Software Development Risk Analysis (SDRA)

Project Proposal

Trevor Jaskot

Eric Jones

Leo Pacatan

George Mason University

OR 699/SYST 699 – Masters Project Proposal

February 2019

Project Sponsor

Derek Eichin

Office of the Deputy Assistant Secretary of the Army

For Cost and Economics

Networks, Information, Software and Electronics Costing Division

TABLE OF CONTENTS

SECTION 1 - PROJECT OVERVIEW

1.1 BACKGROUND

Department of Defense (DOD) software development problems drive cost overruns and delay in capability delivery. The majority of DOD Major Defense Acquisition Programs (MDAPs) rely on software development to varying degrees, which puts the spotlight on software development teams to improve cost and schedule projections along with the ability to predictably execute projects to the plan.

The nature of software development makes the problem of projecting cost and schedule requirements difficult. Software development is an unpredictable process, regardless of the controls in place. The unpredictability of the process is largely due to the technical expertise required to perform the job and business constraints (think staffing, leadership, client involvement, and external deadlines). The DOD has recently taken a number of steps toward identifying a solution.

First, DOD acquisition projects have historically followed a highly structured, top-down, step-by-step process (called Waterfall Development), based on the assumption that an end state of the project is known at inception (Northern, Mayfield, Benito, & Casagni, 2010). Many projects involving information technology (IT) requirements are transitioning to an agile development framework, which is a more resilient development environment where development occurs in small, discrete steps called sprints that culminate in epics. Epics can be used to track multiple stories or tasks which need tracking for a larger body of work. Epics can be of any size and is determined by the scrum team with reasoning described within the team working agreement. Another key characteristic of agile development is the continual delivery of operational software and client involvement throughout the production, not just at the beginning.

The DOD also established the Defense Innovation Board (DIB) in 2016 to address key problems in capability development and acquisition. The DIB is made up of technology leaders across the nation in private industry to offer a third-party perspective on DOD processes. The first, and ongoing, task of the board is to advise on how to improve the software development processes. To date, the DIB has provided a plethora of guidance on how to manage programs and effective metrics to predict impending cost or schedule slip, but no tangible solutions have been identified for making accurate projections early in the project in order to support budgeting.

DOD analysts are attempting to fill the void by building predictive regression models. These models are supported by historical data and resources. Nonetheless, to date, these models have fallen short in aiding decision makers in improving projections. This is likely a result of the nature of software development discussed above. Predictive models will always have a difficult time anticipating changes in the process due to individuals and the model is built at the team level.

With predictive models falling short in many cases, DOD organizations are taking the initiative to pursue other options. This project is sponsored by the Office of the Deputy Assistant Secretary of the Army (ODASA) Cost and Economics (CE) division and will aim to identify a baseline analytic product to assess risk and project cost and schedule requirements of software development projects.

1.2 Project Sponsor

ODASA-CE provides Army decision-makers with cost, performance and economic analysis in the form of expertise, models, data, estimates, and analysis. The ODASA-CE Networks, Information Systems, Software and Electronics Costing (NISEC) Division is the lead for developing life cycle cost estimates for communications-electronics systems, major automated information systems (AIS), defense business systems (DBS) and software intensive systems. NISEC is stationed at Fort Belvoir, VA, and is the government lead commissioning for this study thru George Mason University (GMU) Volgenau School of Engineering.

1.3 Problem Statement

Software development deficiencies in the DOD drive cost overruns and delay in capability delivery. NISEC does not have a reliable analytic method for projecting risk of delays or cost overruns. This project aims to provide an analytic framework and initial capability product to fill this capability gap.

1.4 Where are we going?

1.4.1 Objective

Develop, document, and deliver baseline model that represents Department of the Army (DA) software development team activities in order to support ODASA-CE Operations Research Analysts (ORAs) on analysis of life cycle cost estimates.

1.4.2 End State

The end state of this project will be a common understanding between sponsor and project team of software development teams and analytic framework for analyzing project risk along with the following deliverables:

1. Clearly and concisely **documented analytic framework and concepts of operation (CONOPs) for software development teams** to inform future model development efforts.
2. **Analytic model** capable of confirming or refuting program office estimates (POEs) for cost / schedule projections with modeled results and performing trade off analysis of program management policies.
3. **Analytic results** derived from a test case to serve as an example use case.

1.4.3 Key Performance Indicators

The following items will be the key performance indicators of project:

1. Product is effectively tuned to predict performance of a test case in a repeatable way
2. ODASA-CE acceptance of product for future studies

SECTION 2 - PRELIMINARY REQUIREMENTS

2.1 Scope

NISEC tasked the team to develop a baseline model to project risk of overrunning cost estimates or schedule timelines and analyze program management policies like staffing and development processes. Delivery of a baseline model according to requirements and specifications is expected at the end of the semester. The model will be clearly documented to serve as a baseline for future model development efforts. The benchmark for success will be acceptance of the model by NISEC.

The solution will account for probabilistic variables and attempt to support predictive and prescriptive objectives. It will model software development at the team level using an Agile development process.

2.2 User Requirements

The end user of this product is the NISEC staff. The staff intends to use the developed analytic framework to confirm or refute program manager projections for software development project. The product must be compatible with government systems for immediate use and implementation. It must also be transparent, repeatable, and support predictive, prescriptive, and descriptive analysis. The end product must support further studies, but does not need to fulfill more than the initial capability requirements.

2.3 Team Considerations

2.3.1 Risks

The environment and nature of this project present potential obstacles to the team. The following are identified risks resulting from these obstacles and the control measures in place:

- Compatibility of product on government systems – Controlled by developing a portable solution that can be available over multiple mediums including hosted as a web application. This will make the product accessible and adaptable.
- Input data – Model variables will have to represent team and project characteristics that are qualitative in nature. Many of these inputs will not be readily available, which puts the project at risk by making it harder to be predictive in nature. The team will control this by using subject matter expert inputs for initial values and tuning the model to test cases provided by the sponsor.
- Transparency – There is a large risk to not clearly communicating the development process modeled in the product because it will fail the criteria of supporting future development. The team will control this by developing separate documentation for all inputs and assumptions and holding frequent update meetings with the project sponsor.

- Timeline – Time available for completion is a risk due to the condensed project timeline driven by the semester schedule. The team will mitigate this risk by controlling the scope and being explicit about products delivered to enable future teams to effectively carry the project forward.
- Personnel availability – All personnel staffed on this project, including the sponsor, work full time jobs that require frequent travel and long hours. This makes coordination of the project difficult as it is on a short timeline. The team mitigates this risk by being forward looking with scheduling reviews and meetings. Additionally, the project staff will hold at least weekly meetings for progress updates and tag ups.

2.3.2 Constraints

The team is constrained by multiple environmental and personal constraints that must be considered.

- Access to unclassified operations and sustainment cost data for software development personnel must be sanitized prior to being made available by the project sponsor.
- There a one semester time limit to develop a SDRA tool.
- This project has the potential to span additional semesters for future GMU graduate students to continue work on, in the event SDRA tool is not ready for operational use at the completion of this semester.
- The team will be working remotely to develop a solution, thus geographical challenges may limit in-person meetings with both team members and the project sponsor.
- The portability of developing the SDRA solution to fit a Government system may arise.

2.3.3 Resources

The sponsor has subject-matter experts (SMEs) and access to the input data on the SDRA project. Python 2.5 and 3.7 computing language programs will be used primarily to construct the coding framework for model development. The use of Microsoft Office Suite will be used for documentation and data tracking in order to make the end result accessible.

2.4 DELIVERABLES

Deliverables under this project will be split between two audiences. The first list of deliverables is client based and will be provided to NISEC. The second list of in support of SYST 699 course deliverables.

NISEC Deliverables:

1. Clearly and concisely **documented analytic framework and concepts of operation (CONOPs) for software development teams** to inform future model development efforts
2. **Analytic model** capable of confirming or refuting program office estimates (POEs) for cost / schedule projections with modeled results and performing trade off analysis of program management policies
3. **Analytic results** derived from a test case to serve as an example use case

SYST 699 Deliverables:

1. Problem Statement Brief
2. Project Proposal (this document and accompanying brief)
3. IPR 2 – Progress Update
4. IPR 3 – Progress Update
5. Final Report and accompanying brief
6. Website hosting contents of the final report

SECTION 3 - LITERATURE REVIEW

Software development in the DOD has a long history of troubles with cost and schedule overruns. As a result, many teams have focused on addressing various parts of the issues present. The following list are studies relevant to this project. Of the documents available, there are none that attempt to be predictive in nature and strictly look at prescribing appropriate policies. This project is intended to be an initial attempt to close that capability gap.

3.1 Agile Project Dynamics

“We gain insight into the dynamics of how Agile development compares to classic “waterfall” approaches by constructing a System Dynamics model for software projects. The Agile Project Dynamics (APD) model captures each of the Agile genes as a separate component of the model and allows experimentation with combinations of practices and management policies. Experimentation with the APD model is used to explore how different genes work in combination with one another to produce both positive and negative effects” (Glaiel, Moulton, Madnick, 1).

The Agile Project Dynamics (APD) paper by MIT professors Firas Glaiel, Allen Moulton, and Stuart Madnick intends to develop an initial flexible system dynamics model to represent the multitudes of impacts on software development in waterfall and agile frameworks. The systems model then simulates the tradeoffs between different development frameworks. The research conducted by this team will be implemented in the inputs and considerations of the model build for this product.

3.2 Handbook for Implementing Agile in Department of Defense Information Technology Acquisition

“This report describes how Agile development principles can be applied to an IT systems engineering effort, and explains how an Agile methodology could be used to benefit DOD Government acquisition and development programs” (MITRE, iii).

This paper, developed by MITRE, will be the foundation of how agile is applied to software development in this project. The team at MITRE looks at how to qualitatively implement agile principles in DOD projects regarding information technology, but does not provide the quantitative approach this project is seeking.

3.3 Review on Traditional and Agile Cost Estimation Success Factor in Software Development Project

“This paper aimed to discuss success factors that influence in traditional and agile cost estimation process for software development project. Literature survey is carried out from the past researches. Then, this paper presents the success factors that bring to the successful of traditional and agile cost estimation in software development project. Realization these factors will help software development communities contribute positively to the success of traditional or agile cost estimation process in software development project” (Mansor, 965).

This paper looks at the predictive side of cost estimation and derives factors that will be incorporated in this study. The quantitative solution to cost estimation is not addressed in this paper.

3.4 Exploring Bottlenecks in Market-Driven Requirements Management processes with Discrete Event Simulation

"This paper presents a study where a market-driven requirements management process is simulated. In market-driven software development, generic software packages are released to a market with many customers. New requirements are continuously issued, and the objective of the requirements management process is to elicit, manage, and prioritize the requirements. In the presented study, a specific requirements management process is modelled using discrete event simulation, and the parameters of the model are estimated based on interviews with people from the specific organization where the process is used" (Host, 323).

3.5 Monitoring Bottlenecks in Agile and Lean Software Development Projects – A Method and its Industrial Use

"To achieve the desired high speed of the projects and the optimal capacity, bottlenecks existing in the projects have to be monitored and effectively removed. The objective of this research is to show experiences from a mature software development organization working according to Lean and Agile software development principles. By conducting a formal case study at Ericsson we were able to elicit and automate measures required to monitor bottlenecks in software development workflow, evaluated in one of the projects" (Staron, 1).

The last two resources focus on identifying bottlenecks in software development quantitatively and qualitatively, respectively. This is a key part of the descriptive analysis in this project to support understanding interdependent processes and bottlenecks. Lessons learned from these studies will be applied as applicable to understanding the driving constraints in NISEC software development teams.

SECTION 4 - INITIAL APPROACH

4.1 Criteria of a Good Approach

In development of the initial approach to this project, the team and sponsor convened to concur upon key criteria of a good approach. These criteria were used to evaluate the ranking among a number of approaches to the problem. This provided a tractable basis for the selection of approach. The following were the criteria used. The first four were determined to be approximately twice as important as the last three. (Full evaluation of the approaches is in Appendix A).

1. Compatibility – Must be executable on a government system or utilize software that is known to be operating in the federal gov't (e.g., SAS, JAVA, ARENA, Promodel, Excel, R, etc.). Getting a certificate of net worthiness may be possible for a new software medium.
2. Transparency – Primary users will be ORAs with light database experience. Understanding of how to manipulate input; change interactions or event sequencing; and analyze output must be communicated. Showing how logical operators work or assumptions of the system are also important.
3. Predictive – Rough order magnitude cost intervals are desired. Close as possible, or point solutions, are prescriptive in nature and don't always have the most use when expressing risk. The tradeoff here is that enveloping a prescriptive model with the ability to run a design of experiments off of it can help prescribe policies.
4. Repeatable – To ensure analysis is replicable between multiple users a system must be present to store and catalog input data and parameters set in the model before it is run. If a Discrete Event Simulation is the approach understanding the seed settings, random number generator, and ability to replicate a scenario is critical as well.
5. Descriptive – Describing the output should be tractable from modeling at the team level (e.g., which team held up the process, which team is the most utilized, etc.).
6. Prescriptive – Being able to provide a difficulty probability for product (e.g., SW products have a 0.2 likelihood of being easy, 0.7 likelihood of being moderate, and 0.1 likelihood of being hard) or a behavioral attribute to the teams (e.g., for deliverables 1-5 the SW team is very productive, then for items 6 through N SW team's productivity goes down, something like fatigue) will help to prescribe best staffing policies.
7. Completeness – As long as the documentation is transparent in nature, delivery of 1.0 should be adjustable with guidance from a sponsor in 1.5, 2.0, …, N.0.

4.2 Analytic Approach

Independently develop discrete event simulation for the accomplishment of project tasks by sprint and epic.

- Research items will be leveraged from the MIT Agile Project Dynamics Model to inform simulation design.

- To be designed with modular framework for tradeoff analysis of development processes.
- Solution deployed with standalone executable and web base application.

Further details regarding specific architecture of the model will be identified in the coming weeks.

4.2.1 Required Inputs

A number of inputs will be required for the success of this project. To name a few, the team will require the cost data for software development personnel, breakdowns of actual software development projects, subject matter expert inputs to variables characterizing the model, and test cases to verify and validate results. All required inputs have been discussed with project sponsor and will be available to the team in a sanitized state.

4.2.2 Expected Results

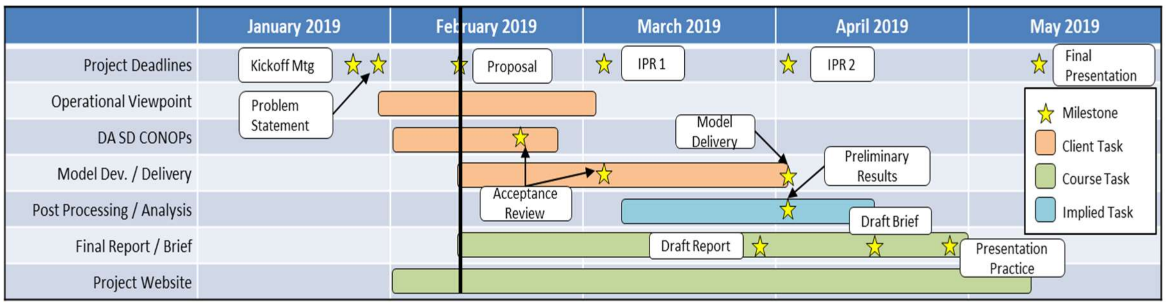
This approach will capture team-level process of software development implementing difference processes inherent to Agile Software Development. The product will output the software development team staffing policies and their associated costs. Also, the product will show where in the process the highest degree of schedule slip or staffing cost risk occurs.

SECTION 5 - PROJECT PLAN

This project will be completed on a quick timeline to conform to the semester and course requirements. Gantt chart and scheme of reviews is below. Risks involved in the timeline and plan are addressed above in the preliminary requirements section.

5.1 Gantt Chart

The following is the schedule of events and deliverables for this project. The black line shows our current position in the timeline. We are on track for all items scheduled to be in progress and head of the schedule on upcoming items.



5.2 Scheme of Review

The team will hold weekly meetings and interim review with the sponsor and among the team during the period of time prior to the first in progress review. Follow on sponsor meetings will be scheduled as needed. Team meetings will continue weekly until completion of the second in progress review. Project requirements will be reevaluated throughout the semester.

WORKS CITED

Glaiel, F., Moulton, A., & Madnick, S. (2014, October 01). Agile Project Dynamics: A System Dynamics Investigation of Agile Software Development Methods. Retrieved February 10, 2019, from https://dspace.mit.edu/handle/1721.1/103024?show=full

Host, M., Regnell, B, Natt, J. Nedstam, J, & Nyuberg, C. (2001 March). "Exploring Bottlenecks in Market-Driven Requirements Management Processes with Discrete Event Simulation." The Fournal of Systems and Software 59. Retrieved February 1, 2019.

Mansor, Z., Yahya, S., & Arshad, N. (2011 January). "Review on Traditional and Agile Cost Estimation Success Factor in Software Development Project." International Journal on New Computer Architectures and Their Applications. Retrieved February 1, 2019.

Northern, C., Mayfield, D., Benito, R., & Casagni, M. (2010, December 15). Handbook for Implementing Agile in Department of Defense Information Technology Acquisition. Retrieved January 28, 2019, from https://apps.dtic.mil/dtic/tr/fulltext/u2/a546756.pdf

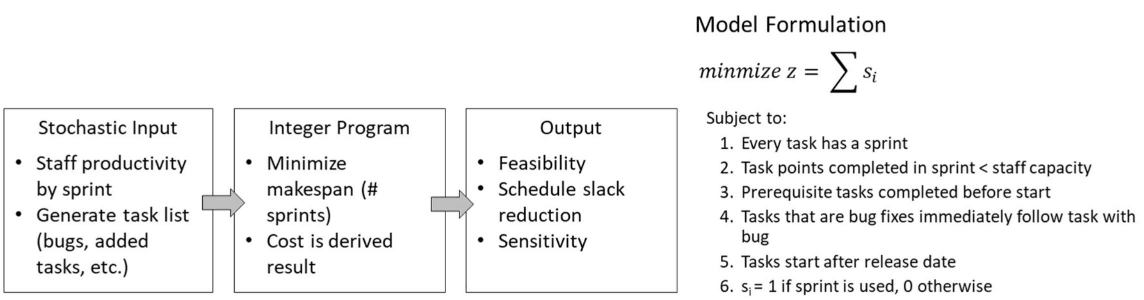
Staron, M. & Meding, W. (2011 January). "Monitoring Bottlenecks in Agile and Lead Software Development Projects – A Method and its Industrial Use." Retrieved February 1, 2019 from https://pdfs.semanticscholar.org/da41/48f65b02c094e19fa7418bb8d724017c2d5d.pdf

APPENDIX A – APPROACH DECISION ANALYSIS

The study team conducted a self-study of the problem-set and amongst each one another provided a self-audit of GMU courses attended to determine the three approaches listed below.

APPROACH 1 – INTEGER / STOCHASTIC PROGRAMMING (I/SP)

Formulate an integer program and wrap in code to adjudicate stochastic influences on constraints prior to optimization.



Pros	Cons
• Transparency – system is visualized through model building • Leverage effort already dedicated to dissect Agile and SW teams • MBSE approach becoming popular for Department of Defense analysis	• SDD modeling tools are not typically commutable (not sure which are available on gov't systems) • High resolution model, with little data to support inputs • Flexibility of task properties is lower than other approaches

APPROACH 2 – SYSTEM DYNAMICS DIAGRAM (SDD)

Largely based on the information from the MIT Agile Project Dynamics model paper was used to develop this approach. It begins with replicating the model from the MIT paper, then attempting to expand and tailor the agile modules.

Pros	Cons
• Transparency – system is visualized through model building • Leverage effort already dedicated to dissect Agile and SW teams • MBSE approach becoming popular for Department of Defense analysis	• SDD modeling tools are not typically commutable (not sure which are available on gov't systems) • High resolution model, with little data to support inputs • Flexibility of task properties is lower than other approaches

APPROACH 3 – DISCRETE EVENT SIMULATION (DES)

Independently develop discrete event simulation for the accomplishment of project tasks by sprint.

- Research items will be leveraged from the MIT Agile Project Dynamics Model to inform simulation design.

- To be designed with modular framework for tradeoff analysis of development processes
- Solution deployed with standalone executable and web base application

Pros	Cons
• Customized approach means DES is most easily deployed solution • Flexible inputs / outputs means DES is highly repeatable and could be implemented in distributed computing environment • Strong for tradeoff analysis of multiple cases	• Coded simulations are inherently less transparent / harder to update and require stricter documentation procedures • Most difficult to deliver an advanced product in one semester • Little data on model inputs means less predictive value

TECHINCAL APPROACH SELECTED METHODOLOGY

All three approaches were evaluated against seven criteria's in an Excel spreadsheet. The study team assigned a numerical ranking to each resulting in Approach 3 being selected, see Table 2.

Table 2. Decision Analysis Approach

$score_a = \sum_{i=1}^{7} s_i r_{a,i} \;\; for\ a\ in\ \{1,2,3\}$

- s_i = criteria weight of criteria i
- $r_{a,i}$ = rank of approach a under criteria i

	Compatible	Transparent	Predictive	Repeatable	Descriptive	Prescriptive	Completeness	WEIGHTED SCORE
1. Optimal Schedule	1	2	3	3	1	1	3	23
2. SDD Replicate	2	3	1	1	3	2	2	21
3. Flexible DES	3	1	2	3	2	3	2	25

Recommend Approach 3 due to its strength in compatibility and utility across analytic objectives