	P	P	P	P	P
	16-System Integration	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P		P	P	P	P	P	P
	17-System Verification	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P		P	P	P	P	P
	18-System Assembly	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P		P	P	P	P
	19-System Environmental T	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P		P	P	P
	20-Project Closeout	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P		P	P
	21-Program Management	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P		P
	22-System Engineering	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	

L - Leads: Finish-to-Start Relationship  
 P - Pacing: Preceding task starts before succeeding task starts and finishes before finishes  
 Column task precedes row task

## Edge Case Validation Settings

Table 49: Validation Settings - Edge Case Team Sizes

Team Size	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Extreme - Low	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Extreme - High	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 50: Validation Settings - Edge Case Task Resource Requirements

Task Required Resources	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Extreme - Low		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Extreme - High		100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

Table 51: Validation Settings - Edge Case Scope Item Resource Requirements

Scope Required Resources	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Extreme - Low		1	1	1	1	1	1	1	1	1	1	1	1	1
Extreme - High		1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000



Table 52: Validation Settings - Edge Case Scope Quality Targets

Scope Quality	Intermediate Quality ( $q_f$ )	Target Quality ( $q_t$ )	Delivery Quality ( $q_x$ )
Extreme-Low	0.01	0.1	0.01
Extreme-High	2	4	2

## Extreme-Low

The extreme-low validation trial uses the flat scope and task dependency structures.

Table 53: Validation Settings: Extreme-Low Scope List

ID	Title	Units	Nominal Effort	Complexity Value	Required Resources	PBS	Tier	Thres-hold	Target	Inter-mediate	Plan
1	Project Charter	Documents	1	0.1	1	1.0.1	2	0	0	1	0
2	Project Plan	Documents	1	0.1	1	1.0.2	2	0	0	1	0
3	Preliminary System Design 1	Schematics	1	0.1	1	1.0.3.1	3	0	0	1	0
4	Detailed System Design 1	Schematics	1	0.1	1	1.0.3.2	3	0	0	1	0
5	Detailed Subsystem Design 1.1	Schematics	1	0.1	1	1.1.0	2	0	0	1	0
6	Detailed Subsystem Design 1.2	Schematics	1	0.1	1	1.2.0	2	0	0	1	0
7	Component 1.1.1	Components	1	0.1	1	1.1.1	2	1	1	1	0
8	Component 1.1.2	Components	1	0.1	1	1.1.2	2	1	1	1	0
9	Component 1.1.3	Components	1	0.1	1	1.1.3	2	1	1	1	0
10	Component 1.2.1	Components	1	0.1	1	1.2.1	2	1	1	1	0
11	Component 1.2.2	Components	1	0.1	1	1.2.2	2	1	1	1	0
12	Subsystem 1.1	Assemblies	1	0.1	1	1.1	1	1	1	1	0
13	Subsystem 1.2	Assemblies	1	0.1	1	1.2	1	1	1	1	0
14	System 1	Systems	1	0.1	1	1	0	1	1	1	0

## Extreme-High

The extreme-high validation trial uses the full network scope and task dependency structures.

Table 54: Validation Settings - Extreme-High Scope List

ID	Title	Units	Nominal Effort	Complexity Value	Required Resources	PBS	Tier	Thres-hold	Target	Inter-mediate	Plan
1	Project Charter	Documents	5000	10	1000	1.0.1	2	10	20	30	40
2	Project Plan	Documents	5000	10	1000	1.0.2	2	10	20	30	40
3	Preliminary System Design 1	Schematics	5000	10	1000	1.0.3.1	3	10	20	30	40
4	Detailed System Design 1	Schematics	5000	10	1000	1.0.3.2	3	10	20	30	40
5	Detailed Subsystem Design 1.1	Schematics	5000	10	1000	1.1.0	2	10	20	30	40
6	Detailed Subsystem Design 1.2	Schematics	5000	10	1000	1.2.0	2	10	20	30	40
7	Component 1.1.1	Components	5000	10	1000	1.1.1	2	10	20	30	40
8	Component 1.1.2	Components	5000	10	1000	1.1.2	2	10	20	30	40
9	Component 1.1.3	Components	5000	10	1000	1.1.3	2	10	20	30	40
10	Component 1.2.1	Components	5000	10	1000	1.2.1	2	10	20	30	40
11	Component 1.2.2	Components	5000	10	1000	1.2.2	2	10	20	30	40
12	Subsystem 1.1	Assemblies	5000	10	1000	1.1	1	10	20	30	40
13	Subsystem 1.2	Assemblies	5000	10	1000	1.2	1	10	20	30	40
14	System 1	Systems	5000	10	1000	1	0	10	20	30	40

## Quality Targets and Experimental Variable Validation Settings

Table 55: Validation Settings - Quality Targets and Experimental Variables

Validation Settings			
Title	Description	Setting	Value
Uncertainty	Slightly uncertain results	Low	0.25
Uncertainty	Very uncertain results	High	1
Measurement Accuracy	Slight deviation between measured and actual values	High	0.25
Measurement Accuracy	Significant deviation between measured and actual values	Low	1
Control Latency	Frequent Control Activity; Slight lag between measurements and control actions	Low	0.25
Control Latency	Infrequent Control Activity; Significant lag between measurements and control actions	High	1
Intermediate Quality	Quality target for intermediate scope ( $q_f$ )	Low	0.1
Intermediate Quality	Quality target for intermediate scope ( $q_f$ )	Moderate	0.8
Intermediate Quality	Quality target for intermediate scope ( $q_f$ )	High	1.6
Target Quality	Quality target for target scope ( $q_t$ )	Low	0.1
Target Quality	Quality target for target scope ( $q_t$ )	Moderate	1.5
Target Quality	Quality target for target scope ( $q_t$ )	High	3.1
Delivery Quantity	Minimum acceptable quantity for delivery of finished product ( $q_x$ )	Low	1
Delivery Quantity	Minimum acceptable quantity for delivery of finished product ( $q_x$ )	Moderate	3
Delivery Quantity	Minimum acceptable quantity for delivery of finished product ( $q_x$ )	High	10

## APPENDIX D: Integrated Project Model Code

```
# -*- coding: utf-8 -*-
"""
Integrated Project Model
In Support of MIT System Design and Management Master's Thesis:
"An Integrated Model-Based Approach to Improving Project Control in Department of Defense Acquisition"
@author: Christopher E. Carson
"""

import numpy as np
import pandas as pd
from datetime import datetime
import random
import PDM
from pathlib import Path

"""-----Functions-----"""

def random_array(length, mu, sigma):
    """generates a 1-d array of size length with random numbers from a gaussian distribution, mean mu, standard
        deviation sigma. Lower bound is 0 (does not return negative values)
    Input: length of array (int), mean array of size length of means for each array value, sigma array of size length of
        sigmas for each array value
    output: ndarray of size length of random values"""
    rand_array = np.zeros(length)
    for i in range(length):
        rand_array[i] = max(0, random.gauss(mu[i], sigma[i]))
    return rand_array

def purchase_resources(new_amount, resources, constants, Ca):
    """Takes in amount of resources to be purchased (scalar), resource dictionary and actual cost variable, returns
        updated resources dictionary and actual project cost"""
    resources['Purchased'] += (1 - constants['Resource_defect_rate']) * new_amount #adds the amount requested
        minus the defect rate to the resource pool
    resources['Available'] = resources['Purchased'] - resources['Consumed']
    Ca += new_amount * constants['Resource_cost'] * random.gauss(1, 0.25 * constants['Model_uncertainty']) #pays
        for the the total amount requested (new_amount) at an uncertain cost (reflects that
        costs change over time)

    return resources, Ca

def consume_resources(consumed_amount, resources):
    """Takes in resources dictionary and amount to be consumed, returns tuple of updated resources dictionary with
        updated consumed and available amounts (0) and actual amount of resources assigned
        to task (1)"""
    actual_consumed = 0
    if consumed_amount <= resources['Available']:
        actual_consumed = consumed_amount
    else:
        actual_consumed = max(0, resources['Available'])
    resources['Consumed'] += actual_consumed
    resources['Available'] = resources['Purchased'] - resources['Consumed']
    return resources, actual_consumed
```

```

def personnel_action(num_hired, num_assigned, humans):
    """updates the humans dictionary (Hired, Assigned, Available) based on the specified action.
    Enter number hired and number assigned for personnel action
    Returns tuple of updated humans dictionary (0) and number of humans actually assigned to the task (scalar) (1)"""
    actual_assigned = 0
    if not humans['Freeze']: #cannot hire during hiring freeze
        humans['Hired'] += max(0,num_hired)
    humans['Available'] = max(0,humans['Hired'] - humans['Assigned'])
    if humans['Available'] >= num_assigned:
        actual_assigned = num_assigned
    else:
        actual_assigned = max(humans['Available'], 0)
    humans['Assigned'] += actual_assigned
    humans['Available'] = max(0, humans['Hired'] - humans['Assigned'])
    return humans, actual_assigned

def item_quality(S,x,GT,i,checked):
    """Gives the expected quality of the last item produced in the quantity S"""
    #S is the required intermediate scope quantity vector for all scope items
    #x is the complexity vector for all scope items
    #GT is the scope quality DSM transpose (G.T)
    #i is the position in GT for the scope item whose quality is being calculated
    #checked is an empty list that keeps track of which items have been added as precessors as the recursions
        happen - if dependencies are circular, the cycle breaks
    #returns q[i] a float representing the quality of the last item produced in the set S
    pred_list = []
    qual = 1
    for j in range(len(GT[:,i])):
        if GT[j,i]:
            pred_list.append(j)
    if GT[:,i].sum() and i not in checked:
        checked.append(i)
        for k in pred_list:
            qual = min(qual, item_quality(S,x,GT,k, checked)[0]) #predecessor item quality is pacing - sets a minimum
                value. Cannot be greater than 1 (predecesor quality can not increase successor quality,
                only decrease)
    return (0.7 * max(0,S[i])**1/2 * qual / x[i]), qual #returns tuple of: (quality of item with all predecessor item
        qualities (qual) factored in (square root function), predecessors quality value)
    else:
        return (0.7 * max(0,S[i])**1/2 / x[i]), 1 #Base scope item with no predecessors does not have a dependent
            quality (qual) factor (square root function) - returns tuple so recursions will work, one is
            returned for items with no predecessors

def quant_for_qual(x,qt,qr,GT,St,Sr,Sp,planned_teams,scope_to_task_matrix,constants):
    """Returns (ndarray of length m) the quantity that must be produced in order to achieve quality threshhold q for
        item complexity x with scope dependencies given in G.Transpose (GT) and required
        quantity at threshhold of S
    Input: ndarrays of length m for scope item complexities x, subject quality threshholds q, intermediate quality
        threshholds qr for predecessor (i.e. all) items, required quanitites at quality threshhold S,
        and scope dependency matrix transposed (mxm) GT
    Returns: Sp: ndarray of length m of necessary planned quantities of scope item j to achieve quantity S at quality
        q"""

```

```

m = GT.shape[0]
n = scope_to_task_matrix.shape[1]
planning_quantities = np.zeros(m)
plan_for_intermediate = np.zeros(m)
plan_for_target = np.zeros(m)
Sp_copy = Sp.copy()
assumed_teams = np.ones(scope_to_task_matrix.shape[1]) * 17 #assumes all teams will be 17 people for
                    quantity planning purposes
team_bonus = np.maximum(np.zeros(n),constants['CoordQB']*(-1*np.power(assumed_teams-10,2)/27 + 3))
                    #Anticipated team bonus when planning for scope quantities, expected teams of 10 to
                    give maximum bonus according to "known" physics from past projects. Gives a decimal
                    value, meant to be used as (1+team_bonus) when applied as a coefficient to quality
scope_team_bonus = team_bonus @ scope_to_task_matrix.T #weighted average of team bonus applied to scope
                    items - gives a decimal value less than 1 - meant to be added to 1 when applied as a
                    factor to give INCREASE (bonus)

loop_run = True
loop_counter = 1000
while loop_run and loop_counter > 0:
    for j in range(m):
        plan_for_intermediate[j] = ((x[j] * qr[j])/(0.7 * (1+scope_team_bonus[j]) *
                    item_quality(np.maximum(Sr,Sp),x,GT,j,[])[1]))**2 + Sr[j] - 1 #intermediate predecessor
                    quality assumes an initial batch of max(1, target quantity) has been produced. Planned
                    scope must be equal to or greater than this.
        plan_for_target[j] = ((x[j] * qt[j])/(0.7 * (1+scope_team_bonus[j]) *
                    item_quality(np.maximum(Sr,Sp),x,GT,j,[])[1]))**2 + St[j] - 1 #target predecessor quantity
                    assumes at least the intermediate requires scope as been produced. Planned amount of
                    scope must be equal to or greater than this.
        planning_quantities[j] = int(max(plan_for_intermediate[j],plan_for_target[j],0)) + 1
        Sp[j] = planning_quantities[j]
        loop_run = not (Sp == Sp_copy).all() #loop runs until the quantities are stable
        Sp_copy = Sp.copy()
        loop_counter -= 1
    return planning_quantities

def actual_quality_of_item(scope_matrix, task_matrix, constants, G, scope_to_task_matrix):
    """Gives actual quality of completed scope
    Input: scope_matrix, task_matrix, constants -- scope_matrix, task_matrix, constants vector
    Returns ndarray of length 3xm with actual quality of completed scope, completed intermediate scope, and
    completed target scope - goes in scope_matrix[6:9] (rows 6,7,8). Also updates last whole
    scope quantity completed - goes in scope_matrix[34] """
    x = scope_matrix[13].copy()
    normalized_task_to_scope = np.diag(np.power((np.diagflat((1+scope_matrix[13]) * scope_matrix[11]) @
                    task_matrix[1:15]).sum(0,-1)) @ task_matrix[1:15].T @
                    np.diagflat((1+scope_matrix[13])*scope_matrix[11]) #nxm tells us what proportion of a
                    task is dedicated to each scope item as determined by required effort - based on actual
                    values here because it is used in determining true quality value
    #qpred = scope_matrix[7].copy() #actual intermediate quality of scope items - used for predecessor scope quality
                    calculation - old
    GT = scope_matrix[15:29].copy()
    m = GT.shape[0]
    n = task_matrix.shape[1]
    scope_item_resources_available = task_matrix[54] @ normalized_task_to_scope #amount of resources available
                    to each scope item

```

```

scope_resource_penalty = constants['Resource_violation'] * np.maximum(0,1 - (scope_item_resources_available
    / (scope_matrix[32] * scope_matrix[2]))) #quality penalty if not enough resources are
    available
predecessor_qual = np.zeros(m)
last_completed_quality = np.zeros(m)
team_working_scope = normalized_task_to_scope.copy() #separate matrix to show which scope items are being
    actively worked by a team assigned multiple scope items
scope_predecessors_complete = scope_dependencies_met(scope_matrix, G)
scope_items_complete = (scope_matrix[5] >= scope_matrix[1]) * (scope_matrix[9] >= scope_matrix[29]) *
    (scope_matrix[10] >= scope_matrix[30]) #measured quantity is greater than required
    intermediate quantity, measured intermediate quality is greater than threshold,
    measured target quality is greater than threshold
for j in range(scope_matrix.shape[1]):
    for i in range(task_matrix.shape[1]):
        if 0 < team_working_scope[i,j] < 1 and (team_working_scope[i] * scope_predecessors_complete *
            ~scope_items_complete).sum():
            #print('Team working scope',i,j,'is',team_working_scope[i,j])
            #print('Denominator is - i', i, 'team_working_scope[i]', team_working_scope[i],
                'scope_predecessors_complete', scope_predecessors_complete,
                '~scope_items_complete', ~scope_items_complete, 'product', (team_working_scope[i] *
                scope_predecessors_complete * ~scope_items_complete).sum())
            team_working_scope[i,j] = ((team_working_scope[i,j] * scope_predecessors_complete[j] *
                ~scope_items_complete[j]) / (team_working_scope[i] * scope_predecessors_complete *
                ~scope_items_complete).sum())
            #print('Updated Team working scope',i,j,'is',team_working_scope[i,j]) #debug
            #time.sleep(3) #debug
team_bonus = np.maximum(np.zeros(n),constants['CoordQB']*(-1*np.power(task_matrix[53]-
    10*(constants['Model_uncertainty']*random.uniform(0.5,1.5)),2)/27 + 3)) #team bonus
    is concave-down parabolic function with maximum at team size of 10 and intercepts (0
    bonus) at team sizes of 1 and 19. Implies maximum quality efficiency is with a team size
    of 10, diminishing at a parabolic rate until bonus reaches 0 at team size limits of 1 and 19.
    No bonus for teams larger than 19. Uncertainty means that actual optimal team size will
    vary uniformly - uncertainty setting of 1 will result in optimal team sizes ranging from 5 to
    15
overtime_penalty = np.maximum(0,task_matrix[0] * (task_matrix[57] - 8)) / 40 #quality penalty for overtime
    work. Applies to active tasks working overtime, with a maximum of 10% penalty when
    working 4 hours overtime.
scope_overtime_penalty = np.maximum(0.9, 1 - (overtime_penalty @ scope_to_task_matrix.T)) #task overtime
    penalty allocated to assigned scope items, value is 1 when no overtime is worked for that
    scope item, up to a minimum value of 0.9 (10% penalty) for maximum overtime (4 hours)
last_completed = scope_matrix[4].astype('int')
for j in range(m):
    last_completed_quality[j], predecessor_qual[j] = item_quality(last_completed,x,GT,j,[]) #new version - returns
        a tuple of (item j quality , item j predecessor quality)
for j in range(m): #for each scope item
    for k in range(last_completed[j]): #for each completed unit of that scope item
        if k >= scope_matrix[34,j]: #if there was an increase in quantity of units from last tick
            new_quality = random.gauss(1,constants['Model_uncertainty']*0.25) * (0.7 * (k+1)**(1/2) / x[j]) *
                predecessor_qual[j] * (1-scope_resource_penalty[j]) * (1+team_bonus[j]) *
                scope_overtime_penalty[j] #quality of k+1th item.
            scope_matrix[6,j] = ((k * scope_matrix[6,j] + new_quality) / (k+1)) #Average of the quality values for each
                completed scope item times a random uncertainty factor to account for quality variation
                and undiscovered rework

```

```

        if k+1 > (scope_matrix[2,j] - scope_matrix[1,j]): #For Intermediate Scope Completed
            diff = scope_matrix[2,j] - scope_matrix[1,j]
            scope_matrix[7,j] = ((k-diff) * scope_matrix[7,j] + new_quality) / (k+1-diff)
        if k+1 > (scope_matrix[2,j] - scope_matrix[0,j]): #For Target Scope Completed
            diff = scope_matrix[2,j] - scope_matrix[0,j]
            scope_matrix[8,j] = ((k-diff) * scope_matrix[8,j] + new_quality) / (k+1-diff)
    scope_matrix[34] = last_completed.copy()
    return scope_matrix[6:9], scope_matrix[34]

def update_schedule(task_matrix, D):
    """updates the project schedule resulting from updated task durations. Returns optimal project duration (along
        critical path), task critical path status (1,0), and task float
    Takes task_matrix as input
    Returns tuple(project_vectors[15,2], task_matrix[47], task_matrix[48]) to update project variables"""
    task_pdm_data = PDM.PDM(D,task_matrix[46])
    n = task_matrix.shape[1]
    day_counter = 0
    for i in range(n):
        task_matrix[47,i] = task_pdm_data[i].critical
        task_matrix[48,i] = task_pdm_data[i].float
        if task_pdm_data[i].critical == 1:
            day_counter += task_pdm_data[i].duration #shortest number of days that project could complete in based
                on inputs
    return (day_counter, task_matrix[47], task_matrix[48])

def dependencies_met(task_matrix, D):
    """Takes in task matrix and task dependency matrix D and returns array of length n of whether task dependencies
        are met such that a task may begin
    Input: Task_matrix, D
    Output: array of length n with values 1 = task dependency met, 0 task dependency not met"""
    n = task_matrix.shape[1]
    dependency_met = np.ones((n,n)) #tracks each individual dependency - assumes met unless for loop sets to not
        met so .all() method will work at end
    task_matrix[52,0] = 1 #starts first task so it begins if it is concurrent with other tasks. Should add logic to check
        that it doesn't have and predecessor start-finish relationships, but that is a limitation for
        now. Assumes first task in DSM has no predecessors as it stands.
    for i in range(n): #rows in D - tasks
        for k in range(n): #columns in D - predecessor tasks (dependencies)
            if D[i,k] == 1: #dependency value -- 1: FS, 2: Pacing (SS + FF)
                dependency_met[i,k] = dependency_met[i,k] * task_matrix[62][k] #if predecessor task is incomplete,
                    dependency is set to not met.
            if D[i,k] == 2: #dependency value -- 1: FS, 2: Pacing (SS + FF)
                if task_matrix[0][k] or task_matrix[59][k] + task_matrix[60][k] > 0 or task_matrix[62][k]: #If the predecessor
                    task has not started, dependency criteria is not met (all conditions are false, dependencies
                    met is false) - measures whether predecessor task is active or if progress has been made
                    (started and then stopped) or if complete
                    dependency_met[i,k] = dependency_met[i,k] * True
            else:
                dependency_met[i,k] = dependency_met[i,k] * False
    return dependency_met.all(1) #array of length n with value of 1 for tasks with all dependencies met and 0
        otherwise

def scope_dependencies_met(scope_matrix, G):

```



```

"""Takes current scope matrix and scope dependency matrix and determines which tasks have predecessors with
target and intermediate scope quantities and qualities met.
Input: scope_matrix, G (scope DSM)
Output: 1xm array with 1 for tasks with all scope dependency criteria satisfied and 0 otherwise
"""
m = scope_matrix.shape[1]
dependency_met = np.ones((m,m)) #tracks each individual dependency - assumes met unless for loop sets to not
met so .all() method will work at end
for j in range(m): #rows in G - scope items
    for k in range(m): #columns in G - predecessor scope items
        if G[j,k] == 1: #entry in DSM represents quality dependency - row item inherits quality from column item
            (propagation path)
            dependency_met[j,k] = dependency_met[j,k] * (scope_matrix[5,k] >= scope_matrix[1,k]) *
            (scope_matrix[9,k] >= scope_matrix[29,k]) * (scope_matrix[10,k] >= scope_matrix[30,k])
return dependency_met.all(1)

def task_complete(task_matrix, scope_matrix, D, scope_to_task_matrix, controls):
    """Takes in task and scope matrices, task dependencies D, and scope_to_task_matrix (based on csv) and returns
array of length n of whether task is complete (assigned scope completed at quality
standard). Also deactivates those tasks which are complete.
Input: Task_matrix, scope_matrix, D, scope_to_task_matrix
Output: array of length n with values 1 = task complete, 0 task not complete - goes in task_matrix[62] and array
of length n with active tasks status (1 is active, 0 is inactive) for task_matrix[0]"""
m = scope_matrix.shape[1]
n = task_matrix.shape[1]
task_to_scope_matrix = np.diag(np.power(task_matrix[1:15].sum(0,-1)) @ task_matrix[1:15].T #nxm
Smji = np.diagflat(scope_matrix[5]) @ scope_to_task_matrix #Quantity of measured scope complete per scope
item apportioned to each task
Spji = task_matrix[1:15].copy() #planned quantity of scope item j assigned to task i #mxn
qrmj = scope_matrix[9].copy() @ task_to_scope_matrix.T #weighted average of measured quality of
intermediate completed scope for task i #1xn
qrxj = scope_matrix[29].copy() @ task_to_scope_matrix.T #weighted average of threshold for intermediate
scope quality for task i #1xn
qtxj = scope_matrix[30].copy() @ task_to_scope_matrix.T #weighted average of threshold for target scope
quality for task i #1xn
qtmj = scope_matrix[10].copy() @ task_to_scope_matrix.T #weighted average of measured quality of target
completed scope for task i #1xn
for i in range(n): #set scope thresholds for each task based on whether other tasks contribute to the assigned
scope item. For successive tasks, each task is responsible for achieving an increasing
level of the quality threshold. For parrallel tasks, the required assigned quality is a
weighted average of the total value
    for j in range(m):
        if scope_to_task_matrix[j,i] and 1 in D[i]:
            qrxj[i] = scope_to_task_matrix[j,0:i+1].sum() * qrxj[i]
            qtxj[i] = scope_to_task_matrix[j,0:i+1].sum() * qtxj[i]
task_scope_complete = np.zeros(n)
tasks_complete = np.zeros(n) #sets all tasks to incomplete, so tasks that previously met completion criteria but
no longer meet the completion criteria will be marked incomplete, even in previously
marked complete
active_tasks = task_matrix[0].copy()
pacing_satisfied = np.ones((n,n))
human_changes = np.zeros(n)

```

```

task_scope_due_coord = task_matrix[59] / (scope_matrix[11] * scope_matrix[13] @ task_to_scope_matrix.T)
#Amount of scope complete due to task coord
task_scope_due_work = task_matrix[60] / (scope_matrix[11] @ task_to_scope_matrix.T) #Amount of scope
complete due to task work
quantity_task_scope_complete = np.minimum(task_scope_due_coord, task_scope_due_work) #ensures tasks
are completed in a waterfall manner - once the task has contributed the amount of work
needed for its portion of the assigned scope, it is complete
if controls['Control_method'] == 'None':
    tasks_complete = (Smji >= Spji).all(0)
    active_tasks = ~tasks_complete * active_tasks #Deactivates active tasks that have completed
    human_changes = -1 * task_matrix[53] * tasks_complete
else:
    incomplete_task_tracker = task_matrix[62].copy()
    for i in range(n): #if measured scope complete attributable to task is greater than planned quantity for each
scope item OR the task has finished the assigned portion of scope as measured by work
and coordination, AND intermediate and target quality thresholds are met, AND pacing
dependencies are complete, task is complete
        for k in range(n): #check for pacing relationships that must be satisfied before completing task i
            if D[i,k] == 2: #pacing dependency on task k satisfied if task k complete
                pacing_satisfied[i,k] = task_matrix[62][k]
            task_scope_complete[i] = ((Smji[:,i] >= Spji[:,i]).all() or quantity_task_scope_complete[i] * (1-
task_matrix[63,i] >= Spji[:,i].sum()) * (qrmj[i] >= qrxj[i]) * (qtmj[i] >= qtxj[i]) *
pacing_satisfied.all(1)[i] #Multiplying logical arrays is the same as an AND gate - all
conditions must be true for value to equal 1 at the end of the product. Adding quantity
task scope complete ensures that one task doesn't accomplish all the scope for scope
items split between multiple tasks, but doesn't account for exception handling
                if task_scope_complete[i]:
                    tasks_complete[i] = True
                    active_tasks[i] = False
                    human_changes[i] = -1 * task_matrix[53,i]
            task_matrix[0] += incomplete_task_tracker - tasks_complete #reactivates tasks that were previously complete
but no longer meet completion criteria
return tasks_complete, active_tasks, human_changes #values for task_matrix[62] - updated complete tasks,
and task_matrix[0] - updated active tasks

def create_budget(scope_matrix, task_matrix, planned_teams, constants):
    """Takes the scope & task matrices and planned team assignments and creates a budget vector with budget values
for each task. Base budget does not account for risks.
Input: scope_matrix, task_matrix, planned_teams, constants
Returns: task_matrix[15]"""
    n = task_matrix.shape[1]
    return task_matrix[44] * (planned_teams * (1+constants['VarOHR']) * constants['Wage'] + constants['FixedOHR']
* np.ones(n)) + task_matrix[51] * constants['Resource_cost'] #for task_matrix[15]

def humans_available_constraint(human_pool, D, planned_teams):
    """takes a pool of available humans and planned team sizes and reduces the planned team sizes if their concurrent
aggregate number exceeds the pool"""
    concurrent_tasks = []
    for i in range(D.shape[1]): #find the tasks are have pacing successor relationships with task i
        concurrence = list(np.where(D[i] == 2)[0])
        for j in concurrence[:]: #remove any tasks that have start-finish relationships with a successor task to task i
            (concurrent with i but start-to-finish with a concurrent task with i, which eliminates the
concurrence)

```

```

    for k in np.where(D[:,j] == 1)[0]:
        if k in concurrence:
            concurrence.remove(k)
    concurrent_tasks.append(concurrence)
concurrent_humans = []
for i in range(len(concurrent_tasks)): #find the total number of humans that are working a set of concurrent tasks
    concurrent_humans.append(planned_teams[i])
    for j in range(len(concurrent_tasks[i])):
        concurrent_humans[i] += planned_teams[j]
for i in range(len(concurrent_humans)): #check to see if the number of humans working each set of concurrent
    tasks is greater than the available pool
    if concurrent_humans[i] > human_pool: #if so, reduce the planned teams for that group of concurrent tasks so
        that the total number will be less than the available pool
        for j in range(len(concurrent_tasks[i])):
            planned_teams[j] -= int(((concurrent_humans[i] - human_pool) / len(concurrent_tasks[i]))+1) #reduce
            each planned team size for each offending current task by the total number of excess
            planned humans divided by the number of offending tasks (rounded up)
planned_teams = np.maximum(1, planned_teams)
return planned_teams

def create_plan(scope_matrix, task_matrix, project_vectors, planned_teams, margin, constants, controls,
               scope_to_task_matrix, D, task_list):
    #Update given scope & task matrices with initial planning data - no risk information included.
    #Planned scope quantity
    #Budget
    #Schedule
    m = scope_matrix.shape[1]
    n = task_matrix.shape[1]
    #create plan takes 1xm vectors: qt - scope_matrix[30] target quality thresholds, qr - scope_matrix[29]
        intermediate quality thresholds, x - scope_matrix[14] item complexities, GT
        scope_matrix[15:29] (mxm) scope DSM transposed, St - scope_matrix[0] target
        quantities, Sr - scope_matrix[1] intermediate quantities
    #quant for qual takes 1xm vectors: x - scope_matrix[14], qt - scope_matrix[30] target quality thresholds, qr -
        scope_matrix[29] intermediate quality thresholds, GT scope_matrix[15:29] (mxm) scope
        DSM transposed, St - scope_matrix[0] target quantities, Sr - scope_matrix[1]
        intermediate quantities, Sp - scope_matrix[2]
    #quant for qual assumes team size, so planned_team values won't affect output - doesn't use the vector at all
        anymore (can remove from function)
    Sp = quant_for_qual(scope_matrix[14].copy(),scope_matrix[30].copy(), scope_matrix[29].copy(),
        scope_matrix[15:29].copy(), scope_matrix[0].copy(), scope_matrix[1].copy(),
        scope_matrix[2].copy(), planned_teams,scope_to_task_matrix, constants)
    project_vectors[0:m,2] = Sp.copy()
    scope_matrix[2] = Sp.copy()
    scope_matrix = np.nan_to_num(scope_matrix)
    Spji = np.diagflat(scope_matrix[2]) @ scope_to_task_matrix #mxn
    Ev = np.diagflat(scope_matrix[12]) @ (np.eye(m) + np.diagflat(scope_matrix[14])) @ Spji

    '''alternative scheduling process to optimize teams to minimize float'''

    cumulative_teams = planned_teams.sum()
    loop_condition = True
    duration_tracker = []
    while loop_condition: #loop redistributes humans on teams but does not add to overall pool of humans.

```

```

Tp = np.eye(m) @ Ev @ np.diag(np.power(planned_teams,-1))
#Add schedule margins based on control method
if controls['Control_method'] == 'EVM':
    Tp = EVM_planning_margin(Tp, controls, margin)
if controls['Control_method'] == 'MRL0' or controls['Control_method'] == 'MRL1' or controls['Control_method']
    == 'MRL2' or controls['Control_method'] == 'MRL3' or controls['Control_method'] ==
    'MRL4' or controls['Control_method'] == 'MRL5':
    Tp = MRL_exception_handling(Tp, controls)
if controls['Control_method'] == 'MRL1' or controls['Control_method'] == 'MRL2' or controls['Control_method']
    == 'MRL3' or controls['Control_method'] == 'MRL4' or controls['Control_method'] ==
    'MRL5':
    Tp = MRL_planning_margin(Tp, margin)
#sum of rows in Tp (sum for each column vector)
TP = np.sum(Tp, axis = 0)
#update task matrix with newly calculated values
task_matrix[1:15] = Spji.copy()
task_matrix[16:30] = Ev.copy()
task_matrix[30:44] = Tp.copy()
task_matrix[44] = TP.copy()
task_matrix[46] = TP / task_matrix[45]
updated_schedule_data = update_schedule(task_matrix, D)
project_vectors[m+1,2] = updated_schedule_data[0] #Planned schedule in days - does not include global risk
margin - effort and resource margins were included in those task-level quantities
task_matrix[47] = updated_schedule_data[1].copy() #critical path tasks
task_matrix[48] = updated_schedule_data[2].copy() #task float
duration_tracker.append(updated_schedule_data[0])
if len(duration_tracker) > 5 and duration_tracker[-1] == min(duration_tracker): #once the loop has run 5 times,
    stop the next iteration that produces a minimum value for project duration (heuristic)
    loop_condition = False
if len(duration_tracker) > 10 and duration_tracker[-1] < (sum(duration_tracker)/len(duration_tracker)): #once
    the loop has run 5 times, stop the next iteration that produces a minimum value for
    project duration (heuristic)
    loop_condition = False
if len(duration_tracker) > 100:
    loop_condition = False
if (task_matrix[48] > 0).any():
    minimum_positive_float = max(1,np.amin(task_matrix[48][np.where(task_matrix[48] > 0)[0]])) #finds the
    smallest float value greater than zero
    relative_float = ~task_matrix[47].astype(bool) * task_matrix[48] / minimum_positive_float #for those tasks
    with float, gives a relative scale of the amount of float with a maximum of 5. Higher
    numbers means the task has more float > should lose more people
    float_reduction = np.minimum(3, planned_teams * task_matrix[48].astype(bool), relative_float).astype(int)
    #returns an integer of the lesser of: the existing planned team size, 5, or the relative
    float value. This is the reduction in team size for tasks with float - can't be greater than
    the planned team size (can't be negative), capping it at 5
    CP_increase = np.minimum(3, (Ev.sum(0) * task_matrix[47] * float_reduction.sum() / (Ev.sum(0) *
    task_matrix[47]).sum()).astype(int)) #Max increase of 5 - distributes the humans saved
    from tasks with float to the critical path proportionate to the nominal effort for each
    task. Task-wise nominal effort (EV.sum(0)) times critical path status (task_matrix[47],1 or
    0) time hours saved from float tasks divided by total effort for critical path tasks
    ((EV.sum(0) * task_matrix[47]).sum())
#add humans to teams on critical path; remove humans from teams with excess float
planned_teams = np.maximum(1, planned_teams + CP_increase - float_reduction)

```

```

        #if rounding reduced the number of total people on teams by more than the number of teams, add one to
        each team
    if cumulative_teams - planned_teams.sum() > n:
        planned_teams += (np.ones(n) * (cumulative_teams - planned_teams.sum()))/n.astype(int)
    #reduce planned teams if concurrent active tasks exceed the pool of humans
    planned_teams = humans_available_constraint(constants['Initial_hire'], D, planned_teams)
task_matrix[49] = np.sum(np.diag(scope_matrix[32]) @ Spji, axis = 0)
task_matrix[50] = planned_teams * task_list['Nominal Resource Ratio']
task_matrix[51] = (np.sum(np.diag(scope_matrix[32]) @ Spji, axis = 0) + planned_teams * task_list['Nominal
    Resource Ratio'])
task_matrix[15] = create_budget(scope_matrix, task_matrix, planned_teams, constants) #Planned cost - does not
    include includes global risk margin - effort and resource margins were included in those
    task-level quantities
project_vectors[m,2] = np.sum(task_matrix[15]) #planned project cost (budget)
if controls['Control_method'] == 'EVM':
    task_matrix[15] = EVM_risk(task_matrix, project_vectors, margin)[0][15]
    project_vectors[m,2] = np.sum(task_matrix[15]) * 1.05 #add 5% management reserve
return scope_matrix, task_matrix, project_vectors, planned_teams

def ready_task(task_matrix, resources, planned_teams, D, humans):
    """Determines whether tasks are ready to begin and returns 1 if so
    Input: task matrix, resources, D
    Output: task ready vector, values are 1 or 0 for each task, length n. Corresponds to task_matrix[52]"""
    n = task_matrix.shape[1]
    dependence_met = dependencies_met(task_matrix, D)
    task_ready = task_matrix[52].copy()
    #if humans available, resources available, and dependencies met: set task ready = 1
    for i in range(n):
        if resources['Available'] >= 0 and humans['Available'] >= 0 and dependence_met[i]:
            task_ready[i] = 1
    return task_ready #for task_matrix[52]

def execute_tasks(scope_matrix, task_matrix, D, G, scope_to_task_matrix, constants, resources, humans,
    planned_teams, task_list, controls):
    """Takes the current scope and task matrices as well as model constants and executes the project-product model,
    iterating one tick at a time
    Input: scope_matrix, task_matrix, constants
    Output: scope_matrix, task_matrix, humans, resources after 1 model tick"""
    m = scope_matrix.shape[1]
    n = task_matrix.shape[1]
    human_changes = np.zeros(n)
    normalized_task_to_scope = np.diag(np.power((np.diagflat((1+scope_matrix[14]) * scope_matrix[12]) @
        task_matrix[1:15]).sum(0,-1)) @ task_matrix[1:15].T @
        np.diagflat((1+scope_matrix[14])*scope_matrix[12]) #nxm tells us what proportion of a
        task is dedicated to each scope item as determined by required effort - based on
        measured values here because it is used in teams determining where to focus work

    Spji = np.diagflat(scope_matrix[2]) @ scope_to_task_matrix #mxn
    Smji = np.diagflat(scope_matrix[5]) @ scope_to_task_matrix #Quantity of measured scope complete per scope
        item apportioned to each task
    Srji = np.diagflat(scope_matrix[1]) @ scope_to_task_matrix #planned quantity of scope item j assigned to task i
        #mxn
    qmrji = np.diagflat(scope_matrix[9]) @ scope_to_task_matrix.astype(bool)

```

```

qmtji = np.diagflat(scope_matrix[10]) @ scope_to_task_matrix.astype(bool)
qrji = np.diagflat(scope_matrix[29]) @ scope_to_task_matrix.astype(bool)
qtji = np.diagflat(scope_matrix[30]) @ scope_to_task_matrix.astype(bool)

xSpji = np.diagflat(scope_matrix[13]) @ normalized_task_to_scope.T #mxn Amount of scope complexity
        attributable to each task
Uncertain_CoordWI = random.gauss(1,0.1*constants['Model_uncertainty']) * constants['CoordWI']

#update completed tasks - this also marks them not complete if they no longer meet quality/quantity thresholds
task_matrix[62], task_matrix[0], human_changes = task_complete(task_matrix, scope_matrix, D,
        scope_to_task_matrix, controls)

if task_matrix[62].all(): #Represents the customer rejecting the project - not enough scope is delivered
    incomplete_tracker = task_matrix[62].copy()
    task_matrix[62] = (Smji.sum(0) >= Srji.sum(0)) * task_matrix[62]
    task_matrix[62] = (qmrji >= qrji).all(0) * task_matrix[62]
    task_matrix[62] = (qmtji >= qtji).all(0) * task_matrix[62]
    reactivated_tasks = incomplete_tracker - task_matrix[62]
    # print('tasks reactivated', reactivated_tasks) #debug
    task_matrix[0] += reactivated_tasks
    for j in range(m):
        if Smji.sum(1)[j] < Srji.sum(1)[j] or scope_matrix[9,j] < scope_matrix[29,j] or scope_matrix[10,j] <
            scope_matrix[30,j]:
            scope_matrix[2,j] += 1

#return humans to pool from completed tasks and add where team size has changed due to project control
for i in range(n):
    humans = personnel_action(0,human_changes[i],humans)[0]
    #look for changes in team size due to project control
    if task_matrix[0,i]: #for active tasks
        if planned_teams[i] != task_matrix[53,i]: #If planned team size different than assigned team size
            humans, human_changes[i] = personnel_action(0,(planned_teams[i] - task_matrix[53,i]),humans) #modify
                by the difference
task_matrix[53] += human_changes

prior_coord_amount = task_matrix[59].copy()
prior_work_amount = task_matrix[60].copy()

#determine which tasks are ready to start
task_matrix[52] = ready_task(task_matrix, resources, planned_teams, D, humans)

#Activate new tasks with conditions met, perform work and coordination on active tasks - ensure all active tasks
        have at least one person assigned - ensure tasks are resourced if available
for i in range(n):
    # print('Task active and no humans condition:',task_matrix[0,i] and not task_matrix[53,i])
    #Activate tasks that are ready to begin but incomplete & assign humans & resources
    if task_matrix[52,i] and not task_matrix[62,i] and not task_matrix[0,i]:
        task_matrix[0,i] = True
        task_matrix[57] = np.minimum(12*np.ones(n), task_matrix[57] + controls['Overtime'])
        humans, task_matrix[53,i] = personnel_action(0,planned_teams[i],humans) #assign humans
        # print("Humans assigned", task_matrix[53]) #debug
        resources, task_matrix[54,i] = consume_resources(controls['Resource_modifier']*task_matrix[51,i],
            resources) #assign resources based on planned resources for each task

```

```

if task_matrix[0,i] and not task_matrix[53,i]: #task active but no humans assigned
    # print('Active task without humans!', humans) #debug
    humans, task_matrix[53,i] = personnel_action(0,planned_teams[i],humans) #assign humans
    # print(task_matrix[53,i]) #debug
if task_matrix[0,i] and task_matrix[54,i] < controls['Resource_modifier']*task_matrix[51,i]: #if task active but
    has insufficient resources
    # print('adding resources to task', i)
    resources, task_matrix[54,i] = consume_resources(controls['Resource_modifier']*task_matrix[51,i],
        resources) #allocate resources
#Perform work and coordination for tasks that are active, update duration active
if task_matrix[0,i]:
    task_matrix[55,i] = (((max(1,task_matrix[54,i] -
        task_matrix[49,i]))/(max(1,task_matrix[53,i])*task_list['Nominal Resource
        Ratio'][i+1]))**constants['Resource_exponent']) #Update TREM: ratio of assigned to
        required nominal resources raised to a sub-unitary exponent for sub-linear scaling -
        modifier for work (nominal resource level = 1)
    task_matrix[56,i] += constants['Tick'] * task_matrix[57,i] #duration
    task_matrix[58,i] += constants['Tick']
    # print('Hours:', constants['Tick'] * task_matrix[57,i],"times effective people:",
        (task_matrix[53,i]**constants['CoordWl']), "times effective complexity:", xSpji.sum(0)[i],
        "times TREM:",task_matrix[55,i],"divided by effective complexity plus one:",
        (xSpji.sum(0)[i] + 1)) #debug
    task_matrix[59,i] += constants['Tick'] * task_matrix[57,i] *
        random.gauss(1,0.1*constants['Model_uncertainty']) *
        (task_matrix[53,i]**Uncertain_CoordWl) * xSpji.sum(0)[i] * task_matrix[55,i] /
        (xSpji.sum(0)[i] + 1) #Increment coordination performed - function of humans and
        resources assigned to task, split with work based on scope complexity
    task_matrix[60,i] += constants['Tick'] * task_matrix[57,i] *
        random.gauss(1,0.1*constants['Model_uncertainty']) *
        (task_matrix[53,i]**Uncertain_CoordWl) * task_matrix[55,i] / (xSpji.sum(0)[i] + 1)
        #increment work performed - function of humans and resources assigned to task, split
        with coordination based on scope complexity
    task_matrix[61,i] += constants['Tick'] * task_matrix[57,i] *
        random.gauss(1,0.1*constants['Model_uncertainty']) * constants['Wage'] *
        task_matrix[53,i] + constants['Tick'] * (task_matrix[57,i] - 8) * constants['Wage'] *
        constants['VarOHR'] * task_matrix[53,i] #increment cost incurred - Wage for time
        worked plus overtime for overtime worked for each human

#Update scope items
scope_coord_complete = np.zeros(m)
scope_work_complete = np.zeros(m)
scope_predecessors_complete = scope_dependencies_met(scope_matrix, G)
scope_items_complete = (scope_matrix[5] >= scope_matrix[1]) * (scope_matrix[9] >= scope_matrix[29]) *
    (scope_matrix[10] >= scope_matrix[30])
incremental_work = task_matrix[60] - prior_work_amount
incremental_coord = task_matrix[59] - prior_coord_amount

for i in range(n): #adding up all of the work done this cycle by each task for each scope item. Incremental
    approach to enable waterfall handling of tasks with multiple scope items assigned
    incremental_work_copy = incremental_work[i]
    incremental_coord_copy = incremental_coord[i]
    for j in range(m): #calculate the amount of new productive work this cycle
        if normalized_task_to_scope[i,j] >= 1: #for tasks with only one scope item assigned

```

```

scope_work_complete[j] += (incremental_work[i]) * max(0,min(1,random.gauss(1 -
    constants['Exception_handling'],constants['Exception_handling']*constants['Model_unce
    rtainty']))) / scope_matrix[11,j] #New scope units complete (due to productive work) for
    tasks with only one assigned scope item is the previously accomplished productive work
    for that scope item plus the amount of work accomplished this cycle times an
    inefficiency coefficient to represent non-nominal work. This coefficient is between 0
    and 1, randomly determined from a normal distribution centered on 1-the exception
    handling constant, with a standard deviation of the exception constant times the
    uncertainty modifier. This gives the amount of "effective" productive work. That amount
    of work is divided by the nominal effort for the scope item to determine the amount of
    scope produced from the (productive) work.
scope_coord_complete[j] += (incremental_coord[i]) * max(0,min(1,random.gauss(1 -
    constants['Exception_handling'],constants['Exception_handling']*constants['Model_unce
    rtainty']))) / (scope_matrix[13,j] * scope_matrix[11,j]) #coord is
incremental_work[i] = 0 #ensure work and coord for the task are not used twice
incremental_coord[i] = 0
if 0 < normalized_task_to_scope[i,j] < 1: #For tasks with more than one scope item assigned
    if scope_predecessors_complete[j] and not scope_items_complete[j] and (normalized_task_to_scope[i] *
        scope_predecessors_complete * ~scope_items_complete).sum(): #if the jth scope item
        assigned to task i has its predecessor scope dependencies met but is not complete,
        perform additional work on a pro-rated basis. Work is split between all scope items that
        are ready for work according to the share of the task that the scope item work represents
        scope_work_complete[j] += (incremental_work[i]) * ((normalized_task_to_scope[i,j] *
            scope_predecessors_complete[j] * ~scope_items_complete[j]) /
            (normalized_task_to_scope[i] * scope_predecessors_complete *
            ~scope_items_complete).sum()) * max(0,min(1,random.gauss(1 -
            constants['Exception_handling'],constants['Exception_handling']*constants['Model_unce
            rtainty']))) / scope_matrix[11,j] #work from this cycle is prorated to assigned scope
            whose dependencies are satisfied and which are not complete, multiplied by a random
            inefficiency coefficient between 0 and 1 to represent non-nominal work
        scope_coord_complete[j] += (incremental_coord[i]) * ((normalized_task_to_scope[i,j] *
            scope_predecessors_complete[j] * ~scope_items_complete[j]) /
            (normalized_task_to_scope[i] * scope_predecessors_complete *
            ~scope_items_complete).sum()) * max(0,min(1,random.gauss(1 -
            constants['Exception_handling'],constants['Exception_handling']*constants['Model_unce
            rtainty']))) / (scope_matrix[13,j] * scope_matrix[11,j])
        incremental_work[i] = max(0,incremental_work[i] - incremental_work_copy *
            ((normalized_task_to_scope[i,j] * scope_predecessors_complete[j] *
            ~scope_items_complete[j]) / (normalized_task_to_scope[i] *
            scope_predecessors_complete * ~scope_items_complete).sum())) #ensure work and
            coord for the task are not used twice
        incremental_coord[i] = max(0, incremental_coord[i] - incremental_coord_copy *
            ((normalized_task_to_scope[i,j] * scope_predecessors_complete[j] *
            ~scope_items_complete[j]) / (normalized_task_to_scope[i] *
            scope_predecessors_complete * ~scope_items_complete).sum()))
    elif not (normalized_task_to_scope[i] * scope_predecessors_complete * ~scope_items_complete).sum():
        #if there isn't an obvious waterfall based on dependencies satisfied and scope incomplete,
        apply work across all assigned scope based on normalized task to scope
        scope_work_complete[j] += (incremental_work[i]) * normalized_task_to_scope[i,j] *
            max(0,min(1,random.gauss(1 -
            constants['Exception_handling'],constants['Exception_handling']*constants['Model_unce
            rtainty']))) / scope_matrix[11,j]

```



```

scope_coord_complete[j] += (incremental_coord[i]) * normalized_task_to_scope[i,j] *
    max(0,min(1,random.gauss(1 -
    constants['Exception_handling'],constants['Exception_handling']*constants['Model_unce
    rtainty']))) / (scope_matrix[13,j] * scope_matrix[11,j])
incremental_work[i] = 0
incremental_coord[i] = 0

incremental_scope_completed = np.minimum(scope_coord_complete, scope_work_complete) #new code for
    for-loop
scope_matrix[4] += incremental_scope_completed #new code for for-loop
scope_matrix[5] += random_array(m, incremental_scope_completed,
    0.25*incremental_scope_completed*constants['Model_measurement_accuracy'])
    #Measured quantity of scope complete - Measurement accuracy is uniform centered on
    actual value, with sigma = 0.25*mean

prior_quality = scope_matrix[7:9].copy() #saves the prior quality of intermediate [0] and target [1] scope
scope_matrix[6:9], scope_matrix[34] = actual_quality_of_item(scope_matrix, task_matrix, constants, G,
    scope_to_task_matrix) #Actual quality of completed total, intermediate, and target scope
incremental_intermediate_quality = scope_matrix[7] - prior_quality[0]
incremental_target_quality = scope_matrix[8] - prior_quality[1]
scope_matrix[9] += random_array(m, incremental_intermediate_quality,
    0.25*incremental_intermediate_quality*constants['Model_measurement_accuracy'])
    #Measure change in intermediate quality and update values
scope_matrix[10] += random_array(m, incremental_target_quality,
    0.25*incremental_target_quality*constants['Model_measurement_accuracy'])
    #measure change in target quality and update values

return scope_matrix, task_matrix, resources, humans

def execution(scope_matrix, task_matrix, D, G, scope_to_task_matrix, constants, resources, humans,
    planned_teams, task_list, Ta, Ca, closing_criteria, project_vectors,
    original_project_vectors, priorities, controls, RL_plans, filenotes, record_details, margin,
    control_accounts, risks):
    """Performs actual project execution. Performs initial project actions (hiring, resource purchase), then iterates
    execution cycles and ticks actual time counter Ta.
    Updates scope_matrix, task_matrix, resources, and humans with each cycle
    Returns scope_matrix, task_matrix, resources, humans, Ta, Ca, closing_criteria, priorities"""
    m = scope_matrix.shape[1]
    n = task_matrix.shape[1]
    metrics_df, scope_metrics_df = record_metrics(scope_matrix, task_matrix, resources, humans, Ta, Ca)
    all_forecast_df = pd.DataFrame(data = np.zeros(m+2), columns = [Ta], index = ['Scope Item 1', 'Scope Item 2',
    'Scope Item 3', 'Scope Item 4', 'Scope Item 5', 'Scope Item 6', 'Scope Item 7', 'Scope Item
    8', 'Scope Item 9', 'Scope Item 10', 'Scope Item 11', 'Scope Item 12', 'Scope Item 13',
    'Scope Item 14', 'Project Cost', 'Project Duration']) #initialize all_forecasts_df
    project_EVM_metrics = ['EV','PV','AC','CV','CPI','SV','SPI','EAC','TCPI','ETC','ES','TV','TPI','TEAC']
    all_project_metrics_df = pd.DataFrame(data = np.zeros((1, len(project_EVM_metrics))), index = [Ta], columns =
    project_EVM_metrics)
    resources, Ca = purchase_resources(np.sum(task_matrix[51]), resources, constants, Ca)
    humans = personnel_action(constants['Initial_hire'],0,humans)[0]
    if humans['Freeze'] == 1:
        humans['Hired'] = constants['Initial_hire']
    while not(closing_criteria['Scope_Complete'] or closing_criteria['Cost'] or closing_criteria['Time']):
        Ta += constants['Tick']

```

```

scope_matrix, task_matrix, resources, humans = execute_tasks(scope_matrix, task_matrix, D, G,
                    scope_to_task_matrix, constants, resources, humans, planned_teams, task_list, controls)

#Hire more humans if none left - move this to basic control function
if humans['Available'] < constants['Personnel_reserve'] and not humans['Freeze']:
    humans = personnel_action(constants['Personnel_reserve'],0,humans)[0]

#Purchase more resources if none left - move this to basic control function
if not resources['Available'] and not resources['Freeze']:
    resource_order = int((controls['Resource_modifier']*task_matrix[51].sum() - task_matrix[54].sum()) /
                        constants['Resource_batch']) #number of batches to order
    resources, Ca = purchase_resources(resource_order * constants['Resource_batch'], resources, constants, Ca)

#this is basic control - changing planned quantities so that enough is completed. Applies to control methods
other than None
Spji = np.diagflat(scope_matrix[2]) @ scope_to_task_matrix #mxn
Smji = np.diagflat(scope_matrix[5]) @ scope_to_task_matrix #Quantity of measured scope complete per scope
item apportioned to each task
Srji = np.diagflat(scope_matrix[1]) @ scope_to_task_matrix #planned quantity of scope item j assigned to task i
#mxn
qmrji = np.diagflat(scope_matrix[9]) @ scope_to_task_matrix.astype(bool)
qmtji = np.diagflat(scope_matrix[10]) @ scope_to_task_matrix.astype(bool)
qrji = np.diagflat(scope_matrix[29]) @ scope_to_task_matrix.astype(bool)
qtji = np.diagflat(scope_matrix[30]) @ scope_to_task_matrix.astype(bool)

#Update resources needed
for i in np.where(task_matrix[0]):
    task_matrix[49,i] = np.sum(np.diag(scope_matrix[32]) @ Spji, axis = 0)[i]
    task_matrix[50,i] = task_matrix[53,i] * task_list['Nominal Resource Ratio'].iloc[i]
    task_matrix[51,i] = task_matrix[49,i] + task_matrix[50,i]

""-----Realize Risks-----""

""-----Risks realized during execution-----""

if filenotes[-1] != 'O':
    for i in risks.index:
        if risks['State'][i] == 0:
            if i == 1 and task_matrix[0,11] == 1: #Risk 1: Customer Adds Requirements after task 11 is complete
                Location - Execute Tasks
                risks.loc[1,'State'] = 1
                if random.uniform(0,1) < risks['Likelihood'][1]:
                    for j in range(len(scope_matrix[30])):
                        project_vectors[j,1] = bool(project_vectors[j,1]) * (project_vectors[j,1] + 1)
                        scope_matrix[30,j] = scope_matrix[30,j] * 1.4
                        scope_matrix[0,j] = project_vectors[j,1]
            if i == 3 and task_matrix[0,9] == 1: #Risk 3: Late Undiscovered Rework - Location - Execute tasks
                risks.loc[3,'State'] = 1
                if random.uniform(0,1) < risks['Likelihood'][3]:
                    scope_matrix[8,3] = 0.25*scope_matrix[8,3]

""-----Impement Control Methods-----""

```

```

forecast = all_forecast_df[Ta-1].to_numpy()

MRL_List = ['MRL0','MRL1','MRL2','MRL3','MRL4','MRL5']
if controls['Control_method'] == 'Simple' or controls['Control_method'] == 'EVM' or controls['Control_method']
    in MRL_List:
    #Control Lag Operations
    #lists to store the time-delayed versions of the input variables for the control functions
    controls_lag = []
    scope_matrix_lag = []
    task_matrix_lag = []
    planned_teams_lag = []
    project_vectors_lag = []
    original_project_vectors_lag = []
    priorities_lag = []
    humans_lag = []
    control_accounts_lag = []
    constants_lag = []
    Ta_lag = []
    scope_to_task_matrix_lag = []
    margin_lag = []
    D_lag = []
    task_list_lag = []
    RL_plans_lag = []
    resources_lag = []

    #lists to store the time-delayed versions of the output variables from the control functions
    controls_imp_lag = []
    scope_matrix_imp_lag = []
    task_matrix_imp_lag = []
    planned_teams_imp_lag = []
    project_vectors_imp_lag = []
    humans_imp_lag = []
    forecast_imp_lag = []
    project_metrics_imp_lag = []

    if Ta % constants['Control_freq'] == 0:
        controls_lag.insert(0,controls.copy())
        scope_matrix_lag.insert(0,scope_matrix.copy())
        task_matrix_lag.insert(0,task_matrix.copy())
        planned_teams_lag.insert(0,planned_teams.copy())
        project_vectors_lag.insert(0,project_vectors.copy())
        original_project_vectors_lag.insert(0,original_project_vectors.copy())
        priorities_lag.insert(0,priorities.copy())
        humans_lag.insert(0,humans.copy())
        control_accounts_lag.insert(0,control_accounts.copy())
        constants_lag.insert(0,constants.copy())
        Ta_lag.insert(0,Ta)
        scope_to_task_matrix_lag.insert(0,scope_to_task_matrix.copy())
        margin_lag.insert(0,margin.copy())
        D_lag.insert(0,D.copy())
        task_list_lag.insert(0,task_list.copy())
        RL_plans_lag.insert(0,RL_plans.copy())
        resources_lag.insert(0,resources.copy())

```

```

if Ta >= constants['Measure_lag']:
    if Ta % constants['Control_freq'] == 0:
        a,b,c,d,e,f,g,h = implement_controls(controls_lag.pop(), scope_matrix_lag.pop(), task_matrix_lag.pop(),
            planned_teams_lag.pop(), project_vectors_lag.pop(), original_project_vectors_lag.pop(),
            priorities_lag.pop(), humans_lag.pop(), control_accounts_lag.pop(), constants_lag.pop(),
            Ta_lag.pop(), scope_to_task_matrix_lag.pop(), margin_lag.pop(), D_lag.pop(),
            task_list_lag.pop(), RL_plans_lag.pop(), resources_lag.pop(), Ca)
        scope_matrix_imp_lag.insert(0,a), task_matrix_imp_lag.insert(0,b), humans_imp_lag.insert(0,c),
            planned_teams_imp_lag.insert(0,d), forecast_imp_lag.insert(0,e),
            project_metrics_imp_lag.insert(0,f), controls_imp_lag.insert(0,g),
            project_vectors_imp_lag.insert(0,h)

    if Ta >= constants['Measure_lag'] + constants['Adaptations_lag']:
        if Ta % constants['Control_freq'] == 0:
            scope_matrix = scope_matrix_imp_lag.pop()
            task_matrix = task_matrix_imp_lag.pop()
            humans = humans_imp_lag.pop()
            planned_teams = planned_teams_imp_lag.pop()
            forecast = forecast_imp_lag.pop()
            project_metrics = project_metrics_imp_lag.pop()
            controls = controls_imp_lag.pop()
            project_vectors = project_vectors_imp_lag.pop()
            if controls['Control_method'] != 'Simple':
                all_project_metrics_df = pd.concat((all_project_metrics_df,
                    project_metrics.rename(index={'Project':Ta})), axis = 0)

#If control method is 'None', project is complete once planned scope quantities are reached
if controls['Control_method'] == 'None':
    forecast = project_vectors[:m+2,2]
    if (scope_matrix[5] >= scope_matrix[2]).all():
        closing_criteria['Scope_Complete'] = 1

if controls['Control_method'] == 0:
    forecast = project_vectors[:m+2,2]

forecast_df = pd.DataFrame(data = forecast, columns = [Ta], index = ['Scope Item 1', 'Scope Item 2', 'Scope Item
    3', 'Scope Item 4', 'Scope Item 5', 'Scope Item 6', 'Scope Item 7', 'Scope Item 8', 'Scope
    Item 9', 'Scope Item 10', 'Scope Item 11', 'Scope Item 12', 'Scope Item 13', 'Scope Item
    14', 'Project Cost', 'Project Duration'])

if (controls['cost_trip'] == 1 or controls['schedule_trip'] == 1) and controls['tripped'] == 0:
    controls['tripped_cost'] = calculate_cost(Ca, Ta, task_matrix, humans, constants)
    controls['tripped_schedule'] = Ta
    controls['tripped'] = 1

#Scope Complete Closing Criteria - completed quantity exceeds intermediate (and therefore also target)
    quantities, measured intermediate scope quality exceeds threshold, target scope quality
    exceeds threshold
if (scope_matrix[5] >= scope_matrix[1]).all() and (scope_matrix[9] >= scope_matrix[29]).all() and
    (scope_matrix[10] >= scope_matrix[30]).all() and task_matrix[62].all():
    closing_criteria['Scope_Complete'] = 1
#Cost threshold is exceeded

```

```

if calculate_cost(Ca, Ta, task_matrix, humans, constants) > project_vectors[m,0]:
    closing_criteria['Cost'] = 1
if Ta > project_vectors[m+1,0]:
    closing_criteria['Time'] = 1

all_forecast_df = pd.concat((all_forecast_df,forecast_df), axis = 1)
metrics_df = pd.concat((metrics_df,record_metrics(scope_matrix, task_matrix, resources, humans, Ta, Ca)[0]),
                        axis = 1)
scope_metrics_df = pd.concat((scope_metrics_df,record_metrics(scope_matrix, task_matrix, resources,
                                                                humans, Ta, Ca)[1]), axis = 1)

if filenotes == 'M-Test':
    now = datetime.now()
    filename1 = Path('Scores/forecast'+str(now.month)+str(now.day)+str(now.hour)+str(now.minute)+'.csv')
    filename2 = Path('Scores/metrics'+str(now.month)+str(now.day)+str(now.hour)+str(now.minute)+'.csv')
    filename3 = Path('Scores/scope_metrics'+str(now.month)+str(now.day)+str(now.hour)+str(now.minute)+'.csv')

else:
    filename1 = Path('Scores/forecast_'+filenotes+'.csv')
    filename2 = Path('Scores/task_metrics_'+filenotes+'.csv')
    filename3 = Path('Scores/scope_metrics_'+filenotes+'.csv')
    filename4 = Path('Scores/control_metrics_'+filenotes+'.csv')

if record_details == 1:
    all_forecast_df.to_csv(filename1)
    metrics_df.to_csv(filename2)
    scope_metrics_df.to_csv(filename3)
    all_project_metrics_df.to_csv(filename4)

return scope_matrix, task_matrix, resources, humans, Ta, calculate_cost(Ca, Ta, task_matrix, humans, constants),
        closing_criteria, priorities, controls

def record_metrics(scope_matrix, task_matrix, resources, humans, Ta, Ca):
    """ Calculate the metrics for analysis following the model run. Returns a pandas dataframe column to be appended
        for each tick, then output as a .csv"""
    m = scope_matrix.shape[1]
    metrics = np.concatenate(((Ca), [scope_matrix[4,m-1]], [scope_matrix[8,m-1]],
                              [humans['Assigned']],task_matrix[53], [resources['Available']], task_matrix[0],
                              task_matrix[62], task_matrix[61]))
    scope_metrics = np.concatenate((scope_matrix[2], scope_matrix[4], scope_matrix[8]))
    metrics_df = pd.DataFrame(data = metrics, columns = [Ta], index = ['Project Cost', 'System Scope Quantity',
                                                                    'System Scope Quality', 'Total Humans Assigned','Humans Assigned 1','Humans Assigned
                                                                    2','Humans Assigned 3','Humans Assigned 4','Humans Assigned 5','Humans Assigned
                                                                    6','Humans Assigned 7','Humans Assigned 8','Humans Assigned 9', 'Humans Assigned
                                                                    10','Humans Assigned 11','Humans Assigned 12','Humans Assigned 13','Humans Assigned
                                                                    14','Humans Assigned 15','Humans Assigned 16', 'Humans Assigned 17','Humans
                                                                    Assigned 18', 'Humans Assigned 19', 'Humans Assigned 20','Humans Assigned
                                                                    21','Humans Assigned 22', 'Resources Available', 'Task 1 Active', 'Task 2 Active', 'Task 3
                                                                    Active', 'Task 4 Active', 'Task 5 Active', 'Task 6 Active', 'Task 7 Active', 'Task 8 Active', 'Task
                                                                    9 Active', 'Task 10 Active', 'Task 11 Active', 'Task 12 Active', 'Task 13 Active', 'Task 14
                                                                    Active', 'Task 15 Active', 'Task 16 Active', 'Task 17 Active', 'Task 18 Active', 'Task 19
                                                                    Active', 'Task 20 Active', 'Task 21 Active', 'Task 22 Active', 'Task 1 Complete', 'Task 2
                                                                    Complete','Task 3 Complete','Task 4 Complete','Task 5 Complete','Task 6 Complete','Task

```

```

7 Complete','Task 8 Complete','Task 9 Complete','Task 10 Complete','Task 11
Complete','Task 12 Complete','Task 13 Complete','Task 14 Complete','Task 15
Complete','Task 16 Complete','Task 17 Complete','Task 18 Complete','Task 19
Complete','Task 20 Complete','Task 21 Complete','Task 22 Complete', 'Actual Task Cost
1','Actual Task Cost 2','Actual Task Cost 3','Actual Task Cost 4','Actual Task Cost 5','Actual
Task Cost 6','Actual Task Cost 7','Actual Task Cost 8','Actual Task Cost 9','Actual Task Cost
10','Actual Task Cost 11','Actual Task Cost 12','Actual Task Cost 13','Actual Task Cost
14','Actual Task Cost 15','Actual Task Cost 16','Actual Task Cost 17','Actual Task Cost
18','Actual Task Cost 19','Actual Task Cost 20','Actual Task Cost 21','Actual Task Cost 22'))

index_list = []
for i in range(m):
    index_list.append('Scope Item '+str(i+1)+' Planned Quantity')
for i in range(m):
    index_list.append('Scope Item '+str(i+1)+' Actual Quantity Completed')
for i in range(m):
    index_list.append('Scope Item '+str(i+1)+' Actual Quality of Completed Target Scope')
scope_metrics_df = pd.DataFrame(data = scope_metrics, columns = [Ta], index = index_list)
return metrics_df, scope_metrics_df

def calculate_cost(Ca, Ta, task_matrix, humans, constants):
    """Calculates project total cost at the point in time when run
    Project cost is cost of resources purchased plus cost of labor plus fixed overhead
    Input: Ca, Ta, task_matrix, humans
    Returns: Ca - should not be stored as Ca unless project execution is finished"""
    return Ca + task_matrix[6].sum() + Ta * 8 * constants['FixedOHR'] * humans['Hired'] #Project cost is cost of
        resources purchased plus cost of labor plus fixed overhead

def closing(scope_matrix, closing_criteria, Ta, Ca, project_vectors, customer_priorities, original_project_vectors):
    """Determines whether product is delivered and calculates project scores"""
    m = scope_matrix.shape[1]
    Sa = -1 * np.ones(m)
    original_target_quantities = original_project_vectors[:m,1]
    original_target_quality = np.ones(m) * original_project_vectors[m+2,1]
    if closing_criteria['Cost'] or closing_criteria['Time']:
        if (scope_matrix[4] >= original_project_vectors[:m,0]).all() and (scope_matrix[8] >=
            (np.ones(m)*original_project_vectors[m+2,0])).all(): #Quantity of scope completed is
                greater than cancellation threshold and actual quality of completed target scope is
                greater than cancellation quality threshold
            closing_criteria['Scope_Threshold'] = 1
        else:
            print('Product does not meet minimum requirements and will not be delivered')
    if closing_criteria['Scope_Complete'] == 1:
        closing_criteria['Scope_Threshold'] = 1
    if closing_criteria['Scope_Threshold'] == 1: #Product is delivered
        met_quantity_targets = scope_matrix[4] >= original_target_quantities #scope quantities meet target
            requirements
        quality_measure = scope_matrix[8] - original_target_quality #positive if scope item quality exceeds target
            requirements, negative if deficient
        Sa = quality_measure

    #Calculate Project Composite Score
    cost_performance = (project_vectors[m,1] - Ca) / project_vectors[m,1] #0 is right on cost target, positive is under
        target, negative is over cost target

```

```

time_performance = (project_vectors[m+1,1] - Ta) / project_vectors[m+1,1] #0 is right on time target, positive
                        means project took less time than target, negative means project took longer than target

scope_score = (1/3) * quality_measure
cost_score = (1/3) * cost_performance
time_score = (1/3) * time_performance

project_score = (customer_priorities[0]*scope_score + customer_priorities[1]*cost_score +
                customer_priorities[2]*time_score) / sum(customer_priorities) #overall score is
                weighted average of individual scores based on project priorities set by customer
return Sa, cost_score, time_score, project_score

"""-----Control Functions-----"""

def implement_controls(controls, scope_matrix, task_matrix, planned_teams, project_vectors,
                      original_project_vectors, priorities, humans, control_accounts, constants, Ta,
                      scope_to_task_matrix, margin, D, task_list, RL_plans, resources, Ca):
    """Implements various control methods"""

    #Simple Control method
    if controls['Control_method'] == 'Simple':
        scope_matrix, task_matrix, humans, planned_teams, forecast, project_metrics, controls, project_vectors =
            simple_control(controls, scope_matrix, task_matrix, planned_teams, project_vectors,
                          original_project_vectors, priorities, humans, D, task_list, scope_to_task_matrix, Ca, Ta,
                          constants)

    #EVM Control method
    if controls['Control_method'] == 'EVM':
        scope_matrix, task_matrix, humans, planned_teams, forecast, project_metrics, controls, project_vectors =
            EVM(controls, scope_matrix, task_matrix, planned_teams, project_vectors,
              original_project_vectors, priorities, humans, control_accounts, constants, Ta,
              scope_to_task_matrix, margin, D, task_list, Ca)

    #MRL Control method
    MRL_List = ['MRL0', 'MRL1', 'MRL2', 'MRL3', 'MRL4', 'MRL5']
    if controls['Control_method'] in MRL_List:
        scope_matrix, task_matrix, humans, planned_teams, forecast, project_metrics, controls, project_vectors =
            MRL(controls, scope_matrix, task_matrix, planned_teams, project_vectors,
              original_project_vectors, priorities, humans, control_accounts, constants, Ta,
              scope_to_task_matrix, margin, D, task_list, RL_plans, resources, Ca)

    return scope_matrix, task_matrix, humans, planned_teams, forecast, project_metrics, controls, project_vectors

def basic_universal_control(scope_matrix, task_matrix, humans, resources, Ca):
    """basic control functions that should be performed no matter which method is employed:
    -- measure and give feedback on when scope items and tasks are complete
    -- ensure tasks have minimum necessary humans on team and resources to accomplish
    -- more to come...
    """

def priority_dimension(priorities, scope_index_value, cost_index_value, time_index_value):

```

"""Determines whether the performance on each project dimension of cost, schedule, and scope (technical performance) is in line with priorities. Used in Simple Control function.  
 Input: priorities dictionary, scalar values for scope, cost, and time performance indices  
 Output: scope\_cost\_compare, scope\_time\_compare, cost\_time\_compare which are greater than one if first dimension outperforming second, less than 1 if second dimension outperforming first"""

```
scope_cost_compare = (scope_index_value / cost_index_value) / priorities['scope_cost'] #Less than one if scope
                    is underperforming its priority ranking compare to cost
scope_time_compare = (scope_index_value / time_index_value) / priorities['scope_time'] #Less than one if scope
                    is underperforming its priority ranking compare to time
cost_time_compare = (cost_index_value / time_index_value) / priorities['cost_time'] #Less than one if cost is
                    underperforming its priority ranking compared to time
return scope_cost_compare, scope_time_compare, cost_time_compare
```

```
def control_measures(controls, scope_matrix, task_matrix, planned_teams, project_vectors,
                    original_project_vectors, priorities):
```

""" Changes project variables to respond to project priorities. Priorities can be adjusted by control method to "pull levers" in project - trading off along iron triangle

Input: controls, scope\_matrix, task\_matrix, planned\_teams, project\_vectors, priorities

Returns: scope\_matrix, task\_matrix, planned\_teams, project\_vectors, controls

"""

```
m = scope_matrix.shape[1]
```

```
n = task_matrix.shape[1]
```

#Adjust cancellation thresholds to emphasize the most important project dimension

if priorities['scope'] > max(priorities['cost'], priorities['time']): #Scope is more important than cost and time

#Increase cost and time thresholds in accordance with relative priorities

```
project_vectors[m,0] = (priorities['scope']/priorities['cost']) * project_vectors[m,0] #Increase cost threshold
```

```
project_vectors[m+1,0] = (priorities['scope']/priorities['time']) * project_vectors[m+1,0] #Increase time
threshold
```

```
project_vectors[m+2,0] = (priorities['scope']/max(priorities['cost'], priorities['time'])) * project_vectors[m+2,0]
#Increase scope cancellation thresholds
```

```
project_vectors[:m,0] = ((priorities['scope']/max(priorities['cost'],
priorities['time']))*project_vectors[:m,0]).astype(int) #Increase scope cancellation
thresholds
```

```
project_vectors[:m,1] = list(original_project_vectors[:m,1]) #restore original project targets
```

if priorities['cost'] > max(priorities['scope'], priorities['time']): #If cost is more important than scope and time

```
project_vectors[m+2,0] = (priorities['scope']/priorities['cost']) * project_vectors[m+2,0] #Decrease scope
cancellation thresholds
```

```
project_vectors[:m,0] = ((priorities['scope']/priorities['cost'])*project_vectors[:m,0]).astype(int) #Decrease
scope cancellation thresholds
```

```
project_vectors[m+1,0] = (priorities['cost']/priorities['time']) * project_vectors[m+1,0] #Increase time threshold
```

```
project_vectors[m,0] = (max(priorities['scope'], priorities['time']/priorities['cost']) * project_vectors[m,0]
#Decrease cost threshold
```

if priorities['time'] > max(priorities['scope'], priorities['cost']): #If time is more important than scope and cost

```
project_vectors[m+2,0] = (priorities['scope']/priorities['time']) * project_vectors[m+2,0] #Decrease scope
cancellation thresholds
```

```
project_vectors[:m,0] = ((priorities['scope']/priorities['time'])*project_vectors[:m,0]).astype(int) #Decrease
scope cancellation thresholds
```

```
project_vectors[m+1,0] = (priorities['time']/priorities['cost']) * project_vectors[m,0] #Increase cost threshold
```

```
project_vectors[m,0] = (max(priorities['scope'], priorities['cost']/priorities['time']) * project_vectors[m+1,0]
#Decrease time threshold
```



```

#If time is more important than cost, increase team sizes, resources, and use overtime - and vice versa
if priorities['time'] > priorities['cost']: #Time is more important than cost
    controls['Team_modifier'] = priorities['time'] / priorities['cost']
    controls['Overtime'] = min(4, 2 * priorities['time'] / priorities['cost'])
    controls['Resource_modifier'] = priorities['time'] / priorities['cost']
if priorities['cost'] > priorities['time']: #Cost is more important than time
    controls['Team_modifier'] = priorities['time'] / priorities['cost']
    controls['Overtime'] = 0
    controls['Resource_modifier'] = priorities['time'] / priorities['cost']

#If time and cost are more important than scope, cut scope targets ** should be plan **
if min(priorities['cost'], priorities['time']) > priorities['scope']: #if the lesser of time and cost priorities is greater
    than scope priority
    project_vectors[:m,2] = ((priorities['scope']/min(priorities['cost'],
        priorities['time']))*project_vectors[:m,2]).astype(int) #target scope quantity
    project_vectors[m+2,1] = ((priorities['scope']/min(priorities['cost'], priorities['time']))*project_vectors[m+2,1])
        #target scope quality
    project_vectors[m+2,2] = ((priorities['scope']/min(priorities['cost'], priorities['time']))*project_vectors[m+2,2])
        #intermediate scope quality

for i in range(n):
    if not task_matrix[62,i]:
        task_matrix[57,i] += controls['Overtime'] #make changes to workday based on overtime
        planned_teams[i] = int(planned_teams[i] * controls['Team_modifier']) #make changes to planned team size
#Apply new thresholds and targets to scope vectors
scope_matrix[2] = project_vectors[:m,2] #target scope quantities
scope_matrix[4] = project_vectors[:m,0] #cancellation threshold scope quantities
scope_matrix[29] = project_vectors[m+2,2]*project_vectors[:m,2].astype(bool) #intermediate scope quality
        standard
scope_matrix[30] = project_vectors[m+2,1]*project_vectors[:m,1].astype(bool) #target scope quality standard
scope_matrix[31] = project_vectors[m+2,0]*project_vectors[:m,0].astype(bool) #cancellation threshold scope
        quality
return scope_matrix, task_matrix, planned_teams, project_vectors, controls

def simple_control(controls, scope_matrix, task_matrix, planned_teams, project_vectors, original_project_vectors,
    priorities, humans, D, task_list, scope_to_task_matrix, Ca, Ta, constants):
    """ Implements project adjustments based on task-level measures -- need to remove changes to the cancellation
        and intermediate thresholds, those are set at project initiation and should only be
        changed through negotiation with the customer
    Input: controls, scope_matrix, task_matrix, planned_teams, project_vectors, priorities, humans
    Returns: scope_matrix, task_matrix, humans, planned_teams, forecast #forecast is 1xm+2 (structured like
        project_vectors without the last (quality) entry)
    """
    m = scope_matrix.shape[1]
    n = task_matrix.shape[1]

    scope_reduced = np.zeros(n)
    cost_reduced = np.zeros(n)
    time_reduced = np.zeros(n)
    project_metrics = []

    task_to_scope_matrix = np.diag(np.power(task_matrix[1:15].sum(0),-1)) @ task_matrix[1:15].T

```

```

Smji = np.diagflat(scope_matrix[5]) @ scope_to_task_matrix #Quantity of measured scope complete per scope
item apportioned to each task
Spji = task_matrix[1:15].copy() #Planned units of scope item per task
qrmj = scope_matrix[9].copy() @ task_to_scope_matrix.T #measured quality of intermediate completed scope
for task i 1xn
qrxj = scope_matrix[29].copy() @ task_to_scope_matrix.T #threshold for intermeadiate scope quality for task i
1xn
task_percent_complete = np.minimum(np.ones(n),np.nan_to_num((np.diagflat(scope_matrix[12]) @
np.diagflat(1+scope_matrix[14]) @ Smji).sum(0) / (np.diagflat(scope_matrix[12]) @
np.diagflat(1 + scope_matrix[14]) @ Spji).sum(0))) #Task percent complete calculated as:
Amount of each task-assigned scope item measured complete times its nominal effort
divided by each task-assigned planned scope item times its nominal effort. Ensures that
scope items with much greater nominal effort are weighted to account for a greater
share of the work for a task. Cannot be greater than 1
scope_percent_complete = np.minimum(np.ones(m),np.nan_to_num((scope_matrix[12] * (1+scope_matrix[14]) *
scope_matrix[5]) / (scope_matrix[12] * (1 + scope_matrix[14]) * scope_matrix[2])))

task_cost_variance = task_matrix[15] * task_percent_complete - task_matrix[61] #Amount over/under budget
(positive is under budget)
task_schedule_variance = task_matrix[46] * task_percent_complete - task_matrix[58] #Amount ahead/behind
schedule in days (positive is ahead of schedule)
task_scope_variance = Smji.sum(0) * qrmj - Spji.sum(0) * qrxj * task_percent_complete #Amount ahead/behind
planned scope amount (in units of scope item) (positive is exceeding scope planned
amount)

cost_index = np.nan_to_num(task_matrix[15] * task_percent_complete / task_matrix[61]) # Percent of plan that
should be complete based on task progress divided by measured current status. 1 is right
on track, greater than 1 is ahead of plan (under budget, ahead of schedule, more scope
than planned)
cost_index = np.where(cost_index == 0, 0.1, cost_index) #eliminate divide by zero condition by replacing 0 cost
index with 0.1
schedule_index = np.nan_to_num(task_matrix[46] * task_percent_complete / task_matrix[58]) # Percent of plan
that should be complete based on task progress divided by measured current status. 1 is
right on track, greater than 1 is ahead of plan (under budget, ahead of schedule, more
scope than planned)
schedule_index = np.where(schedule_index == 0, 0.1, schedule_index)
scope_index = np.nan_to_num(Smji.sum(0) * qrmj / (Spji.sum(0) * qrxj * task_percent_complete)) # Measured
current status divided by percent of plan that should be complete. 1 is right on track,
greater than 1 is ahead of plan (under budget, ahead of schedule, more scope than
planned)
scope_index2 = np.nan_to_num(scope_matrix[5] * scope_matrix[9] / (scope_matrix[2] * scope_matrix[29] *
scope_percent_complete))

#Implement task-level adaptations
#Adjust cancellation thresholds to emphasize the most important project dimension
for i in range(n):
    if not task_matrix[61,i]: #If task hasn't incurred any cost, cost index is 1
        cost_index[i] = 1
    if not task_matrix[58,i]: #If task hasn't begun (no days elapsed), schedule index is 1
        schedule_index[i] = 1
    if not Smji.sum(0)[i]: #If task doesn't have any measured scope, scope index is 1
        scope_index[i] = 1

```

```

priority_compare = priority_dimension(priorities, scope_index[i], cost_index[i], schedule_index[i]) #returns a
    tuple of the prioritized performance comparisons for (scope/cost, scope/time, cost/time)
    where greater than one means numerator is outperforming denominator
#print('Priority Compare is:', priority_compare) #debug
if task_matrix[0,i]: #if the task is active
    if cost_index[i] < 1: #If cost is behind plan
        if priority_compare[0] > 1 and not scope_reduced[i]: #If cost is underperforming scope according to
            priorities, relax requirements (target, intermediate, and cancellation thresholds)
            scope_matrix[30] += -1 * min(0.05, (1-(1/priority_compare[0]))) * (task_matrix[1:15,i].astype(bool) *
                scope_matrix[30]) #Reduce the scope quality target threshold by the lesser of 5% or the
                scope/cost prioritized performance comparison
            scope_matrix[29] += -1 * min(0.05, (1-(1/priority_compare[0]))) * (task_matrix[1:15,i].astype(bool) *
                scope_matrix[29]) #Reduce the scope quality intermediate threshold by the lesser of 5%
                or the scope/cost prioritized performance comparison
            scope_reduced[i] = 1
    if priority_compare[2] < 1 and not time_reduced[i]: #If cost is underperforming time according to
        priorities, cut overtime and team size
        if task_matrix[57,i] == 8:
            planned_teams[i] = max(1,planned_teams[i]-1) #If no overtime is being used, reduce team size
            if task_matrix[0,i]: #if a task has already started, changed the actual assigned humans on the task
                task_matrix[53,i] = max(1,planned_teams[i]) #Minimum of one person working on an active task
                humans = personnel_action(0,-1,humans)[0]
            task_matrix[57,i] = max(8, task_matrix[57,i] - 1) #If overtime is being used, cut it
            time_reduced[i] = 1
    if schedule_index[i] < 1: #If project is behind schedule
        if priority_compare[1] > 1 and not scope_reduced[i]: #Time is underperforming scope on a prioritized basis
            scope_matrix[30] += -1 * min(0.05, (1-(1/priority_compare[0]))) * (task_matrix[1:15,i].astype(bool) *
                scope_matrix[30]) #Reduce the scope quality target threshold by the lesser of 5% or the
                scope/cost prioritized performance comparison
            scope_matrix[29] += -1 * min(0.05, (1-(1/priority_compare[0]))) * (task_matrix[1:15,i].astype(bool) *
                scope_matrix[29]) #Reduce the scope quality intermediate threshold by the lesser of 5%
                or the scope/cost prioritized performance comparison
            scope_reduced[i] = 1
        if priority_compare[2] > 1 and not cost_reduced[i]: #Time is underperforming cost on a prioritized basis
            if task_matrix[57,i] == 12:
                humans,h = personnel_action(0,1,humans)
                planned_teams[i] = planned_teams[i]+h #If overtime is maxed out, add to team
                if task_matrix[0,i]: #if a task has already started, changed the actual assigned humans on the task,
                    minimum of one person working
                    task_matrix[53,i] = max(1,planned_teams[i])
                task_matrix[57,i] = min(12, task_matrix[57,i] + 1) #Increase overtime up to 12 hour workday
                cost_reduced[i] = 1
    if scope_index[i] < 1: #Project execution will ensure scope targets are met until cancellation thresholds are
        met. Only action needed is to restore any quality thresholds that were previously
        reduced
    if priority_compare[0] < 1: #Scope is underperforming cost
        scope_matrix[30] = np.minimum(project_vectors[m+2,1],scope_matrix[30] + min(0.05, (1-
            priority_compare[0])) * (task_matrix[1:15,i].astype(bool) * scope_matrix[30]))
        scope_matrix[29] = np.minimum(project_vectors[m+2,2],scope_matrix[29] + min(0.05, (1-
            priority_compare[0])) * (task_matrix[1:15,i].astype(bool) * scope_matrix[29]))
    if priority_compare[1] < 1 and not time_reduced[i]: #Scope is underperforming time
        scope_matrix[30] = np.minimum(project_vectors[m+2,1],scope_matrix[30] + min(0.05, (1-
            priority_compare[0])) * (task_matrix[1:15,i].astype(bool) * scope_matrix[30]))

```

```

scope_matrix[29] = np.minimum(project_vectors[m+2,2],scope_matrix[29] + min(0.05, (1-
    priority_compare[0])) * (task_matrix[1:15,i].astype(bool) * scope_matrix[29]))

if (schedule_index < 0.2).any():
    schedule_index[np.where(schedule_index < 0.2)] = 0.2
new_task_duration_estimate = np.nan_to_num(task_matrix[58] + task_matrix[46] * (1-task_percent_complete) /
    schedule_index)
new_task_matrix = task_matrix.copy()
new_task_matrix[46] = new_task_duration_estimate.copy()
new_project_duration_estimate = update_schedule(new_task_matrix, D)[0]

#Create task-level forecasts
cost_forecast = np.nan_to_num(task_matrix[61] + task_matrix[15] * (1-task_percent_complete) / cost_index) +
    Ca + Ta * 8 * constants['FixedOHR'] * humans['Hired'] #Current task-level actual
    expenditure plus the remaining budgeted amount for the task divided by the rate at
    which the task is under/overrunning the budget
schedule_forecast = new_project_duration_estimate
scope_forecast = np.nan_to_num(scope_matrix[5] + scope_matrix[2] * (1-scope_percent_complete) /
    scope_index2) #Quantity of scope expected to be complete
forecast = np.concatenate((scope_forecast, [cost_forecast.sum()], [schedule_forecast])) #should add quality of
    completed system scope item (14)

if Smji.sum() / Spji.sum() > 0.25:
    if cost_forecast.sum() > (1 + max(0,(0.5-(Smji.sum() / Spji.sum())/2))) * original_project_vectors[m,0]:
        controls['cost_trip'] = 1
    if schedule_forecast > (1 + max(0,(0.5-(Smji.sum() / Spji.sum())/2))) * original_project_vectors[m+1,0]:
        controls['schedule_trip'] = 1

return scope_matrix, task_matrix, humans, planned_teams, forecast, project_metrics, controls, project_vectors

def EVM(controls, scope_matrix, task_matrix, planned_teams, project_vectors, original_project_vectors, priorities,
    humans, control_accounts, constants, Ta, scope_to_task_matrix, margin, D, task_list, Ca):
    """ Implements project adjustments based on EVM methods
    Input: controls, scope_matrix, task_matrix, planned_teams, project_vectors, priorities, humans
    Returns: scope_matrix, task_matrix, humans, planned_teams, forecast #forecast is 1xm+2 (structured like
        project_vectors without the last (quality) entry)
    """
    m = scope_matrix.shape[1]
    n = task_matrix.shape[1]

    project_budget = project_vectors[m,2] #Project budget includes MR - applied at project (global) level
    BAC = task_matrix[15].sum() #Budget at completion does not include MR, only CR (applied at control-account
        level and below)
    SAC = project_vectors[m+1,2] #Schedule at completion is total networked project schedule. Should include
        schedule risk margin, applied separately to project planned schedule
    PAR = BAC / SAC #dollars per day for project burn rate
    AT = Ta
    AC = task_matrix[61].sum() + Ca + Ta * 8 * constants['FixedOHR'] * humans['Hired']
    EV = BAC * min(1,max((task_matrix[59].sum() + task_matrix[60].sum()) / (task_matrix[16:30].sum() *
        (1+controls['Estimated_exception_rate'] + margin['Effort'])),
        (task_matrix[62]*task_matrix[16:30].sum(0)).sum() / task_matrix[16:30].sum()))
        #max(budgeted value times amount of actual effort complete divided by amount of
        planned effort (with risk included), or completed tasks times their nominal effort divided

```

by total nominal effort) -- review whether risk margin should be included (if risks don't materialize, earned value won't show as much progress as is actually made...)

```
CA_EVM_metrics = ('PV','EV','AC','CV','CPI','SV','SPI','ES','AT','TV','TPI','Float','PD','Budget') #PV - Planned Value, EV
- Earned Value, AC - Actual Cost, CV - Cost Variance, CPI - Cost Performance Index, SV -
Schedule Variance, SPI - Schedule Performance Index, ES - Earned Schedule, AT - Actual
Time, TV - Time Variance, TPI - Time Performance Index, Float - Network Schedule
Analysis Float (PDM), PD - Planned Duration (cumulative, no network analysis), Budget -
Control Account cumulative budget
df_empty = np.zeros((control_accounts.shape[1], len(CA_EVM_metrics)))
control_account_metrics = pd.DataFrame(data = df_empty, index = control_accounts.columns, columns =
CA_EVM_metrics) #EVM & ES metrics dataframe for control accounts
```

```
for i in control_accounts.columns:
```

```
    CA_PV = 0
```

```
    CA_EV = 0
```

```
    CA_AC = 0
```

```
    CA_ES = 0
```

```
    CA_AT = 0
```

```
    CA_PD = 0
```

```
    CA_float = 0
```

```
    CA_budget = 0
```

```
for j in control_accounts[i]: #traditional EVM metrics at the control account level - walk through each task in
each control account and compile EVM data
```

```
    if j != 0: #empty cells read as zeros
```

```
        j -= 1 #control accounts list tasks starting at 1, so much be decremented to match python schema
```

```
        CA_PV += task_matrix[15,j] * task_matrix[58,j] / task_matrix[46,j] #sum of task level planned value -
budgeted dollar amount times days task is active divided by total number of days task is
planned to take
```

```
        CA_EV += task_matrix[15,j] * min(1,max((task_matrix[59,j] + task_matrix[60,j]) /
(task_matrix[16:30,j].sum() * (1+task_matrix[63,j] + margin['Effort'])), task_matrix[62,j]))
# % actually complete as measured by work/coord done times budget
```

```
        CA_AC += task_matrix[61,j] #sum of actual task costs in control account
```

```
        CA_AT += task_matrix[58,j]
```

```
        CA_PD += task_matrix[46,j]
```

```
        CA_float += task_matrix[48,j]
```

```
        CA_budget += task_matrix[15,j]
```

```
control_account_metrics['PV'].loc[i] = CA_PV
```

```
control_account_metrics['EV'].loc[i] = CA_EV
```

```
control_account_metrics['AC'].loc[i] = CA_AC
```

```
control_account_metrics['AT'].loc[i] = CA_AT
```

```
control_account_metrics['CV'].loc[i] = CA_EV - CA_AC
```

```
control_account_metrics['Float'].loc[i] = CA_float
```

```
control_account_metrics['Budget'].loc[i] = CA_budget
```

```
if CA_AC != 0:
```

```
    control_account_metrics['CPI'].loc[i] = CA_EV / CA_AC
```

```
else:
```

```
    control_account_metrics['CPI'].loc[i] = max(CA_EV,1)
```

```
control_account_metrics['SV'].loc[i] = CA_EV - CA_PV
```

```
if CA_PV != 0:
```

```
    control_account_metrics['SPI'].loc[i] = CA_EV / CA_PV
```

```
else:
```

```
    control_account_metrics['SPI'].loc[i] = max(CA_EV,1)
```

```

#Calculating ES at the control account level requires using the control account level PMB and PAR. CA PMB is
the CA_PV. Project-level PAR is applied, but the implications should be assessed.
CA_ES = CA_PV / PAR
control_account_metrics['ES'].loc[i] = CA_ES
control_account_metrics['TV'].loc[i] = CA_ES - CA_AT
if CA_AT != 0:
    control_account_metrics['TPI'].loc[i] = CA_ES / CA_AT
else:
    control_account_metrics['TPI'].loc[i] = max(1,CA_ES)

#error check
# if control_account_metrics['AC'].sum() != AC:
#     print('Control account actual cost does not equal project actual cost', control_account_metrics['AC'].sum(),
#           task_matrix[61].sum(), AC)
# if control_account_metrics['EV'].sum() != EV:
#     print('Control account earned value does not equal project earned value',
#           control_account_metrics['EV'].sum(), EV)

#Implement control at control account and project level based on metrics. For scope, Quality is reduced at
control account level and quantity is reduced at project level (since quantity should be a
pretty major decision/infrequently changed)
if (control_account_metrics['CPI'].values < 0.9).any() or (control_account_metrics['TPI'].values < 0.9).any():
    for i in control_account_metrics.index:
        #returns a tuple of the prioritized performance comparisons for ( 0 - scope/cost, 1 - scope/time, 2 -
        cost/time) where greater than one means numerator is outperforming denominator
        CA_priority_compare = priority_dimension(priorities, 1, control_account_metrics['CPI'].loc[i],
        control_account_metrics['TPI'].loc[i])
        if control_account_metrics['CPI'].loc[i] < 0.9 and (control_account_metrics['EV'].loc[i] /
        control_account_metrics['Budget'].loc[i]) > 0.1: #control account progress is at least 10%
        - early indicators are unreliable
            #address cost overrun - Cut overtime (cost pri > time pri), reduce team sizes to 10 or fewer (cost pri > time
            pri), reduce scope thresholds (cost pri > scope pri)
            if CA_priority_compare[0] > 1: #if scope is outperforming cost, reduce scope quality thresholds. otherwise
            no action (b/c scope is doing even worse)
                for j in control_accounts[i]: #make changes to each task in the underperforming control account
                    if j != 0:
                        j -= 1
                        for k in np.where(scope_to_task_matrix[:,j].astype(bool))[0]: #for each scope item assigned to task
                        from control account
                            scope_matrix[29,k] = max(0.99 * scope_matrix[29,k], scope_matrix[31,k] / 2) #reduce the
                            intermediate quality threshold by 1%
                            scope_matrix[30,k] -= 0.1 * (scope_matrix[30,k] - scope_matrix[31,k]) #bring target scope quality
                            threshold closer to cancellation threshold
            if CA_priority_compare[2] < 1: #if cost is underperforming time, cut overtime and reduce team sizes.
            Otherwise do nothing - time is doing even worse.
                human_counter = task_matrix[53].copy()
                for j in control_accounts[i]: #for each task in the control account
                    if j != 0:
                        j -= 1
                        if task_matrix[0,j]: #if the task is active
                            task_matrix[53,j] = max(1,int(planned_teams[j]/4),min(10,task_matrix[53,j] - 1)) #team must be
                            at least 1, no larger than 10. Decrease size by one each time control is run.
                            task_matrix[57,j] = 8 #set workday to 8 hours (no overtime)

```

```

human_changes = (task_matrix[53] - human_counter).sum()
humans = personnel_action(0, human_changes, humans)[0]
if control_account_metrics['TPI'].loc[i] < 0.9 and ((1-control_account_metrics['TPI'].loc[i]) *
control_account_metrics['PD'].loc[i] >= control_account_metrics['Float'].loc[i]) and
(control_account_metrics['EV'].loc[i] / control_account_metrics['Budget'].loc[i]) > 0.1:
#TPI is below 90% and the control account is projected to eat up the total float; control
account progress is at least 10% - early indicators are unreliable
#address schedule overrun - Add resources and increase team size (time pri > cost pri), reduce scope
thresholds (time pri > scope pri), violate dependencies (time pri > scope pri)
if CA_priority_compare[1] > 1: #if scope is outperforming time, reduce scope quality thresholds.
Otherwise do nothing, scope is doing worse.
for j in control_accounts[i]:
if j != 0:
j -= 1
for k in np.where(scope_to_task_matrix[:,j].astype(bool))[0]: #for each scope item assigned to task
from control account
scope_matrix[29,k] = max(0.99 * scope_matrix[29,k], scope_matrix[31,k] / 2) #reduce the
intermediate quality threshold by 1%
scope_matrix[30,k] -= 0.1*(scope_matrix[30,k] - scope_matrix[31,k]) #bring target scope quality
threshold closer to cancellation threshold
if CA_priority_compare[2] > 1: #if cost is outperforming time, increase overtime and team size and
resources
for j in control_accounts[i]: #for each task j in control account i
if j != 0:
j -= 1
if task_matrix[0,j]: #if task is active
desired_increase = max(10-task_matrix[53,j], int((planned_teams[j] - task_matrix[53,j])/2))
#desired increase should bring to at least ten, or increase team by 1
humans,h = personnel_action(0,desired_increase,humans)
task_matrix[53,j] = task_matrix[53,j] + h #increase team by number of available humans up to
desired_increase
controls['Overtime'] = min(4, controls['Overtime']+1)
task_matrix[57,j] = min(12, task_matrix[57,j]+1) #increase overtime
controls['Resource_modifier'] = min(2,1.1*controls['Resource_modifier']) #increase amount of
resources being allocated

project_EVM_metrics = ['EV','PV','AC','CV','CPI','SV','SPI','EAC','TCPI','ETC','ES','TV','TPI','TEAC']
project_metrics = pd.DataFrame(data = np.zeros((1, len(project_EVM_metrics))), index = ['Project'], columns =
project_EVM_metrics) #EVM & ES metrics dataframe for project level. One row for
project-level values

PV = control_account_metrics['PV'].sum()
project_metrics['EV'] = EV
project_metrics['PV'] = PV
project_metrics['AC'] = AC
project_metrics['CV'] = EV - AC
if AC != 0:
project_metrics['CPI'] = EV / AC
else:
project_metrics['CPI'] = max(EV,1)
project_metrics['SV'] = EV - PV
if PV != 0:
project_metrics['SPI'] = EV / PV

```

```

else:
    project_metrics['SPI'] = max(EV,1)
CPI = project_metrics['CPI'].iloc[0]
SPI = project_metrics['SPI'].iloc[0]
project_metrics['EAC'] = AC + (BAC-EV) / (CPI * SPI)
EAC = project_metrics['EAC'].iloc[0]
project_metrics['TCPI'] = (BAC - EV) / (EAC-AC)
project_metrics['ETC'] = EAC - AC
project_metrics['ES'] = EV / PAR
ES = project_metrics['ES'].iloc[0]
project_metrics['TV'] = ES - AT
project_metrics['TPI'] = ES / AT
TPI = project_metrics['TPI'].iloc[0]
project_metrics['TEAC'] = SAC / TPI

#calculate critical path time metrics - essentially just treat critical path as a separate control account
critical_path_tasks = np.where(task_matrix[47]==1)[0] #np.where returns the positions of the values of an array
that meets the specified criteria. Returns a tuple where the first value [0] of the tuple is
the position values in a 1-d array
critical_path_EVM_metrics = ['PV','ES','TV','TPI','TEAC']
critical_path_metrics = pd.DataFrame(data = np.zeros((1,len(critical_path_EVM_metrics))), index = ['Critical
Path'], columns = critical_path_EVM_metrics)

CP_PV = 0
CP_ES = 0
CP_AT = 0
CP_PD = 0
CP_budget = 0
for j in critical_path_tasks:
    CP_PV += task_matrix[15,j] * task_matrix[58,j] / task_matrix[46,j]
    CP_AT += task_matrix[58,j]
    CP_PD += task_matrix[46,j]
    CP_budget += task_matrix[15,j]
CP_ES = CP_PV / PAR
critical_path_metrics['ES'] = CP_ES
critical_path_metrics['TV'] = CP_ES - CP_AT
if CP_AT != 0:
    critical_path_metrics['TPI'] = CP_ES / CP_AT
else:
    critical_path_metrics['TPI'] = max(1,CP_ES)
critical_path_metrics['TEAC'] = SAC / critical_path_metrics['TPI'].iloc[0]

#returns a tuple of the prioritized performance comparisons for (scope/cost, scope/time, cost/time) where
greater than one means numerator is outperforming denominator
priority_compare = priority_dimension(priorities, project_metrics['SPI'].iloc[0], project_metrics['CPI'].iloc[0],
project_metrics['TPI'].iloc[0]) #SPI used for scope measurement - indicator of whether
effort (EV measurement of scope) is ahead or behind plan
#Implement project-level control measures. Scope changes will be both quantity and quality, with focus on later
tasks. Schedule should focus on critical path (global view of project schedule)
if ((project_metrics['CPI'].loc['Project'] < 0.9) or (critical_path_metrics['TPI'].loc['Critical Path'] < 0.9)) and (EV /
BAC) > 0.1: #project is at least 10% complete - early indicators are unreliable
    if project_metrics['CPI'].loc['Project'] < 0.9:
        #address cost overruns - cut overtime, reduce team sizes, cut scope

```



```

if priority_compare[0] > 1.2: #if scope is outperforming cost, reduce scope quality thresholds. otherwise no
    action (b/c scope is doing even worse)
for i in np.where(task_matrix[62] == 0)[0]: #for incomplete tasks
    if EV / BAC > 0.2: #Project is at least 20% complete measured by effort --> based on Lukas 2008, CPI
        should be reliable indicator of performance at this point. Prudent to begin cutting scope
        targets
        for k in np.where(scope_to_task_matrix[:,i].astype(bool))[0]: #for each scope item assigned to the
            incomplete tasks
            scope_matrix[29,k] = max(0.99 * scope_matrix[29,k], scope_matrix[31,k] / 2) #reduce the
                intermediate quality threshold by 1% down to a limit of half the cancellation threshold
            scope_matrix[30,k] -= 0.1*(scope_matrix[30,k] - scope_matrix[31,k]) #bring target scope quality
                threshold closer to cancellation threshold
    if EV / BAC > 0.5: #Once project is halfway finished, if CPI hasn't recovered, time to start cutting scope
        quantities
        for k in np.where(scope_to_task_matrix[:,i].astype(bool))[0]: #for each scope item assigned to the
            incomplete tasks
            scope_matrix[1,k] = max(scope_matrix[1,k] - scope_matrix[0,k], scope_matrix[1,k]-1) #reduce
                intermediate scope required by one, stopping at the required excess quantity of
                intermediate scope over target scope that was specified at project origination. This must
                be done before recalculating target scope because it relies on the old value.
            scope_matrix[0,k] = max(scope_matrix[3,k], scope_matrix[0,k]-1) #reduce target quantity by one,
                stopping at cancellation threshold
            scope_matrix[2,k] = max(scope_matrix[1,k], scope_matrix[2,k] - 1) #reduce planned scope with
                lower limit of required intermediate scope. Must be calculated after required
                intermediate scope change
if priority_compare[2] < 0.8: #if cost is underperforming time, cut overtime and reduce team sizes.
    Otherwise do nothing - time is doing even worse.
task_matrix[57] = 8*np.ones((n)) #set all workdays to 8 hours
human_counter = task_matrix[53].copy()
for i in np.where(task_matrix[62] == 0)[0]: #for incomplete tasks not on the critical path
    if task_matrix[0,i] and task_matrix[47,i] == 0:
        task_matrix[53,i] = max(1,int(planned_teams[i]/4),min(10,task_matrix[53,i] - 1)) #team must be at
            least 1, no larger than 10. Decrease size by one each time control is run.
        task_matrix[57,i] = 8 #set workday to 8 hours (no overtime)
    human_changes = (task_matrix[53] - human_counter).sum()
    humans = personnel_action(0,human_changes,humans)[0]
if critical_path_metrics['TPI'].loc['Critical Path'] < 0.9: #TPI that matters is along the critical path - the rest is just
    fluff
#address schedule shortfall in those tasks on the critical path - add resources, increase team sizes, reduce
    scope thresholds, violate dependencies
if priority_compare[1] > 1.2: #if scope is outperforming time, reduce scope quality thresholds. Otherwise do
    nothing, scope is doing worse. Same actions as for cost-scope
for i in np.where(task_matrix[62] == 0)[0]: #for incomplete tasks on the critical path
    if EV / BAC > 0.2 and task_matrix[47,i] == 1: #Project is at least 20% complete measured by effort -->
        basing on philosophy for CPI, although TPI is not proven to be a reliable indicator - need
        to research
        for k in np.where(scope_to_task_matrix[:,i].astype(bool))[0]: #for each scope item assigned to the
            incomplete tasks
            scope_matrix[29,k] = max(0.99 * scope_matrix[29,k], scope_matrix[31,k] / 2) #reduce the
                intermediate quality threshold by 1%
            scope_matrix[30,k] -= 0.1*(scope_matrix[30,k] - scope_matrix[31,k]) #bring target scope quality
                threshold closer to cancellation threshold

```

```

if EV / BAC > 0.5: #Once project is halfway finished, if CPI hasn't recovered, time to start cutting scope
    quantities
    for k in np.where(scope_to_task_matrix[:,i].astype(bool))[0]: #for each scope item assigned to the
        incomplete tasks
        scope_matrix[1,k] = max(scope_matrix[1,k] - scope_matrix[0,k], scope_matrix[1,k]-1) #reduce
            intermediate scope required by one, stopping at the required excess quantity of
            intermediate scope over target scope that was specified at project origination. This must
            be done before recalculating target scope because it relies on the old value.
        scope_matrix[0,k] = max(scope_matrix[3,k], scope_matrix[0,k]-1) #reduce target quantity by one,
            stopping at cancellation threshold
        scope_matrix[2,k] = max(scope_matrix[1,k], scope_matrix[2,k] - 1) #reduce planned scope with
            lower limit of required intermediate scope. Must be calculated after required
            intermediate scope change
if priority_compare[2] > 1.2: #if cost is outperforming time, increase overtime and team size and resources
    for i in np.where(task_matrix[62] == 0)[0]:
        if task_matrix[0,i] and task_matrix[47,i] == 1: #if task is active
            desired_increase = max(10-task_matrix[53,i],int((planned_teams[i] - task_matrix[53,i])/2)) #desired
                increase should bring to at least ten, or increase team by 1
            humans,h = personnel_action(0,desired_increase,humans)
            task_matrix[53,i] = task_matrix[53,i] + h #increase team by number of available humans up to
                desired_increase
            controls['Overtime'] = min(4, controls['Overtime']+1)
            task_matrix[57,i] = min(12, task_matrix[57,i]+1) #increase overtime
            controls['Resource_modifier'] = min(2,1.1*controls['Resource_modifier']) #increase amount of
                resources being allocated

cost_forecast = project_metrics['EAC'].loc['Project']
schedule_forecast = project_metrics['TEAC'].loc['Project']
scope_forecast = scope_matrix[2]

forecast = np.concatenate((scope_forecast, [cost_forecast], [schedule_forecast]))

if EV / BAC > 0.25:
    # print('Forecast Threshold',(1 + max(0,(0.5-(EV / (2*BAC)))))) * original_project_vectors[m,0], 'cost forecast',
        cost_forecast)
    # print(cost_forecast > (1 + max(0,(0.5-(EV / (2*BAC)))))) * original_project_vectors[m,0])
    if cost_forecast > (1 + max(0,(0.5-(EV / (2*BAC)))))) * original_project_vectors[m,0]:
        controls['cost_trip'] = 1
    if schedule_forecast > (1 + max(0,(0.5-(ES / (2*SAC)))))) * original_project_vectors[m+1,0]:
        controls['schedule_trip'] = 1
    # print(controls['cost_trip'], controls['schedule_trip'])

# if forecast is drastically different from plan, replan - removing replan - adjustments to quality and quantity result
    in skewed replanning
if (abs((project_vectors[m,2] - cost_forecast)/project_vectors[m,2]) > controls['Replanning_threshold'] or
    abs((project_vectors[m+1,2] - schedule_forecast)/project_vectors[m+1,2]) >
    controls['Replanning_threshold']) and abs((Ta - controls['Last_replan']) / SAC) >
    controls['Replanning_frequency']:

    # print('replanning')
    scope_matrix[1,1] = scope_matrix[1,1] + 1 #assumes scope item 2 is project plan - adds an additional plan
        scope unit to target and intermediate scope

```

```

scope_matrix, task_matrix, project_vectors, planned_teams = create_plan(scope_matrix, task_matrix,
                                project_vectors, planned_teams, margin, constants, controls, scope_to_task_matrix, D,
                                task_list)
controls['Last_replan'] = Ta
task_matrix, project_vectors = EVM_risk(task_matrix, project_vectors, margin)

return scope_matrix, task_matrix, humans, planned_teams, forecast, project_metrics, controls, project_vectors

def EVM_risk(task_matrix, project_vectors, margin):
    """Budget updates to incorporate contingency reserve for task-level budgets
    Need to apply after every replanning cycle"""
    m = project_vectors.shape[0] - 3
    task_matrix[15] = task_matrix[15] * (1 + margin['Unknown']) # General contingency reserve for task-level risks
    task_matrix[15, 11:20] = task_matrix[15, 11:20] * 1.1 # Additional 10% Contingency reserve for late undiscovered
    rework in tasks 12-20
    project_vectors[m,2] = (1 + margin['Unknown']) * project_vectors[m,2] #Adds 20% management reserve to
    budget
    return task_matrix, project_vectors

def EVM_planning_margin(Tp, controls, margin):
    """Adds schedule risk margin for EVM control method - accounts for exception handling and risk
    Must be applied before scope quantities are calculated"""
    Tp = (1+controls['Estimated_exception_rate'] + margin['Effort']) * Tp.copy()
    return Tp

def MRL_exception_handling(Tp, controls):
    """Adds schedule exception handling margin for MRL control method
    Must be applied before scope quantities are calculated"""
    Tp = (1+controls['Estimated_exception_rate']) * Tp.copy()
    return Tp

def MRL_planning_margin(Tp, margin):
    """Adds schedule exception handling margin for MRL control method
    Must be applied before scope quantities are calculated"""
    Tp = (1+margin['Effort']) * Tp.copy()
    return Tp

#MRL_plans function needs to be run on a for loop over the MRL control method settings to return a plan for each
level
def MRL_plans(scope_matrix, task_matrix, project_vectors, planned_teams, margin, constants, controls,
              scope_to_task_matrix, D, task_list):
    m = scope_matrix.shape[1]
    n = task_matrix.shape[1]

    #work with copies so as not to change the original vectors in the project when creating different plans
    scope_matrix_copy = scope_matrix.copy()
    task_matrix_copy = task_matrix.copy()
    project_vectors_copy = project_vectors.copy()
    planned_teams_copy = planned_teams.copy()
    margin_copy = margin.copy()
    constants_copy = constants.copy()
    controls_copy = controls.copy()
    scope_to_task_matrix_copy = scope_to_task_matrix.copy()

```

```

D_copy = D.copy()
task_list_copy = task_list.copy()

# Risk Level 0: Best-case base plan, add non-nominal exception handling
if controls_copy['Control_method'] == 'MRLO':
    #exception handling function is run in create_plan based on Control Method = MRLO
    scope_matrix_0, task_matrix_0, project_vectors_0, planned_teams_0 = create_plan(scope_matrix_copy,
                                          task_matrix_copy, project_vectors_copy, planned_teams_copy, margin_copy,
                                          constants_copy, controls_copy, scope_to_task_matrix_copy, D_copy, task_list_copy)
    RL0 = (scope_matrix_0, task_matrix_0, project_vectors_0, planned_teams_0)
# Risk Level 1: Risks due to environment
if controls_copy['Control_method'] == 'MRL1':
    #runs exception handling (MRLO) and schedule risk (planning, MRL1) margin functions in create plan
    #below includes budget risk (unknown) margin and resource margin - must be carried forward in successive
    MRLs
    task_matrix_copy[15] = task_matrix[15] * (1 + margin['Unknown']) * random_array(n,np.ones(n),np.ones(n) *
                                          constants['Model_measurement_accuracy'])
    task_matrix_copy[51] = task_matrix[51] * (1 + margin['Resource']) * random_array(n,np.ones(n),np.ones(n) *
                                          constants['Model_measurement_accuracy'])
    scope_matrix_1, task_matrix_1, project_vectors_1, planned_teams_1 = create_plan(scope_matrix_copy,
                                          task_matrix_copy, project_vectors_copy, planned_teams_copy, margin_copy,
                                          constants_copy, controls_copy, scope_to_task_matrix_copy, D_copy, task_list_copy)
    RL1 = (scope_matrix_1, task_matrix_1, project_vectors_1, planned_teams_1)
# Risk Level 2: RL 1 + Late Undiscovered Rework Risk (Risk ID 3)
if controls_copy['Control_method'] == 'MRL2':
    #runs exception handling (MRLO) and schedule risk (planning, MRL1) margin functions in create plan
    #budget and resource margins from MRL1
    task_matrix_copy[15] = task_matrix[15] * (1 + margin['Unknown']) * random_array(n,np.ones(n),np.ones(n) *
                                          constants['Model_measurement_accuracy'])
    task_matrix_copy[51] = task_matrix[51] * (1 + margin['Resource']) * random_array(n,np.ones(n),np.ones(n) *
                                          constants['Model_measurement_accuracy'])
    #Increase intermediate scope quantities and qualities to account for undiscovered rework accumulating in
    project >> instead improve measurements?
    scope_matrix_copy[1], scope_matrix_copy[29] = scope_matrix_copy[1] *
        (1+controls['Undiscovered_rework_rate']) * random_array(m,np.ones(m),np.ones(m) *
        constants['Model_measurement_accuracy']), scope_matrix_copy[29] *
        (1+controls['Undiscovered_rework_rate']) * random_array(m,np.ones(m),np.ones(m) *
        constants['Model_measurement_accuracy'])
    scope_matrix_2, task_matrix_2, project_vectors_2, planned_teams_2 = create_plan(scope_matrix_copy,
                                          task_matrix_copy, project_vectors_copy, planned_teams_copy, margin_copy,
                                          constants_copy, controls_copy, scope_to_task_matrix_copy, D_copy, task_list_copy)
    RL2 = (scope_matrix_2, task_matrix_2, project_vectors_2, planned_teams_2)
# Risk Level 3: RL 2 + Additional Requirements Risk (Risk ID 1)
if controls_copy['Control_method'] == 'MRL3':
    #runs exception handling (MRLO) and schedule risk (planning, MRL1) margin functions in create plan
    #budget and resource margins from MRL1
    task_matrix_copy[15] = task_matrix[15] * (1 + margin['Unknown']) * random_array(n,np.ones(n),np.ones(n) *
                                          constants['Model_measurement_accuracy'])
    task_matrix_copy[51] = task_matrix[51] * (1 + margin['Resource']) * random_array(n,np.ones(n),np.ones(n) *
                                          constants['Model_measurement_accuracy'])
    #Increase intermediate scope quantities and qualities to account for undiscovered rework accumulating in
    project >> instead improve measurements?

```

```

scope_matrix_copy[1], scope_matrix_copy[29] = scope_matrix_copy[1] *
    (1+controls['Undiscovered_rework_rate'] ) * random_array(m,np.ones(m),np.ones(m) *
    constants['Model_measurement_accuracy']), scope_matrix_copy[29] *
    (1+controls['Undiscovered_rework_rate']) * random_array(m,np.ones(m),np.ones(m) *
    constants['Model_measurement_accuracy'])
#Customer increases target quantity and quality (adds requirements) - new to MRL3
#save intermediate-target quantity margins to preserve following increase
intermediate_quantity_margin = scope_matrix_copy[1] - scope_matrix_copy[0]
scope_matrix_copy[0], scope_matrix_copy[30] = (1.25 * scope_matrix_copy[0]*
    random_array(m,np.ones(m),np.ones(m) * constants['Model_measurement_accuracy']) +
    1).astype(int) , 1.25 * scope_matrix_copy[30] * random_array(m,np.ones(m),np.ones(m)
    * constants['Model_measurement_accuracy']) #anticipates a 25% increase in quantity
    (rounded up to whole units) and 25% increase in quality (requirements)
scope_matrix_copy[1], scope_matrix_copy[29] = scope_matrix_copy[0] + intermediate_quantity_margin, 1.25
    * scope_matrix_copy[29] * random_array(m,np.ones(m),np.ones(m) *
    constants['Model_measurement_accuracy']) #increases intermediate scope quantities
    and qualities to keep pace with target increases
scope_matrix_3, task_matrix_3, project_vectors_3, planned_teams_3 = create_plan(scope_matrix_copy,
    task_matrix_copy, project_vectors_copy, planned_teams_copy, margin_copy,
    constants_copy, controls_copy, scope_to_task_matrix_copy, D_copy, task_list_copy)
RL3 = (scope_matrix_3, task_matrix_3, project_vectors_3, planned_teams_3)
# Risk Level 4: RL 3 + Defective Parts (Risk ID 4)
if controls_copy['Control_method'] == 'MRL4':
    #runs exception handling (MRL0) and schedule risk (planning, MRL1) margin functions in create plan
    #budget and resource margins from MRL1
    task_matrix_copy[15] = task_matrix[15] * (1 + margin['Unknown']) * random_array(n,np.ones(n),np.ones(n) *
    constants['Model_measurement_accuracy'])
    task_matrix_copy[51] = task_matrix[51] * (1 + margin['Resource']) * random_array(n,np.ones(n),np.ones(n) *
    constants['Model_measurement_accuracy'])
    #Increase intermediate scope quantities and qualities to account for undiscovered rework accumulating in
    project >> instead improve measurements?
    scope_matrix_copy[1], scope_matrix_copy[29] = scope_matrix_copy[1] *
    (1+controls['Undiscovered_rework_rate'] ) * random_array(m,np.ones(m),np.ones(m) *
    constants['Model_measurement_accuracy']), scope_matrix_copy[29] *
    (1+controls['Undiscovered_rework_rate']) * random_array(m,np.ones(m),np.ones(m) *
    constants['Model_measurement_accuracy'])
    #Customer increases target quantity and quality (adds requirements) - new to MRL3
    #save intermediate-target quantity margins to preserve following increase
    intermediate_quantity_margin = scope_matrix_copy[1] - scope_matrix_copy[0]
    scope_matrix_copy[0], scope_matrix_copy[30] = (1.25 * scope_matrix_copy[0]*
    random_array(m,np.ones(m),np.ones(m) * constants['Model_measurement_accuracy'])
    + 1).astype(int) , 1.25 * scope_matrix_copy[30] *
    random_array(m,np.ones(m),np.ones(m) * constants['Model_measurement_accuracy'])
    #anticipates a 25% increase in quantity (rounded up to whole units) and 25% increase in
    quality (requirements)
    scope_matrix_copy[1], scope_matrix_copy[29] = scope_matrix_copy[0] + intermediate_quantity_margin, 1.25
    * scope_matrix_copy[29] * random_array(m,np.ones(m),np.ones(m) *
    constants['Model_measurement_accuracy']) #increases intermediate scope quantities
    and qualities to keep pace with target increases
    #Increase resource margin to anticipate defective parts
    margin_copy['Resource'] = 0.2 * random.gauss(1,constants['Model_measurement_accuracy'])
    task_matrix_copy[51] = task_matrix[51] * (1 + margin_copy['Resource']) *
    random_array(n,np.ones(n),np.ones(n) * constants['Model_measurement_accuracy'])

```

```

scope_matrix_4, task_matrix_4, project_vectors_4, planned_teams_4 = create_plan(scope_matrix_copy,
    task_matrix_copy, project_vectors_copy, planned_teams_copy, margin_copy,
    constants_copy, controls_copy, scope_to_task_matrix_copy, D_copy, task_list_copy)
RL4 = (scope_matrix_4, task_matrix_4, project_vectors_4, planned_teams_4)
#Risk Level 5: RL 4 + Hiring Freeze (Risk ID 2)
if controls_copy['Control_method'] == 'MRL5':
    #runs exception handling (MRLO) and schedule risk (planning, MRL1) margin functions in create plan
    #budget and resource margins from MRL1
    task_matrix_copy[15] = task_matrix[15] * (1 + margin['Unknown']) * random_array(n,np.ones(n),np.ones(n) *
        constants['Model_measurement_accuracy'])
    task_matrix_copy[51] = task_matrix[51] * (1 + margin['Resource']) * random_array(n,np.ones(n),np.ones(n) *
        constants['Model_measurement_accuracy'])
    #Increase intermediate scope quantities and qualities to account for undiscovered rework accumulating in
    project >> instead improve measurements?
    scope_matrix_copy[1], scope_matrix_copy[29] = scope_matrix_copy[1] *
        (1+controls['Undiscovered_rework_rate'] ) * random_array(m,np.ones(m),np.ones(m) *
        constants['Model_measurement_accuracy']), scope_matrix_copy[29] *
        (1+controls['Undiscovered_rework_rate']) * random_array(m,np.ones(m),np.ones(m) *
        constants['Model_measurement_accuracy'])
    #Customer increases target quantity and quality (adds requirements) - new to MRL3
    #save intermediate-target quantity margins to preserve following increase
    intermediate_quantity_margin = scope_matrix_copy[1] - scope_matrix_copy[0]
    scope_matrix_copy[0], scope_matrix_copy[30] = (1.25 * scope_matrix_copy[0]*
        random_array(m,np.ones(m),np.ones(m) * constants['Model_measurement_accuracy'])
        + 1).astype(int) , 1.25 * scope_matrix_copy[30] *
        random_array(m,np.ones(m),np.ones(m) * constants['Model_measurement_accuracy'])
    #anticipates a 25% increase in quantity (rounded up to whole units) and 25% increase in
    quality (requirements)
    scope_matrix_copy[1], scope_matrix_copy[29] = scope_matrix_copy[0] + intermediate_quantity_margin, 1.25
        * scope_matrix_copy[29] * random_array(m,np.ones(m),np.ones(m) *
        constants['Model_measurement_accuracy']) #increases intermediate scope quantities
        and qualities to keep pace with target increases
    #Increase resource margin to anticipate defective parts
    margin_copy['Resource'] = 0.2 * random.gauss(1,constants['Model_measurement_accuracy'])
    task_matrix_copy[51] = task_matrix[51] * (1 + margin_copy['Resource']) *
        random_array(n,np.ones(n),np.ones(n) * constants['Model_measurement_accuracy'])
    #Anticipated hiring freeze and reduced human pool
    constants_copy['Initial_hire'] = 70 * random.gauss(1,constants['Model_measurement_accuracy'])
    task_matrix_copy[45] = np.ones(n) * 12
    scope_matrix_5, task_matrix_5, project_vectors_5, planned_teams_5 = create_plan(scope_matrix_copy,
        task_matrix_copy, project_vectors_copy, planned_teams_copy, margin_copy,
        constants_copy, controls_copy, scope_to_task_matrix_copy, D_copy, task_list_copy)
    RL5 = (scope_matrix_5, task_matrix_5, project_vectors_5, planned_teams_5)

if controls_copy['Control_method'] == 'MRLO':
    return RL0
elif controls_copy['Control_method'] == 'MRL1':
    return RL1
elif controls_copy['Control_method'] == 'MRL2':
    return RL2
elif controls_copy['Control_method'] == 'MRL3':
    return RL3
elif controls_copy['Control_method'] == 'MRL4':

```

```

    return RL4
elif controls_copy['Control_method'] == 'MRL5':
    return RL5
else:
    return None

def MRL(controls, scope_matrix, task_matrix, planned_teams, project_vectors, original_project_vectors, priorities,
        humans, control_accounts, constants, Ta, scope_to_task_matrix, margin, D, task_list,
        RL_plans, resources, Ca):
    """Multiple Risk Level control method - modify from paper to include technical performance and make more
    forward looking"""

    # Adapt EV metrics to monitor technical performance and project schedule and budget based on product progress
    # Use MRL plans to calculate multiple PVs

    m = scope_matrix.shape[1]
    n = task_matrix.shape[1]

    RL0 = RL_plans[0] #contains: [0] scope_matrix, [1] task_matrix, [2] project_vectors, [3] planned_teams,
    RL1 = RL_plans[1] #Most Likely Case
    RL2 = RL_plans[2] # Late Undiscovered Work
    RL3 = RL_plans[3] # Customer Adds Requirements
    RL4 = RL_plans[4] # Defective Parts
    RL5 = RL_plans[5] # Hiring Freeze

    #measure for risks and swap plans if needed
    #start with least likely and move down so only one plan is implemented
    risk_realized = 0
    #MRL 5: Hiring Freeze
    if (project_vectors[0:m,1] > original_project_vectors[0:m,1]).any() and resources['Purchased'] < 0.9 *
        task_matrix[51].sum() and humans['Freeze'] == 1:
        scope_matrix[1], scope_matrix[2], scope_matrix[29] = RL5[0][1], RL5[0][2], RL5[0][29] #replace planned and
            intermediate scope quantities and intermediate quality with preplanned values
        task_matrix[1:16], task_matrix[30:52] = RL5[1][1:16], RL5[1][30:52] #Update task plan
        project_vectors[:,2] = RL5[2][:,2]
        planned_teams = RL5[3]
        controls['Control_method'] = 'MRL5'
        risk_realized = 1

    #MRL 4: Defective Parts
    elif (project_vectors[0:m,1] > original_project_vectors[0:m,1]).any() and resources['Purchased'] < 0.9 *
        task_matrix[51].sum():
        scope_matrix[1], scope_matrix[2], scope_matrix[29] = RL4[0][1], RL4[0][2], RL4[0][29] #replace planned and
            intermediate scope quantities and intermediate quality with preplanned values
        task_matrix[1:16], task_matrix[30:52] = RL4[1][1:16], RL4[1][30:52] #Update task plan
        project_vectors[:,2] = RL4[2][:,2]
        planned_teams = RL4[3]
        controls['Control_method'] = 'MRL4'
        risk_realized = 1
        controls['Last_replan'] = Ta

    #MRL 3: Added requirements
    elif (project_vectors[0:m,1] > original_project_vectors[0:m,1]).any():

```

```

scope_matrix[1], scope_matrix[2], scope_matrix[29] = RL3[0][1], RL3[0][2], RL3[0][29] #replace planned and
intermediate scope quantities and intermediate quality with preplanned values
task_matrix[1:16], task_matrix[30:52] = RL3[1][1:16], RL3[1][30:52] #Update task plan
project_vectors[:,2] = RL3[2][:,2]
planned_teams = RL3[3]
controls['Control_method'] = 'MRL3'
risk_realized = 1
controls['Last_replan'] = Ta

#Baseline plan is MRL1 - most likely - implemented in inition function at beginning of project

#Update plan based on project priorities? - separate functions for each control method to update plans to include
appropriate risk margin and adjust for global priorities (EVM and MRL)

project_budget = project_vectors[m,2]
BAC = task_matrix[15].sum()
SAC = project_vectors[m+1,2] #Schedule at completion is total networked project schedule.
PAR = BAC / SAC #dollars per day for project burn rate
task_PAR = task_matrix[15] / task_matrix[46]
AT = Ta
AC = task_matrix[61].sum() + Ca + Ta * 8 * constants['FixedOHR'] * humans['Hired']
scope_quantity_status = scope_matrix[5] / scope_matrix[1] #greater than 1 is excess
scope_quality_status = scope_matrix[10] / np.where(scope_matrix[30] == 0,1,scope_matrix[30])[0] #greater than
one is excess
task_EV = np.minimum(1,scope_quantity_status,scope_quality_status) @ scope_to_task_matrix *
task_matrix[15] #scope progress assigned to task times task budget
EV = task_EV.sum()
task_PV = task_matrix[15] * np.power((task_matrix[58] / task_matrix[46]),2) #parabolic increasing function - task
planned value is task budget times (time progress of the task) ^2 Approaches total
budget in quadratic fashion as planned duration is approached. Acknowledges that
target scope is not completed in linear fashion
PV = task_PV.sum()

project_MRL_metrics = ['EV','PV','AC','CV','CPI','SV','SPI','EAC','TCPI','ETC','ES','TV','TPI','TEAC']
project_metrics = pd.DataFrame(data = np.zeros((1, len(project_MRL_metrics))), index = ['Project'], columns =
project_MRL_metrics) #EVM & ES metrics dataframe for project level. One row for
project-level values

project_metrics['EV'] = EV
project_metrics['PV'] = PV
project_metrics['AC'] = AC
project_metrics['CV'] = EV - AC
if AC != 0:
    project_metrics['CPI'] = EV / AC
else:
    project_metrics['CPI'] = max(EV,1)
project_metrics['SV'] = EV - PV
if PV != 0:
    project_metrics['SPI'] = EV / PV
else:
    project_metrics['SPI'] = max(EV,1)
CPI = project_metrics['CPI'].iloc[0]
SPI = project_metrics['SPI'].iloc[0]

```



```

project_metrics['EAC'] = AC + (BAC-EV) / (CPI * SPI)
EAC = project_metrics['EAC'].iloc[0]
project_metrics['TCPI'] = (BAC - EV) / (EAC-AC)
project_metrics['ETC'] = EAC - AC
project_metrics['ES'] = EV / PAR
ES = project_metrics['ES'].iloc[0]
project_metrics['TV'] = ES - AT
project_metrics['TPI'] = ES / AT
TPI = project_metrics['TPI'].iloc[0]
project_metrics['TEAC'] = SAC / TPI

critical_path_tasks = np.where(task_matrix[47]==1)[0] #np.where returns the positions of the values of an array
#that meets the specified criteria. Returns a tuple where the first value [0] of the tuple is
#the position values in a 1-d array
critical_path_EVM_metrics = ['PV','ES','TV','TPI','TEAC']
critical_path_metrics = pd.DataFrame(data = np.zeros((1,len(critical_path_EVM_metrics))), index = ['Critical
#Path'], columns = critical_path_EVM_metrics)

CP_PV = 0
CP_ES = 0
CP_AT = 0
CP_PD = 0
CP_budget = 0
for j in critical_path_tasks:
    CP_PV += task_matrix[15,j] * task_matrix[58,j] / task_matrix[58,j]
    CP_AT += task_matrix[58,j]
    CP_PD += task_matrix[46,j]
    CP_budget += task_matrix[15,j]
CP_ES = CP_PV / PAR
critical_path_metrics['ES'] = CP_ES
critical_path_metrics['TV'] = CP_ES - CP_AT
if CP_AT != 0:
    critical_path_metrics['TPI'] = CP_ES / CP_AT
else:
    critical_path_metrics['TPI'] = max(1,CP_ES)
critical_path_metrics['TEAC'] = SAC / critical_path_metrics['TPI'].iloc[0]

#calculate alternative MRL PVs
#RL contains: [0] scope_matrix, [1] task_matrix, [2] project_vectors, [3] planned_teams,
RL_PVs = []
for RL in RL_plans:
    RL_task_PV = RL[1][15] * np.power((task_matrix[58] / RL[1][46]),2) #parabolic increasing function - task
#planned value is task budget times (time progress of the task) ^2 Approaches total
#budget in quadratic fashion as planned duration is approached. Acknowledges that
#target scope is not completed in linear fashion
    RL_PV = RL_task_PV.sum()
    RL_PVs.append(RL_PV)

#find the PV that closest matches the project EV. Current active plan PV should be same as the active plan from
#the MRL set (baseline is RL1), so doesn't need to be added
EV_PV_comparisons = {} #Create a dictionary to hold tuples of (EV/PV, 1-EV/PV) for each RL (SPI, 1-SPI) aka how
#well work is being accomplished compared to plan
for i in range(len(RL_PVs)):
    key = 'MRL' + str(i)

```

```

diff = abs(1 - EV / RL_PVs[i]) #a project that is executed exactly according to plan will give a 0 diff
EV_PV_comparisons[key] = (EV / RL_PVs[i],diff)

min_diff = 1
min_key = controls['Control_method']
for i in EV_PV_comparisons.keys():
    min_diff = min(EV_PV_comparisons[i][1], min_diff) #finds the MRL where EV and PV are closest
    if EV_PV_comparisons[i][1] == min_diff:
        min_key = i

if controls['Control_method'] != min_key and not risk_realized and (Ta - controls['Last_replan']) >
    controls['Replanning_frequency'] * SAC:
    scope_matrix[1], scope_matrix[2], scope_matrix[29] = RL_plans[int(min_key[3])][0][1],
        RL_plans[int(min_key[3])][0][2], RL_plans[int(min_key[3])][0][29] #replace planned and
        intermediate scope quantities and intermediate quality with preplanned values min_key
        is a string, the [3] value gives the integer of the correct risk level
    task_matrix[1:16], task_matrix[30:52] = RL_plans[int(min_key[3])][1][1:16], RL_plans[int(min_key[3])][1][30:52]
        #Update task plan
    project_vectors[:,2] = RL_plans[int(min_key[3])][2][:,2]
    planned_teams = RL_plans[int(min_key[3])][3]
    controls['Control_method'] = min_key
    controls['Last_replan'] = Ta

#returns a tuple of the prioritized performance comparisons for (scope/cost, scope/time, cost/time) where
    greater than one means numerator is outperforming denominator
priority_compare = priority_dimension(priorities, project_metrics['SPI'].iloc[0], project_metrics['CPI'].iloc[0],
    project_metrics['TPI'].iloc[0]) #SPI used for scope measurement - indicator of whether
    effort (EV measurement of scope) is ahead or behind plan
#Implement project-level control measures. Scope changes will be both quantity and quality, with focus on later
    tasks. Schedule should focus on critical path (global view of project schedule)
if ((project_metrics['CPI'].loc['Project'] < 0.9) or (critical_path_metrics['TPI'].loc['Critical Path'] < 0.9)) and (EV /
    BAC) > 0.1: #project is at least 10% complete - early indicators are unreliable
    if project_metrics['CPI'].loc['Project'] < 0.9:
        #address cost overruns - cut overtime, reduce team sizes, cut scope
        if priority_compare[0] > 1.2: #if scope is outperforming cost, reduce scope quality thresholds. otherwise no
            action (b/c scope is doing even worse)
            for i in np.where(task_matrix[62] == 0)[0]: #for incomplete tasks
                if EV / BAC > 0.2: #Project is at least 20% complete measured by effort --> based on Lukas 2008, CPI
                    should be reliable indicator of performance at this point. Prudent to begin cutting scope
                    targets
                    for k in np.where(scope_to_task_matrix[:,i].astype(bool))[0]: #for each scope item assigned to the
                        incomplete tasks
                            scope_matrix[29,k] = max(0.99 * scope_matrix[29,k], scope_matrix[31,k] / 2) #reduce the
                                intermediate quality threshold by 1% down to a limit of half the cancellation threshold
                            scope_matrix[30,k] -= 0.1*(scope_matrix[30,k] - scope_matrix[31,k]) #bring target scope quality
                                threshold closer to cancellation threshold
                    if EV / BAC > 0.5: #Once project is halfway finished, if CPI hasn't recovered, time to start cutting scope
                        quantities
                        for k in np.where(scope_to_task_matrix[:,i].astype(bool))[0]: #for each scope item assigned to the
                            incomplete tasks
                                scope_matrix[1,k] = max(scope_matrix[1,k] - scope_matrix[0,k], scope_matrix[1,k]-1) #reduce
                                    intermediate scope required by one, stopping at the required excess quantity of

```

```

        intermediate scope over target scope that was specified at project origination. This must
        be done before recalculating target scope because it relies on the old value.
    scope_matrix[0,k] = max(scope_matrix[3,k], scope_matrix[0,k]-1) #reduce target quantity by one,
        stopping at cancellation threshold
    scope_matrix[2,k] = max(scope_matrix[1,k], scope_matrix[2,k] - 1) #reduce planned scope with
        lower limit of required intermediate scope. Must be calculated after required
        intermediate scope change
if priority_compare[2] < 0.8: #if cost is underperforming time, cut overtime and reduce team sizes.
    Otherwise do nothing - time is doing even worse.
    # print('Cost less than sched', priority_compare[2]) #debug
    task_matrix[57] = 8*np.ones((n)) #set all workdays to 8 hours
    human_counter = task_matrix[53].copy()
    for i in np.where(task_matrix[62] == 0)[0]: #for incomplete tasks not on the critical path
        if task_matrix[0,i] and task_matrix[47,i] == 0:
            task_matrix[53,i] = max(1,int(planned_teams[i]/4),min(10,task_matrix[53,i] - 1)) #team must be at
                least 1, no larger than 10. Decrease size by one each time control is run.
            task_matrix[57,i] = 8 #set workday to 8 hours (no overtime)
            human_changes = (task_matrix[53] - human_counter).sum()
            humans = personnel_action(0,human_changes,humans)[0]
if critical_path_metrics['TPI'].loc['Critical Path'] < 0.9: #TPI that matters is along the critical path - the rest is just
    fluff
    #address schedule shortfall in those tasks on the critical path - add resources, increase team sizes, reduce
        scope thresholds, violate dependencies
if priority_compare[1] > 1.2: #if scope is outperforming time, reduce scope quality thresholds. Otherwise do
    nothing, scope is doing worse. Same actions as for cost-scope
    # print('Time less than scope',priority_compare[1]) #debug
    for i in np.where(task_matrix[62] == 0)[0]: #for incomplete tasks on the critical path
        if EV / BAC > 0.2 and task_matrix[47,i] == 1: #Project is at least 20% complete measured by effort -->
            basing on philosophy for CPI, although TPI is not proven to be a reliable indicator - need
            to research
            for k in np.where(scope_to_task_matrix[:,i].astype(bool))[0]: #for each scope item assigned to the
                incomplete tasks
                scope_matrix[29,k] = max(0.99 * scope_matrix[29,k], scope_matrix[31,k] / 2) #reduce the
                    intermediate quality threshold by 1%
                scope_matrix[30,k] -= 0.1*(scope_matrix[30,k] - scope_matrix[31,k]) #bring target scope quality
                    threshold closer to cancellation threshold
if EV / BAC > 0.5: #Once project is halfway finished, if CPI hasn't recovered, time to start cutting scope
    quantities
    for k in np.where(scope_to_task_matrix[:,i].astype(bool))[0]: #for each scope item assigned to the
        incomplete tasks
        scope_matrix[1,k] = max(scope_matrix[1,k] - scope_matrix[0,k], scope_matrix[1,k]-1) #reduce
            intermediate scope required by one, stopping at the required excess quantity of
            intermediate scope over target scope that was specified at project origination. This must
            be done before recalculating target scope because it relies on the old value.
        scope_matrix[0,k] = max(scope_matrix[3,k], scope_matrix[0,k]-1) #reduce target quantity by one,
            stopping at cancellation threshold
        scope_matrix[2,k] = max(scope_matrix[1,k], scope_matrix[2,k] - 1) #reduce planned scope with
            lower limit of required intermediate scope. Must be calculated after required
            intermediate scope change
if priority_compare[2] > 1.2: #if cost is outperforming time, increase overtime and team size and resources
    # print('Time less than cost',priority_compare[2]) #debug
    for i in np.where(task_matrix[62] == 0)[0]:
        if task_matrix[0,i] and task_matrix[47,i] == 1: #if task is active

```

```

        desired_increase = max(10-task_matrix[53,i],int((planned_teams[i] - task_matrix[53,i])/2)) #desired
            increase should bring to at least ten, or increase team by 1
        humans,h = personnel_action(0,desired_increase,humans)
        task_matrix[53,i] = task_matrix[53,i] + h #increase team by number of available humans up to
            desired_increase
        controls['Overtime'] = min(4, controls['Overtime']+1)
        task_matrix[57,i] = min(12, task_matrix[57,i]+1) #increase overtime
        controls['Resource_modifier'] = min(2,1.1*controls['Resource_modifier']) #increase amount of
            resources being allocated

cost_forecast = project_metrics['EAC'].loc['Project']
schedule_forecast = project_metrics['TEAC'].loc['Project']
scope_forecast = scope_matrix[2]
forecast = np.concatenate((scope_forecast, [cost_forecast], [schedule_forecast]))

if EV / BAC > 0.25:
    if cost_forecast > (1 + max(0,(0.5-(EV / (2*BAC)))) * original_project_vectors[m,0]:
        controls['cost_trip'] = 1
    if schedule_forecast > (1 + max(0,(0.5-(ES / (2*SAC)))) * original_project_vectors[m+1,0]:
        controls['schedule_trip'] = 1

return scope_matrix, task_matrix, humans, planned_teams, forecast, project_metrics, controls, project_vectors

def project_initiation(qx, qt, qr, filenotes, team_size, resource_size, uncertainty, accuracy, dependency,
                    control_method, scope_priority, cost_priority, time_priority, record_details, cx, ct, tx, tt,
                    latency):
    print('Initiating', filenotes)
    ""-----Global Variables-----""

    #Must have the following .csv files in the home directory: <Scope & Task DSMs>, Scope_List.csv, Task_List.csv,
        Scope_to_Task.csv, Risk_Registry.csv (when risks are enabled), Planned_Teams.csv
    #Scope DSM: mxm, columns represent scope quality dependence for row entries. Values = 0 or 1
    #Task DSM: nxn, columns represent task dependence for row entries. Values = 0, 1 (Finish-to-Start), 2 (Pacing
        SS/FF)
    #Scope List: Information on Scope Items - Columns are 'ID' (int, starts at 1), 'Title' (str), 'Units' (str), 'Nominal
        Effort' (int), 'Complexity' (float), 'Required Resources' (int), 'PBS' (str), 'Tier' (int),
        'Threshold' (Quantity, int), 'Target' (Quantity, int), 'Intermediate' (Quantity, int), 'Plan'
        (Quantity, int)
    #Task List: Information on Tasks - Columns are 'ID' (int, starts at 1), 'Title' (str), 'WBS' (str), 'Nominal resource
        Ratio' (int)
    #Planned Teams: length m, int values for team size for each task - eventually should be replaced by logic in
        program

resources = {}

resources['Purchased'] = 0 #resource units
resources['Consumed'] = 0 #resource units
resources['Available'] = 0 #resource units
resources['Freeze'] = 0 #1 if no additional resources can be purchased after initial buy

#Initialize Actual Time (Ta) Variable
#Ta measures real time passed in days

```

```

Ta = 0 #days

#Initialize project expenditures aka actual project cost (Ca) Variable
#Ca measures the actual project expenditures in dollars

Ca = 0 #dollars

# Initialize Project Score global variable
# Used to capture the multi-attribute utility of the final project outcome, represented in abstract utility units
    determined as a weighted average based on the final project priorities. 0 is meeting all
    targets exactly, greater than 0 is exceeding aggregate targets, less than 0 is missing
    aggregate targets

project_score = 0

#Project Closing Criteria:
closing_criteria = {}
closing_criteria['Scope_Complete'] = 0 #Intermediate scope quantities and target scope quality thresholds are
    met - Can deliver product that meets requirements
closing_criteria['Scope_Threshold'] = 0 #Cancellation scope quantities are met and cancellation scope quality
    thresholds are met - Can deliver product at minimum quantities
closing_criteria['Cost'] = 0 #Project exceeds cost threshold for cancellation
closing_criteria['Time'] = 0 #Project exceeds schedule threshold for cancellation

# Initialize project priorities - triad of priorities to represent the relative importance of each aspect of the "iron
    triangle" from the customer's perspective
# [scope priority, cost priority, time priority] higher numbers are proportionately higher priority

priorities = {}
customer_priorities = (scope_priority, cost_priority, time_priority)
priorities['scope'], priorities['cost'], priorities['time'] = customer_priorities[0], customer_priorities[1],
    customer_priorities[2]
priorities['scope_cost'] = priorities['scope'] / priorities['cost']
priorities['scope_time'] = priorities['scope'] / priorities['time']
priorities['cost_time'] = priorities['cost'] / priorities['time']

#Initialize and set global project risk margins. Arbitrary here - should be reset in control function to correspond to
    control method
margin = {}
margin['Resource'] = 0.1
margin['Effort'] = 0.1
margin['Unknown'] = 0.2

#Initialize the number of humans available to work on the project
humans = {}
humans['Hired'] = 0 #humans available for project teams - humans assigned to active tasks cannot exceed this
    amount
humans['Assigned'] = 0
humans['Available'] = 0
humans['Freeze'] = 0 #indicates that a hiring freeze has occurred

#Control levers for project control methods
possible_control_methods = [0, 'None', 'Simple', 'EVM', 'MRL0', 'MRL1', 'MRL2', 'MRL3', 'MRL4', 'MRL5']

```

```

controls = {}
controls['Resource_modifier'] = 1 #Project-level: Greater than one gives more resources to tasks - less than one
give fewer
controls['Overtime'] = 0 #Project-level: number of hours of daily overtime authorized
controls['Team_modifier'] = 1 #Project-level: Greater than 1 is larger team sizes; less than 1 is smaller team sizes
controls['Scope_modifier'] = 1 #Project-level: Greater than 1 is increase in target scope; less than 1 is reduction
in target scope
controls['Control_method'] = control_method #'None', 'Simple', 'EVM', 'MRL0', 'MRL1', 'MRL2', 'MRL3', 'MRL4',
'MRL5'
controls['Estimated_exception_rate'] = 0.2 #this is the estimated global exception handling rate at the beginning
of the project - applied to task_matrix[63]
controls['Last_replan'] = 0 #Time value (in actual days) that the last plan/replan was conducted
controls['Replanning_threshold'] = 0.5 #exceeding the current plan by this amount triggers replanning
controls['Replanning_frequency'] = 0.2 #portion of planned duration that must have passed since last replan
before embarking on another replanning effort
controls['Undiscovered_rework_rate'] = 0.05
controls['cost_trip'] = 0
controls['schedule_trip'] = 0
controls['tripped_cost'] = 0
controls['tripped_schedule'] = 0
controls['tripped'] = 0

if controls['Control_method'] not in possible_control_methods:
    print('Invalid Control Method')

#Control Accounts: Dataframe - control account number in header (1.0, 2.0, 3.0, 3.1, 3.2), tasks in each control
account are elements of dataframe
control_accounts = pd.read_csv('Control Accounts.csv').fillna(0).astype(int)

"""-----Global Constants-----"""

# Initialize Global Constants vector
# Represents values for the entire project model that do not change over time

constants = {}

constants['Wage'] = 40 #dollars per labor (effort) hour
constants['Overtime'] = 1.5 #wage factor for overtime pay
constants['FixedOHR'] = 0.77 #total overhead rate as a proportion of the wage
constants['VarOHR'] = 0.77 #variable overhead rate - as a proportion of the wage
constants['FixedOH'] = constants['FixedOHR'] * humans['Hired'] * constants['Wage'] #Fixed overhead per hour,
determined as the number of total humans available for the project times the fixed
overhead rate
constants['CoordWI'] = 0.9 #coordination work inefficiency - represents loss of productivity resulting from working
in groups (intra-group coordination)
constants['CoordQB'] = 0.1 #coordination quality boost - represents increase in quality output resulting from
working in groups
constants['Resource_exponent'] = 0.7 #task resource effort modifier exponent - represents decreasing marginal
returns of adding non-human resources to (or subtracting resources from) a task to
increase/reduce its rate of production
constants['Exception_handling'] = 0.2 #time required for non-nominal exception handling as a proportion of
nominal effort - real value (not estimated or perceived)
constants['Resource_cost'] = 20 #dollars per resource unit

```

```

constants['Tick'] = 1 #days per model cycle (tick)
constants['Control_freq'] = 5 + latency*4 #ticks per control function execution
constants['Measure_lag'] = 2 + latency*2 #delay for measurement values
constants['Adaptations_lag'] = 5 + latency*5 #delay for project control adaptations to take effect
constants['Model_uncertainty'] = uncertainty #Amount of uncertainty incorporated into project model
constants['Model_measurement_accuracy'] = accuracy #How accurate (or inaccurate) measurements are in
    project model 0 is perfect accuracy (no variation)
constants['Dependency_structure'] = dependency #Defines which G and D matrices will be used. Options are:
    'Flat', 'Full Network', 'Project_1'...more to come
constants['Resource_violation'] = 0.5 #quality penalty for working on scope without necessary scope resources -
    factor multiplied by fraction of required resources available
constants['Resource_batch'] = 500
constants['Personnel_reserve'] = 20 #used to signal when to hire more personnel so human pool doesn't reach 0
constants['Initial_hire'] = 100 #size of initial hire
constants['Resource_defect_rate'] = 0.02

```

""-----Risk Realization - Risks realized during initiation -----""

#Risk register

```
risks = pd.read_csv('Risk_Register.csv', index_col = 0)
```

```
if filenotes[-1] != 'O':
```

```
    for i in risks.index:
```

```
        if risks['State'][i] == 0:
```

```
            if i == 2: #Risk 2: Hiring Freeze - Location - Initiation
```

```
                risks.loc[2,'State'] = 1
```

```
                if random.uniform(0,1) < risks['Likelihood'][2]:
```

```
                    constants['Initial_hire'] = 75
```

```
                    humans['Freeze'] = 1
```

```
            if i == 4: #Risk 4: Defective Parts - Location - Initiation
```

```
                risks.loc[4,'State'] = 1
```

```
                if random.uniform(0,1) < risks['Likelihood'][4]:
```

```
                    constants['Resource_defect_rate'] = 0.15
```

""-----Scope & Task Lists, Dependency Matrices-----""

#Scope Items list format: pandas dataframe. Rows: Scope Item ID 1-14. Columns: 'Title' (String Values), 'Units' (String Values), 'Nominal Effort' (Int values), 'Complexity Value' (Float), 'Required Resources' (Int), 'PBS' (String), 'Tier' (Int), 'Threshold' (Int), 'Target' (Int), 'Intermediate' (Int), 'Plan' (Int)

```
scope_item_list = pd.read_csv('Scope_List.csv', index_col = 0)
```

```
if filenotes == 'Quantity_High':
```

```
    scope_file = Path('Validation Files/High Quantity/Scope_List_High.csv')
```

```
    scope_item_list = pd.read_csv(scope_file, index_col = 0)
```

```
if filenotes == 'Quantity_Low':
```

```
    scope_file = Path('Validation Files/Low Quantity/Scope_List_Low.csv')
```

```
    scope_item_list = pd.read_csv(scope_file, index_col = 0)
```

#Task list format: pandas dataframe. Rows: Task ID 1-22. Columns: 'Title' (String Values), 'WBS' (String Values), 'Nominal Resource Ratio' (Int values)

#Different possible lists with differen required resource amounts: 'base', 'double', or 'half'

```

resource_location = 'Validation Files/Resource Level/'
if resource_size == 'base':
    task_list = pd.read_csv('Task_List.csv', index_col = 0)
elif resource_size == 'double':
    resource_double_task = Path(resource_location + 'Task_List_Double.csv')
    resource_double_scope = Path(resource_location + 'Scope_List_Double.csv')
    task_list = pd.read_csv(resource_double_task, index_col = 0)
    scope_item_list = pd.read_csv(resource_double_scope, index_col = 0)
elif resource_size == 'half':
    resource_half_task = Path(resource_location + 'Task_List_Half.csv')
    resource_half_scope = Path(resource_location + 'Scope_List_Half.csv')
    task_list = pd.read_csv(resource_half_task, index_col = 0)
    scope_item_list = pd.read_csv(resource_half_scope, index_col = 0)
else:
    task_list = pd.read_csv('Task_List.csv', index_col = 0)

m = scope_item_list.shape[0] #store the number of scope items

n = task_list.shape[0]

# Dependency Arrays
G = np.empty((m,m)) #Scope item dependency matrix
D = np.empty((n,n))

#Simple Linear Dependency Structure - each scope item depends on all previous items being completed
if constants['Dependency_structure'] == 'Flat':
    dependency_location = Path('Validation Files/Flat/' + "Scope_to_Task.csv")
    G = np.tril(np.ones((m,m)), k=-1)
    D = np.tril(np.ones((n,n)), k=-1)
    scope_to_task_matrix = np.nan_to_num(np.genfromtxt(dependency_location, delimiter = ','))

#Fully networked dependency structure - each scope item is fully linked to all others (all progress depends on all
#other progress) -- Model probably can't handle a G matrix of all 1s, so may have to
#eliminate that part of the full-network dependency (make all 0s)
if constants['Dependency_structure'] == 'Full Network':
    dependency_location = Path('Validation Files/Full Network/' + "Scope_to_Task.csv")
    G = np.zeros((m,m)) #np.ones((m,m)) - np.eye(m) #Represents linear scope dependence - model doesn't
#account for or allow for working scope items out of order with exceptions which would
#occur
    D = 2*np.tril(np.ones((n,n)), k=-1)
    scope_to_task_matrix = np.nan_to_num(np.genfromtxt(dependency_location, delimiter = ','))

#Arbitrary project structure chosen for model - logical flow of scope items with mix of hard and soft dependencies
if constants['Dependency_structure'] == 'Project_1':
    G = np.nan_to_num(np.genfromtxt("Project_1_Scope.csv", delimiter = ',')) #Important to use .csv and not UTF-
#8 .csv formats - byte order mark skips first element in UTF-8
    D = np.nan_to_num(np.genfromtxt("Project_1_Task.csv", delimiter = ','))
    scope_to_task_matrix = np.nan_to_num(np.genfromtxt("Scope_to_Task.csv", delimiter = ','))

#Debugging Project - simple & small
if constants['Dependency_structure'] == 'Debug_Simple':
    G = np.nan_to_num(np.genfromtxt("Simple_Scope_DSM.csv", delimiter = ',')) #Important to use .csv and not
#UTF-8 .csv formats - byte order mark skips first element in UTF-8

```



```

D = np.nan_to_num(np.genfromtxt("Simple_Task_DSM.csv", delimiter = ','))
scope_to_task_matrix = np.nan_to_num(np.genfromtxt("Scope_to_Task.csv", delimiter = ','))

""-----Threshold, Target, and Planned Vectors-----""

#Planned teams assigned in a csv specific to the project, length n
#Planned humans assigned to each team - each task gets a new "team" assigned, even if humans are recycled.
    Arbitrary, but ideally would be an optimization function

#team_size can be 'base', 'double', or 'half' to show performance with different relative team sizes
if team_size == 'base':
    planned_teams = np.nan_to_num(np.genfromtxt("Planned_Teams.csv", delimiter = ','))
elif team_size == 'double':
    constants['Initial_hire'] = 200
    team_location = Path('Validation Files/Team Size/' + "Planned_Teams_Double.csv")
    planned_teams = np.nan_to_num(np.genfromtxt(team_location, delimiter = ','))
elif team_size == 'half':
    constants['Initial_hire'] = 50
    team_location = Path('Validation Files/Team Size/' + "Planned_Teams_Half.csv")
    planned_teams = np.nan_to_num(np.genfromtxt(team_location, delimiter = ','))
else:
    planned_teams = np.nan_to_num(np.genfromtxt("Planned_Teams.csv", delimiter = ','))

if filenotes == 'Extreme-High':
    extreme_location = 'Validation Files/Extreme High/'
    extreme_task = Path(extreme_location + 'Task_List_Extreme_High.csv')
    extreme_scope = Path(extreme_location + 'Scope_List_Extreme_High.csv')
    extreme_teams = Path(extreme_location + "Planned_Teams_Extreme_High.csv")
    task_list = pd.read_csv(extreme_task, index_col = 0)
    scope_item_list = pd.read_csv(extreme_scope, index_col = 0)
    planned_teams = np.nan_to_num(np.genfromtxt(extreme_teams, delimiter = ','))
    humans['Freeze'] = 1
    constants['Resource_defect_rate'] = 0.5
    constants['Initial_hire'] = 22

if filenotes == 'Extreme-Low':
    extreme_location = 'Validation Files/Extreme Low/'
    extreme_task = Path(extreme_location + 'Task_List_Extreme_Low.csv')
    extreme_scope = Path(extreme_location + 'Scope_List_Extreme_Low.csv')
    extreme_teams = Path(extreme_location + "Planned_Teams_Extreme_Low.csv")
    task_list = pd.read_csv(extreme_task, index_col = 0)
    scope_item_list = pd.read_csv(extreme_scope, index_col = 0)
    planned_teams = np.nan_to_num(np.genfromtxt(extreme_teams, delimiter = ','))
    constants['Initial_hire'] = 2200

project_plan_vector = np.nan_to_num(scope_item_list['Plan'].to_numpy().reshape(m,1))
project_plan_vector = np.vstack((project_plan_vector, [[0],[0],[qr]])) #Project Plan

project_target_vector = np.nan_to_num(scope_item_list['Target'].to_numpy().reshape(m,1))
project_target_vector = np.vstack((project_target_vector, [[ct],[tt],[qt]])) #Project targets

project_threshold_vector = np.nan_to_num(scope_item_list['Threshold'].to_numpy().reshape(m,1))

```

```

project_threshold_vector = np.vstack((project_threshold_vector, [[cx],[tx],[qx]])) #Project cancellation
                                thresholds

project_vectors = np.hstack((project_threshold_vector, project_target_vector, project_plan_vector)) #Project
                                Vectors - threshold[:,0], target[:,1], plan[:,2]

""-----Scope and Task Vector Arrays-----""

#Initialize Scope Item Vectors for each scope item in the project, 33 x m
scope_matrix = np.vstack((project_vectors[0:m,1].copy(), #Item Target Scope, St - 0
                            scope_item_list['Intermediate'].to_numpy(), #Item Required Intermediate Scope Quantity
                                (inclusive of eventual target scope quantity), Sr - 1
                            project_vectors[0:m,2].copy(), #Item Planned Scope, Sp - 2
                            project_vectors[0:m,0].copy(), #Item Cancellation Scope quantity Threshold - 3
                            np.zeros((1,m)), #Item Actual Scope Quantity Completed - 4
                            np.zeros((1,m)), #Item Measured Scope Quantity Complete - 5
                            np.zeros((1,m)), #Item Actual Quality of Completed Scope - 6
                            np.zeros((1,m)), #Item Actual Quality of Completed Intermediate Scope - 7
                            np.zeros((1,m)), #Item Actual Quality of Completed Target Scope - 8
                            np.zeros((1,m)), #Item Measured Quality of Completed Intermediate Scope - 9
                            np.zeros((1,m)), #Item Measured Quality of Completed Target Scope - 10
                            np.zeros((1,m)), #Item Actual Nominal Effort - work only (coordination is extra, complexity value *
                                Item Nominal Effort) - 11
                            scope_item_list['Nominal Effort'].to_numpy(), #item Estimated Nominal Effort - 12
                            np.zeros((1,m)), #Item Actual Complexity Value - 13
                            scope_item_list['Complexity Value'].to_numpy(), #Item Estimated Complexity - 14
                            G.T.copy(), #Item dependencies on other items - transpose of G matrix so columns show which
                                items (in rows) impact the item in the column (mxm) - 15-28
                            qr * scope_item_list['Intermediate'].to_numpy(dtype = bool), #Item intermediate scope quality
                                threshold - 29
                            qt * project_vectors[0:m,1].astype(bool), #Item target scope quality threshold - 30
                            qx * project_vectors[0:m,0].astype(bool), #Item Cancellation scope Quality threshold - 31
                            scope_item_list['Required Resources'].to_numpy(), #Scope Item resources required per unit scope
                                item - 32
                            margin['Resource'] * scope_item_list['Required Resources'].to_numpy(), #Additional resources
                                allocated for item risk - 33
                            np.zeros((1,m)) #Last whole scope quantity completed - 34
                        ))

scope_matrix = np.nan_to_num(scope_matrix)

#set actual nominal effort and complexity values - incorporates random number generator to create uncertainty
scope_matrix[11] = random_array(m,scope_matrix[12],0.1*scope_matrix[12]*constants['Model_uncertainty'])
scope_matrix[13] = random_array(m,scope_matrix[14],0.1*scope_matrix[14]*constants['Model_uncertainty'])

#Initialize Task vectors for each task in the project, 62 x n
task_matrix = np.vstack((np.zeros((n)), #Task Active Status: 1 = Active - 0
                            np.zeros((m,n)), #Spji.copy(), #amount of planned scope allocated to each task (mxn) - 1-14
                            np.zeros((n)), #Budget (dollars) for each task - 15
                            np.zeros((m,n)), #Ev.copy(), #Nominal effort (hours) for the quantity of each scope item assigned to
                                each task (mxn) - includes work and coordination - 16-29

```

```

np.zeros((m,n)), #Tp.copy(), #Planned duration (hours) for each scope item assigned to each task
(mxn) - 30-43
np.zeros((n)), #TP.copy(), #Total planned duration (hours) for each task - 44
8*np.ones((n)), #length of planned workday for each task - 45
np.zeros((n)), #TP / (8*np.ones((n))), #planned number of days for each task - 46
np.zeros((n)), #Critical path determined by precedence diagram method (PDM) - 1 is critical task, 0
is not - 47
np.zeros((n)), #Task float calculated by precedence diagram method (PDM) - 48
np.zeros((n)), #np.sum(np.diag(scope_matrix[32]) @ Spji, axis = 0), #Resources required due to
planned scope for each task - 49
np.zeros((n)), #Resources required due to planned humans assigned to task - 50
np.zeros((n)), #(np.sum(np.diag(scope_matrix[32]) @ Spji, axis = 0) + planned_teams *
task_list['Nominal Resource Ratio']), #Planned resources for each task (Scope resources +
team resources) - 51
np.zeros((n)), #Task ready to begin (defined by function) - 1 for ready, 0 for not ready - 52
np.zeros((n)), #Task actual humans assigned (determined when activated) - 53
np.zeros((n)), #Task actual resources assigned (determined when activated) - 54
np.zeros((n)), #Task Resource Effort Modifier (TREM) (determined by function) - effect of non-
nominal resources on productivity - 55
np.zeros((n)), #Task actual duration (hours) - increment by number of actual hours in workday per
tick - 56
8*np.ones((n)), #Actual workday for active task during execution (real-time tracker) -57
np.zeros((n)), #Actual number of days elapsed for task - increment by value of constants['Tick'] per
tick that the task is active - 58
np.zeros((n)), #Task coordination performed - Increment by formula each tick - 59
np.zeros((n)), #Task work performed - increment by formula per tick - 60
np.zeros((n)), #Task actual cost incurred - increment by formula per tick - tracks labor costs only,
resources are calculated in aggregate when purchased, fixed overhead is calculated in
aggregate when project closes - 61
np.zeros((n)), #Task Complete - 62
np.ones((n)) * controls['Estimated_exception_rate'] #this is the estimated exception handling rate
for each task - should be modified during control. Also should be factored into planning
))

```

""-----Initiating Module-----""

""-----Planning Module-----""

#Create Initial Project Plan. Contains nominal values for schedule and budget - does not account for risk, work inefficiency, or team boost effects

```
if controls['Control_method'] == 'EVM':
```

```
    task_matrix, project_vectors = EVM_risk(task_matrix, project_vectors, margin)
```

```
RL_plans = []
```

```
if controls['Control_method'] == 'MRLO' or controls['Control_method'] == 'MRL1' or controls['Control_method'] ==
'MRL2' or controls['Control_method'] == 'MRL3' or controls['Control_method'] == 'MRL4'
or controls['Control_method'] == 'MRL5':
```

```
    print('MRL Run')
```

```
    risk_levels = ('MRLO', 'MRL1', 'MRL2', 'MRL3', 'MRL4', 'MRL5')
```

```

scope_matrix[1,1] = scope_matrix[1,1] + len(risk_levels) - 1 #creating multiple plans adds to scope item for
                    project plans - adds one unit for each plan beyond the first (assumed all projects need
                    one)
for i in risk_levels: #creates plans for each risk level
    controls['Control_method'] = i
    RL_plans.append(MRL_plans(scope_matrix, task_matrix, project_vectors, planned_teams, margin, constants,
                             controls, scope_to_task_matrix, D, task_list))
controls['Control_method'] = 'MRL1'
scope_matrix, task_matrix, project_vectors, planned_teams = RL_plans[1] #set MRL1 as baseline plan for
project start
original_project_vectors = project_vectors.copy()
else:
    scope_matrix, task_matrix, project_vectors, planned_teams = create_plan(scope_matrix, task_matrix,
    project_vectors, planned_teams, margin, constants, controls, scope_to_task_matrix, D,
    task_list)
    original_project_vectors = project_vectors.copy()

print('Planned Teams are:', planned_teams)

"""-----Execution Module-----"""
#While (exit conditions not met)
#Exit conditions
## Intermediate and Target Scope Quantities and Thresholds met
## Cost or Time Cancellation Thresholds met
scope_matrix, task_matrix, resources, humans, Ta, Ca, closing_criteria, priorities, controls =
    execution(scope_matrix, task_matrix, D, G, scope_to_task_matrix, constants, resources,
    humans, planned_teams, task_list, Ta, Ca, closing_criteria, project_vectors,
    original_project_vectors, priorities, controls, RL_plans, filenotes, record_details, margin,
    control_accounts, risks)

"""-----Control Module-----"""
### Executed within execution function
"""-----No Control Sub-Module-----"""
"""-----Simple Control Sub-Module-----"""
"""-----EVM Sub-Module-----"""
"""-----MRL DS Sub-Module-----"""

"""-----Closing Module-----"""

Sa, cost_score, time_score, project_score = closing(scope_matrix, closing_criteria, Ta, Ca, project_vectors,
customer_priorities, original_project_vectors)

if record_details == 1:
    print("\n\nClosing Criteria:", closing_criteria)
    print("\nProduct Final Values are:\n", scope_matrix[4]) #, '\nRequired Intermediate Quantity
    is\n',scope_matrix[1],'\nQuality values are (qr, qt, qx):\n',scope_matrix[29:32])
    print("\nSa is:\n",Sa)
    print("\nCost Score is:", cost_score)
    print("Time Score is:", time_score)
    print("Project Score is:", project_score)
    print("Ta is:", Ta)

```

```

print("Ca is:", Ca)

now = datetime.now()
score_dict = {}
score_dict["Run Name:"] = filenotes
score_dict["Uncertainty"] = uncertainty
score_dict["Accuracy"] = accuracy
score_dict["Control Frequency"] = constants['Control_freq']
score_dict["Measurement lag"] = constants['Measure_lag']
score_dict["Adaptations lag"] = constants['Adaptations_lag']
score_dict["Architecture"] = dependency
score_dict["Cost Score:"] = cost_score
score_dict["Final Cost:"] = Ca
score_dict["Cost Target:"] = project_vectors[m,1]
score_dict["Time Score:"] = time_score
score_dict["Final Time:"] = Ta
score_dict["Time Target:"] = project_vectors[m+1,1]
score_dict["Scope Score:"] = Sa[13]
score_dict["System Quantity"] = scope_matrix[4,m-1]
score_dict["System Quality"] = scope_matrix[8,m-1]
score_dict["Scope Target Quantity"] = original_project_vectors[m-1,1]
score_dict["Scope Target Quality"] = original_project_vectors[m+2,1]
score_dict["Project Score:"] = project_score
score_dict["Closing Criteria: Scope_Complete"] = closing_criteria['Scope_Complete']
score_dict["Closing Criteria: Scope_Threshold"] = closing_criteria['Scope_Threshold']
score_dict["Closing Criteria: Cost"] = closing_criteria['Cost']
score_dict["Closing Criteria: Time"] = closing_criteria['Time']
score_dict["Quality_thesholds: qx"] = original_project_vectors[m+2,0]
score_dict["Quality_thesholds: qr"] = original_project_vectors[m+2,2]
score_dict["Constants"] = constants
score_dict["Cost Forecast Tripped"] = controls['cost_trip']
score_dict["Cancelled Cost"] = controls['tripped_cost']
score_dict["Schedule Forecast Tripped"] = controls['schedule_trip']
score_dict["Cancelled Duration"] = controls['tripped_schedule']

results = []
index_names = []
for key in score_dict.keys():
    if key == "Run Name:" or key == "Constants" :
        None
    else:
        index_names.append(key)
        results.append(score_dict[key])

results_df = pd.DataFrame(data = results, columns = [filenotes], index = index_names)

if record_details == 1:
    # filename =
        Path('Scores/Scores'+filenotes+str(now.month)+str(now.day)+str(now.hour)+str(now.min
        ute)+".txt")
    filename = Path('Scores/Scores'+filenotes+".txt")
    f = open(filename,"w+")
    for i in score_dict.keys():

```

```

        f.write(i + " " + str(score_dict[j]) + ",\n")
    f.write("Date-Time: " + str(now))
    f.close()

return results_df

def main():

    run = 0 #1 to run simulation, 0 just to load functions

    record_details = 0 #1 for detailed csv files on each run with full information on simulation, 0 if not (suitable for
                        large batch runs)
    filenotes = 'MRL0'
    team_size = 'base' #'double','half','base' - specifies initial team sizes for project initiation, reference different csv
                  files in model folder - used for validation only
    resource_size = 'base' #'base','double','half' - specifies amount of resources that teams need for each task -
                    specified in task_list .csv - used for validation only

    now = datetime.now()

    uncertainty = 1 #Set in range of 0 to 1
    accuracy = 1 #Set in range of 0 to 1 - 0 is perfect accuracy (no variation)
    latency = 1 #additional days of latency in measurements and control implementation

    scope_priority = 1
    cost_priority = 1
    time_priority = 1

    #Choose 1:
    # dependency = 'Flat' #Can be 'Flat', 'Full Network', 'Project_1'
    # dependency = 'Full Network'
    # dependency = 'Debug_Simple'
    dependency = 'Project_1'

    #Choose 1:
    # control_method = 'None'
    # control_method = 'Simple'
    # control_method = 'EVM'
    control_method = 'MRL0'
    # control_method = 'MRL1'
    # control_method = 'MRL2'
    # control_method = 'MRL3'
    # control_method = 'MRL4'
    # control_method = 'MRL5'
    control_method = 0

    #Cost, Schedule, Quality Thresholds, Targets
    cx = 80000000
    ct = 15364723 #Set to median for 0 control for low/high uncertainty/accuracy settings

    tx = 4001
    tt = 1216 #Set to median for 0 control for low/high uncertainty/accuracy settings

```

```

qx = 0.9 #/ (priorities['scope'] / max(priorities['cost'], priorities['time']))# Minimum Quality Threshold for Delivery
qt = 1.0 #/ (priorities['scope'] / max(priorities['cost'], priorities['time']))# Target Scope Quality Theshhold
qr = 0.85 #/ (priorities['scope'] / max(priorities['cost'], priorities['time']))# Intermediate Scope Quality Threshold

```

```

score_dict = {}
score_dict["Run Name:"] = 1
score_dict["Uncertainty"]= 1
score_dict["Accuracy"] = 1
score_dict["Architecture"]= 1
score_dict["Cost Score:"]= 1
score_dict["Final Cost:"]= 1
score_dict["Cost Target:"]= 1
score_dict["Time Score:"]= 1
score_dict["Final Time:"] = 1
score_dict["Time Target:"]= 1
score_dict["Scope Score:"] = 1
score_dict["System Quantity"] = 1
score_dict["System Quality"]= 1
score_dict["Scope Target Quantity"] = 1
score_dict["Scope Target Quality"]= 1
score_dict["Project Score:"] = 1
score_dict["Closing Criteria: Scope_Complete"] = 1
score_dict["Closing Criteria: Scope_Threshold"] = 1
score_dict["Closing Criteria: Cost"] = 1
score_dict["Closing Criteria: Time"] = 1
score_dict["Constants"] = 1
score_dict["Quality_thesholds: qx"] = 1
score_dict["Quality_thesholds: qr"] = 1
score_dict["Cost Forecast Tripped"] = 1
score_dict["Cancelled Cost"] = 1
score_dict["Schedule Forecast Tripped"] = 1
score_dict["Cancelled Duration"] = 1

```

```

if run == 1:
    index_names = []
    for key in score_dict.keys():
        if key == "Run Name:" or key == "Constants" :
            None
        else:
            index_names.append(key)

```

```

all_results_df = pd.DataFrame(data = np.zeros(len(index_names)), columns = ['Initialize'], index = index_names)

```

```

results_df = project_initiation(qx, qt, qr, filenotes, team_size, resource_size, uncertainty, accuracy,
                                dependency, control_method, scope_priority, cost_priority, time_priority,
                                record_details, cx, ct, tx, tt, latency)

```

```

all_results_df = pd.concat((all_results_df,results_df), axis = 1)

```

```

filename = Path('Scores/simulation_'+filenotes+str(now.month)+str(now.day)+'.csv')
# filename = filenotes+str(now.month)+str(now.day)+'.csv' #for laptop
all_results_df.to_csv(filename)

```

```
return None
```

```
if __name__ == "__main__":  
    main()
```



## BIBLIOGRAPHY

- [1] J. Mattis, “Summary of the 2018 National Defense Strategy.” Department of Defense, 2018, [Online]. Available: <https://dod.defense.gov/Portals/1/Documents/pubs/2018-National-Defense-Strategy-Summary.pdf>.
- [2] D. Trump, “National Security Strategy of the United States of America.” Office of the President of the United States, Dec. 2017, [Online]. Available: <https://www.whitehouse.gov/wp-content/uploads/2017/12/NSS-Final-12-18-2017-0905.pdf>.
- [3] D. Egel, H. J. Shatz, K. B. Kumar, and E. R. Harshberger, “Defense Budget Implications of the COVID-19 Pandemic,” *The RAND Blog*, Apr. 07, 2020. <https://www.rand.org/blog/2020/04/defense-budget-implications-of-the-covid-19-pandemic.html> (accessed Jan. 05, 2021).
- [4] A. Cordesman, “The Coronavirus and Changes in the FY2021 Budget: Keeping Cuts in American Defense Spending in Perspective,” Center for Strategic and International Studies, Washington, D.C., Apr. 2020. Accessed: Jan. 05, 2021. [Online]. Available: [https://csis-website-prod.s3.amazonaws.com/s3fs-public/publication/200406\\_Coronavirus\\_Budget.GH2\\_.pdf](https://csis-website-prod.s3.amazonaws.com/s3fs-public/publication/200406_Coronavirus_Budget.GH2_.pdf).
- [5] T. Captain, “Can we afford our own future?,” Deloitte LLC, McLean, VA, 2009. [Online]. Available: <https://www2.deloitte.com/content/dam/Deloitte/us/Documents/manufacturing/us-ad-canweaffordourownfuture-0127.pdf>.
- [6] O. Younossi, M. Arena, R. Leonard, C. Roll, J. Arvind, and J. Sollinger, *Is Weapon System Cost Growth Increasing? A quantitative assessment of completed and ongoing programs*. Santa Monica: RAND Project Air Force, 2007.
- [7] M. Schwartz, *Twenty-five Years of Acquisition Reform: Where Do We Go From Here?* Washington, D.C.: Congressional Research Service, 2013, p. 22.
- [8] L. Baldwin and C. Cook, “Lessons from a Long History of Acquisition Reform,” *The RAND Blog*, Jul. 17, 2015. <https://www.rand.org/blog/2015/07/lessons-from-a-long-history-of-acquisition-reform.html> (accessed Jan. 05, 2021).
- [9] F. Kendall, “Implementation Directive for Better Buying Power 2.0 - Achieving Greater Efficiency and Productivity in Defense Spending,” Apr. 24, 2013.
- [10] “Department of Defense Digital Engineering Strategy.” Office of the Deputy Assistant Secretary of Defense for Systems Engineering, Jun. 2018.
- [11] “DEFENSE ACQUISITIONS ANNUAL ASSESSMENT: Drive to Deliver Capabilities Faster Increases Importance of Program Knowledge and Consistent Data for Oversight,” Government Accountability Office, Washington, D.C., Report to Congressional Committees GAO-20-439, Jun. 2020. [Online]. Available: <https://www.gao.gov/assets/710/707359.pdf>.
- [12] Y. H. Kwak and F. T. Anbari, “History, Practices, and Future of Earned Value Management in Government: Perspectives from NASA,” *Project Management Journal*, vol. 43, no. 1, pp. 77–90, Feb. 2012, doi: 10.1002/pmj.20272.
- [13] Project Management Institute, Inc., “PMI Lexicon of Project Management Terms.” Project Management Institute, 2021, Accessed: Sep. 15, 2020. [Online]. Available: <https://www.pmi.org/pmbok-guide-standards/lexicon>.

- [14] C. J. Van Wyngaard, J. H. C. Pretorius, and L. Pretorius, “Theory of the Triple Constraint; A conceptual review,” in *2012 IEEE International Conference on Industrial Engineering and Engineering Management*, Hong Kong, China, Dec. 2012, pp. 1991–1997, doi: 10.1109/IEEM.2012.6838095.
- [15] B. R. Moser, “Scope Patterns for Projects Modeled as Sociotechnical Systems,” *Journal Modern Project Management*, no. VOL04-NUM03, pp. 118–122, Jan. 2017, doi: 10.19255/JMPM01213.
- [16] Project Management Institute, Inc., Ed., *A Guide to the Project Management Body of Knowledge (PMBOK Guide), 6th Ed.*, 6th ed. Newtown Square, PA: Project Management Institute, 2017.
- [17] J. M. Siegelaub, “Six (yes six!) constraints: an enhanced model for project control.,” presented at the PMI® Global Congress 2007—North America, Newtown Square, PA, 2007, [Online]. Available: <https://www.pmi.org/learning/library/six-constraints-enhanced-model-project-control-7294>.
- [18] R. Vought, “Circular A-11.” Executive Office of the President, Office of Management and Budget, Dec. 18, 2019.
- [19] B. R. Moser and R. Wood, “Design of Complex Programs as Sociotechnical Systems,” in *Concurrent Engineering in the 21st Century*, J. Stjepandić, N. Wognum, and W. J. C. Verhagen, Eds. Switzerland: Springer International Publishing, 2015.
- [20] M. D. Cohen, J. G. March, and J. P. Olsen, “A Garbage Can Model of Organizational Choice,” *Administrative Science Quarterly*, vol. 17, no. 1, pp. 1–25, Mar. 1972.
- [21] J. R. Galbraith, “Organization Design: An Information Processing View,” *Interfaces*, vol. 4, no. 3, pp. 28–36, May 1974, doi: 10.1287/inte.4.3.28.
- [22] G. Van Der Vegt, B. Emans, and E. Van De Vliert, “Effects of Interdependencies in Project Teams,” *The Journal of Social Psychology*, vol. 139, no. 2, pp. 202–214, Apr. 1999, doi: 10.1080/00224549909598374.
- [23] N. B. Moe, T. Dingsøy, and T. Dybå, “A teamwork model for understanding an agile team: A case study of a Scrum project,” *Information and Software Technology*, vol. 52, no. 5, pp. 480–491, May 2010, doi: 10.1016/j.infsof.2009.11.004.
- [24] S. Maitlis and M. Christianson, “Sensemaking in Organizations: Taking Stock and Moving Forward,” *ANNALS*, vol. 8, no. 1, pp. 57–125, Jan. 2014, doi: 10.5465/19416520.2014.873177.
- [25] J. E. Mathieu, M. A. Wolfson, and S. Park, “The Evolution of Work Team Research since Hawthorne.,” *American Psychologist*, vol. 73, no. 4, pp. 308–321, May 2018, doi: 10.1037/amp0000255.
- [26] D. D. Walden, G. J. Roedler, K. J. Forsberg, R. D. Hamelin, and T. M. Shortell, Eds., *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 4th ed. Hoboken, New Jersey: John Wiley & Sons, Inc., 2015.
- [27] R. Shishko, *NASA systems engineering handbook*. National Aeronautics and Space Administration, 1995.
- [28] J. R. Turner, *The Handbook of Project-based Management*, 3rd ed. McGraw Hill, 2009.
- [29] A. Sharon, V. Perelman, and D. Dori, “A Project-Product Lifecycle Management approach for improved systems engineering practices,” *INCOSE International Symposium*, vol. 18, no. 1, pp. 942–957, Jun. 2008, doi: 10.1002/j.2334-5837.2008.tb00854.x.

- [30] “Addresses and Discussions at the Conference on Scientific Management Held October 12, 13, 14 Nineteen Hundred and Eleven,” presented at the Conference on Scientific Management, Hanover, NH, 1912.
- [31] W. Clark, *The Gantt Chart: A Working Tool of Management*. New York: The Ronald Press Company, 1922.
- [32] D. G. Malcolm, J. H. Roseboom, C. E. Clark, and W. Fazar, “Application of a Technique for Research and Development Program Evaluation,” *Operations Research*, vol. 7, no. 5, pp. 646–669, Oct. 1959.
- [33] F. K. Levy, G. L. Thompson, and J. D. Wiest, “The ABCs of the Critical Path Method,” *Harvard Business Review*, vol. 41, no. 5, pp. 98–108, Oct. 1963.
- [34] F. P. Brooks, *The Mythical Man-Month: Essays on software engineering*, Anniversary ed. Reading, Mass: Addison-Wesley Pub. Co, 1995.
- [35] V. Ambrosini, C. Bowman, and N. Collier, “Dynamic Capabilities: An Exploration of How Firms Renew their Resource Base,” *British Journal of Management*, vol. 20, pp. S9–S24, Mar. 2009, doi: 10.1111/j.1467-8551.2008.00610.x.
- [36] M. Danilovic and T. R. Browning, “Managing Complex Product Development Projects with Design Structure Matrices and Domain Mapping Matrices,” *International Journal of Project Management*, vol. 25, no. 3, pp. 300–314, 2007.
- [37] A. A. B. Pritsker, “GERT: Graphical Evaluation and Review Technique,” RAND Corporation, Memorandum RM-4973-NASA, Apr. 1966.
- [38] Y.-W. Kim and G. Ballard, “EARNED VALUE METHOD AND CUSTOMER EARNED VALUE,” *J. Construct. Res.*, vol. 03, no. 01, pp. 55–66, Mar. 2002, doi: 10.1142/S1609945102000096.
- [39] P. Solomon and R. Young, *Performance-Based Earned Value*. Wiley-IEEE Press, 2006.
- [40] F. T. Anbari, “EARNED VALUE PROJECT MANAGEMENT METHOD AND EXTENSIONS,” *Project Management Journal*, vol. 34, no. 4, p. 14, Dec. 2003.
- [41] S. Vandevoorde and M. Vanhoucke, “A comparison of different project duration forecasting methods using earned value metrics,” *International Journal of Project Management*, vol. 24, no. 4, pp. 289–302, May 2006, doi: 10.1016/j.ijproman.2005.10.004.
- [42] W. Lipke, O. Zwikael, K. Henderson, and F. Anbari, “Prediction of project outcome,” *International Journal of Project Management*, vol. 27, no. 4, pp. 400–407, May 2009, doi: 10.1016/j.ijproman.2008.02.009.
- [43] F. T. Anbari, “The Earned Schedule - PMI - Anbari.pdf,” presented at the PMI® Research and Education Conference, Limerick, Munster, Ireland, Newtown Square, PA, Jul. 2012.
- [44] H. Khamooshi and H. Golafshani, “EDM: Earned Duration Management, a new approach to schedule performance management and measurement,” *International Journal of Project Management*, vol. 32, no. 6, pp. 1019–1041, Aug. 2014, doi: 10.1016/j.ijproman.2013.11.002.
- [45] M. Ni, P. B. Luh, and B. Moser, “An Optimization-Based Approach for Design Project Scheduling,” *IEEE Trans. Automat. Sci. Eng.*, vol. 5, no. 3, pp. 394–406, Jul. 2008, doi: 10.1109/TASE.2008.916928.
- [46] H.-K. Chang, W.-D. Yu, and S.-T. Cheng, “A Risk-Based Critical Path Scheduling Method (I) : Model and Prototype Application System,” presented at the 34th International Symposium on Automation and Robotics in Construction, Taipei, Taiwan, Jul. 2017, doi: 10.22260/ISARC2017/0072.

- [47] H.-K. Chang, W. Yu, S.-T. Cheng, and T.-M. Cheng, “The Use of a Multiple Risk Level Model to Tackle the Duration of Risk for Construction Activity,” *KSCE J Civ Eng*, vol. 23, no. 6, pp. 2397–2408, Jun. 2019, doi: 10.1007/s12205-019-1757-8.
- [48] M. R. Asadabadi and O. Zwikael, “Integrating risk into estimations of project activities’ time and cost: A stratified approach,” *European Journal of Operational Research*, p. S0377221719309294, Nov. 2019, doi: 10.1016/j.ejor.2019.11.018.
- [49] P. Urgilés, J. Claver, and M. A. Sebastián, “Analysis of the Earned Value Management and Earned Schedule Techniques in Complex Hydroelectric Power Production Projects: Cost and Time Forecast,” *Complexity*, vol. 2019, pp. 1–11, Apr. 2019, doi: 10.1155/2019/3190830.
- [50] W. Z. Yang, S. Q. Xie, Q. S. Ai, and Z. D. Zhou, “Recent development on product modelling: a review,” *International Journal of Production Research*, vol. 46, no. 21, pp. 6055–6085, Nov. 2008, doi: 10.1080/00207540701343895.
- [51] A. Maheshwari, “Industrial Adoption of Model-Based Systems Engineering: Challenges and Strategies,” Purdue University, 2015.
- [52] R. E. Giachetti, “Evaluation of the DoDAF Meta-model’s Support of Systems Engineering,” *Procedia Computer Science*, vol. 61, pp. 254–260, 2015, doi: 10.1016/j.procs.2015.09.208.
- [53] M.-A. Vigneault, C. Botton, H.-Y. Chong, and B. Cooper-Cooke, “An Innovative Framework of 5D BIM Solutions for Construction Cost Management: A Systematic Review,” *Arch Computat Methods Eng*, vol. 27, no. 4, pp. 1013–1030, Sep. 2020, doi: 10.1007/s11831-019-09341-z.
- [54] J. E. McGrath, *Social psychology: A brief introduction*. New York: Holt, Rinehart, and Winston, 1964.
- [55] J. Mihm, C. Loch, and A. Huchzermeier, “Problem–Solving Oscillations in Complex Engineering Projects,” *Management Science*, vol. 49, no. 6, pp. 733–750, Jun. 2003, doi: 10.1287/mnsc.49.6.733.16021.
- [56] I. Nonaka, R. Toyama, and N. Konno, “SECI, Ba and Leadership: a Uni@ed Model of Dynamic Knowledge Creation,” *Long Range Planning*, vol. 33, p. 30, 2000.
- [57] L. Argote and E. Miron-Spektor, “Organizational Learning: From Experience to Knowledge,” *Organization Science*, vol. 22, no. 5, pp. 1123–1137, Oct. 2011, doi: 10.1287/orsc.1100.0621.
- [58] A. W. Woolley, C. F. Chabris, A. Pentland, N. Hashmi, and T. W. Malone, “Evidence for a Collective Intelligence Factor in the Performance of Human Groups,” *Science*, vol. 330, no. 6004, pp. 686–688, Oct. 2010, doi: 10.1126/science.1193147.
- [59] D. N. Ford, J. M. Lyneis, and T. R. B. Taylor, “Project Controls to Minimize Cost and Schedule Overruns: A Model, Research Agenda, and Initial Results,” p. 27, Jan. 2007.
- [60] S. D. Eppinger and T. R. Browning, *Design Structure Matrix Methods and Applications*. Cambridge, MA: Massachusetts Institute of Technology, 2012.
- [61] E. Crawley, B. Cameron, and D. Selva, *System Architecture: Strategy and Product Development for Complex Systems*. Hoboken, New Jersey: Pearson Higher Education, 2016.
- [62] D. Dori, *Object-Process Methodology: a Holistic Systems Paradigm*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.
- [63] D. Dori, A. Bibliowicz, and N. Wengrowicz, *OPCloud*. Haifa, Israel: Technion - Israel Institute of Technology, 2020.

- [64] A. Sharon, O. L. de Weck, and D. Dori, “Improving Project-Product Lifecycle Management with Model-Based Design Structure Matrix: A joint project management and systems engineering approach: A JOINT PROJECT MANAGEMENT AND SYSTEMS ENGINEERING APPROACH,” *Syst. Engin.*, vol. 16, no. 4, pp. 413–426, Dec. 2013, doi: 10.1002/sys.21240.
- [65] A. Sharon and D. Dori, “A Project–Product Model–Based Approach to Planning Work Breakdown Structures of Complex System Projects,” *IEEE Systems Journal*, vol. 9, no. 2, pp. 366–376, 2014, doi: 10.1109/JSYST.2013.2297491.
- [66] M. Vanhoucke, *Project Management with Dynamic Scheduling*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.
- [67] “Adaptive Acquisition Framework,” 2020. <https://aaf.dau.edu/>.
- [68] “NASA Systems Engineering Handbook.” National Aeronautics and Space Administration, 2019, Accessed: Dec. 04, 2020. [Online]. Available: <https://www.nasa.gov/seh/3-project-life-cycle>.
- [69] D. L. Nordquist, “DoD Directive 5000.01: The Defense Acquisition System.” U.S. Department of Defense, Sep. 09, 2020, [Online]. Available: <https://www.esd.whs.mil/DD/>.
- [70] E. Rebentisch, Ed., *Integrating Program Management and Systems Engineering*. Hoboken, New Jersey: John Wiley & Sons, Inc., 2017.
- [71] Defense Acquisition University, “Defense Acquisition Guidebook,” *Defense Acquisition Guidebook*, 2020. <https://www.dau.edu/tools/dag> (accessed Jul. 06, 2020).
- [72] M. T. Pich, C. H. Loch, and A. D. Meyer, “On Uncertainty, Ambiguity, and Complexity in Project Management,” *Management Science*, vol. 48, no. 8, pp. 1008–1023, Aug. 2002, doi: 10.1287/mnsc.48.8.1008.163.
- [73] S. A. Devaux, *Total Project Control*, 2nd ed. CRC Press, 2015.
- [74] E. M. Lord, “DoD Instruction 5000.02: Operation of the Adaptive Acquisition Framework.” U.S. Department of Defense, Jan. 23, 2020, [Online]. Available: <https://www.esd.whs.mil/DD/>.
- [75] *National Defense Authorization Act for Fiscal Year 2016*. 2015.
- [76] Defense Acquisition University, “EVMS Application Requirements Table,” *DAU Milestone Document Identification (MDID)*, 2020. <https://www.dau.edu/mdid/Pages/EVMSApplicationRequirements.aspx> (accessed Dec. 06, 2020).
- [77] Defense Acquisition University, “EVMS Reporting Requirements Table,” *DAU Milestone Document Identification (MDID)*, 2020. <https://www.dau.edu/mdid/Pages/EVMSReportingRequirements.aspx> (accessed Dec. 06, 2020).
- [78] E. M. Lord, “DoD Instruction 5000.85: Major Capability Acquisition.” Office of the Under Secretary of Defense for Acquisition and Sustainment, Aug. 06, 2020.
- [79] McGregor, John S., “Department of Defense Earned Value Management Implementation Guide.” Office of the Under Secretary of Defense for Acquisition and Sustainment, Jan. 18, 2019.
- [80] Department of Defense, *Defense Federal Acquisition Regulation Supplement*. 2020.
- [81] J. S. McGregor, “Department of Defense Earned Value Management System Interpretation Guide.” Office of the Under Secretary of Defense for Acquisition and Sustainment, Feb. 01, 2018.

- [82] D. S. Christensen, "A Review of Cost/Schedule Control System Criteria Literature," *Project Management Journal*, vol. 25, no. 3, pp. 32–39, Sep. 1994.
- [83] Defense Acquisition University, "Milestone Document Identification (MDID)," 2021. <https://www.dau.edu/mdid/pages/Default.aspx> (accessed Jan. 05, 2021).
- [84] N.-C. Wei, C.-P. Bao, S.-Y. Yao, and P.-S. Wang, "EARNED VALUE MANAGEMENT VIEWS ON IMPROVING PERFORMANCE OF ENGINEERING PROJECT MANAGEMENT," *International Journal of Organizational Innovation*, vol. 8, no. 4, p. 20, 2016.
- [85] D. S. Christensen, "The Costs and Benefits of the Earned Value Management Process," *Journal of Parametrics*, vol. 18, no. 2, pp. 1–16, Nov. 1998, doi: 10.1080/10157891.1998.10462568.
- [86] J. Batselier and M. Vanhoucke, "Empirical Evaluation of Earned Value Management Forecasting Accuracy for Time and Cost," *J. Constr. Eng. Manage.*, vol. 141, no. 11, p. 05015010, Nov. 2015, doi: 10.1061/(ASCE)CO.1943-7862.0001008.
- [87] W. Lipke, "Schedule is Different," *The Measureable News*, pp. 10–15, Mar. 2003.
- [88] W. Abba, "The Evolution of Earned Value Management," *Defense AT&L Magazine*, no. March-April 2017, pp. 2–5, 2017.
- [89] Project Management Institute, Inc., *Practice Standard for Earned Value Management*, 2nd ed. Newtown Square, PA: Project Management Institute, 2011.
- [90] A. Nizam and D. A. Elshannaway, "Review of Earned Value Management (EVM) Methodology, its Limitations, and Applicable Extensions," *The Journal of Management and Engineering Integration*, vol. 12, no. 1, p. 13, 2019.
- [91] J. A. Lukas, "Earned Value Analysis - Why it Doesn't Work," p. 10, 2008.
- [92] S. H. Russell, "Earned Value Management: Uses and Misuses," *Air Force Journal of Logistics*, vol. 33, no. 1, pp. 156–161, 2009.
- [93] P. Efe and O. Demirors, "Applying EVM in a Software Company: Benefits and Difficulties," in *2013 39th Euromicro Conference on Software Engineering and Advanced Applications*, Santander, Spain, Sep. 2013, pp. 333–340, doi: 10.1109/SEAA.2013.55.
- [94] P. Solomon, "A Contract Requirement Rule for Program Managers," *Defense AT&L*, no. November-December 2015, p. 4, 2015.
- [95] I. Bembers, M. Jones, E. Knox, and J. Traczyk, "Better Earned Value Management System Implementation Study." Joint Space Cost Council, Jun. 30, 2017.
- [96] P. Seshadri, "rEVaMp: Risk Elicited Earned Value Management Procedure - A Systematic Framework for Accounting for Project Uncertainties," p. 99, Jan. 2009.
- [97] S. T. H. Mortaji, M. Bagherpour, and S. Noori, "Fuzzy earned value management using L-R fuzzy numbers," *Journal of Intelligent & Fuzzy Systems*, vol. 24, no. 2, pp. 323–332, 2013, doi: 10.3233/IFS-2012-0556.
- [98] S. Hendiani, M. Bagherpour, A. Mahmoudi, and H. Liao, "Z-number based earned value management (ZEVM): A novel pragmatic contribution towards a possibilistic cost-duration assessment," *Computers & Industrial Engineering*, vol. 143, p. 106430, May 2020, doi: 10.1016/j.cie.2020.106430.
- [99] J. Batselier and M. Vanhoucke, "Improving project forecast accuracy by integrating earned value management with exponential smoothing and reference class forecasting," *International Journal of Project Management*, vol. 35, no. 1, pp. 28–43, Jan. 2017, doi: 10.1016/j.ijproman.2016.10.003.

- [100] D. S. Christensen, "The Estimate at Completion Problem: A Review of Three Studies," *PROJECT MANAGEMENT JOURNAL*, vol. XXIV, no. n, p. 8, Mar. 1993.
- [101] A. Li, "NASA: Lack of Disciplined Cost-Estimating Processes Hinders Effective Program Management," Government Accountability Office, Report to the Committee on Science, House of Representatives GAO-04-642, May 2004.
- [102] D. S. Cadman, "Integrated Program Management Data and Analysis Report (IPMDAR) Data Item Description (DID)." Acting Principal Deputy Assistant Secretary of Defense, Acquisition Enablers, Apr. 22, 2020.
- [103] J. R. Wertz, J. J. Puschell, and D. F. Everett, Eds., *Space Mission Engineering: The New SMAD*, 1st ed. Microcosm Press, 2011.
- [104] A. Mao, W. Mason, S. Suri, and D. J. Watts, "An Experimental Study of Team Size and Performance on a Complex Task," *PLoS ONE*, vol. 11, no. 4, p. e0153048, Apr. 2016, doi: 10.1371/journal.pone.0153048.