IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 27 November 2023, accepted 14 December 2023, date of publication 18 December 2023, date of current version 26 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3344651

THEORY

Probabilistic Enhancement to the LEAP Process for Identifying Technical Debt in Iterative System Development

HOWARD KLEINWAKS[1,2], ANN BATCHELOR[1], AND THOMAS H. BRADLEY[1], (Member, IEEE)
[1]Department of Systems Engineering, Colorado State University, Fort Collins, CO 80523, USA
[2]Modern Technology Solutions Inc., Alexandria, VA 22312, USA

Corresponding author: Howard Kleinwaks (howard.kleinwaks@colostate.edu)

ABSTRACT The List, Evaluate, Achieve, Procure (LEAP) process defines a methodology for mathematically associating the delivery of system capabilities with the temporal satisfaction of stakeholder needs while identifying technologies at high risk of imparting technical debt into the system. The original process is qualitative, relying on binary definitions of timelines for technology development – the technology either is or is not developed in a specific time period. The binary definitions allow for rapid high-level assessments of the potential for technical debt. However, they fail to capture more realistic scenarios of uncertain technology development timelines. This paper resolves these issues by introducing probability into LEAP process. This paper also provides examples of using the probability in the LEAP process and compares the probabilistic (quantitative) and binary (qualitative) models. These examples show improvements in the ability to assess the likelihood of delivering capabilities in time to meet stakeholder needs when using the probabilistic version of the LEAP process. Since the impact of technical debt is uncertain, the inclusion of probabilities within the LEAP process provides a higher fidelity decision support system for iterative release planning and system development.

INDEX TERMS Iterative development planning, technical debt.

I. INTRODUCTION

Kleinwaks et al. [1] developed the List, Evaluate, Achieve, Procure (LEAP) process to provide a structured approach to identifying technologies that are critical to meeting the stakeholders' needs. This process uses matrix operations to mathematically combine a system functional breakdown with stakeholder needs to identify capabilities that will be delivered late to need and the technologies that drive the delivery timelines. The LEAP process is designed for use within increasingly volatile, uncertain, complex, and ambiguous (VUCA) system development and operating environments [2]. By applying LEAP in an iterative manner, the system developer can identify investments that reduce level of non-recurring engineering (NRE) in system development to enable rapid and successful iterative development cycles [3]. The LEAP process can also be used as a decision support system to assess the long-term impacts of decisions made to achieve a short-term benefit, known as technical debt [4]. Examples of technical debt include minimizing documentation or system modeling and analysis to ensure an on-time release, which can result in increased effort to change the system in the future.

The LEAP process was developed to assist decision makers in identifying critical technologies within iterative system development that may be overlooked by traditional analysis processes. Specifically, the LEAP process includes technical debt contributes to technology development, allowing a decision maker to estimate the long-term effects of short-term decisions.

The LEAP process consists of four major steps [1]:

1. List: establish the system definition by decomposing the stakeholder needs into capabilities and perform a

The associate editor coordinating the review of this manuscript and approving it for publication was Dominik Strzalka.

functional breakdown of the capabilities into enabling technologies
2. Evaluate: assess the capabilities and technologies to determine the need dates and expected development timelines and compute the ability of the system to meet the needs
3. Achieve: identify the technologies that have the largest contribution to late need satisfaction, either in the current time period or in the future. The system developer can invest in these technologies to reduce the development timeline
4. Procure: include mature technologies within a larger-scale development cycle to develop the system that meets the stakeholders' needs

The ability of the system to meet the temporal needs of the stakeholders is computed using matrix operations. Kleinwaks et al. define the process in detail, including the explanation of the supporting mathematics [1]. The baseline process, referred to in this paper as the qualitative LEAP process, is shown in Figure 1.

The qualitative LEAP process relies on three primary inputs: the Functional Matrix (F), the Development Matrix (V), and the Need Matrix (N). These inputs are uniquely defined for each system of interest, based on an analysis of the stakeholder needs and system requirements. The Functional Matrix defines the functional breakdown of capabilities into supporting technologies. The Development Matrix defines the development timelines for each of the technologies. The Need Matrix defines the times at which the stakeholders require each capability. In the qualitative LEAP process, the values in each of these matrices are binary: either zero (0) or one (1). In these matrices, one (1) indicates that the rows and columns are connected – the technology supports the capability, the technology will be developed in the time period, or the capability is needed by the stakeholders in the time period.

The binary inputs enable rapid instantiation of the process and support standard matrix multiplication methods. However, the use of binary inputs provides a qualitative assessment of absolute technology development timelines: technologies will or will not be developed in a given time period. Unfortunately, technology development rarely follows well defined timelines. Technical debt can cause unplanned extensions of project timelines. Within system development, the impact of technical debt is probabilistic. Technical debt interest, the expected amount of work that must be put back into the system to restore functionality [5], may or may not have to be repaid. Repaying the interest can increase the duration of a project, and therefore the impact of technical debt must be modeled as a probabilistic and non-binary contribution to the technology development timeline.

Several methods exist to estimate the duration of a technology development program, such as the critical path method and the program evaluation and review technique (PERT) [6]. Schedule risk analysis combines these methods with Monte Carlo analysis to produce the probability of a technology being developed in a specific time period [7]. To increase the usability of the LEAP process, it needs to be able to input these probabilities into the Development Matrix and to propagate the probabilities through the rest of the analysis.

Therefore, this paper develops improvements and updates to the LEAP process and equations to account for probability and to produce the likelihood of delivering capabilities on time to the stakeholders. In doing so, this paper addressed the following research question:

How can the LEAP process be updated to estimate the probability of delivering capabilities on time to the stakeholders?

The updated process generated as a result of this research will be referred to as the quantitative LEAP process. The rest of this paper is structured in four sections. First, an overview of related work is presented. Next, the quantitative LEAP model is described in detail and an example of its usage and comparison to the qualitative model is provided followed by a discussion of the limitations of the research. Finally, the paper is concluded and recommendations for future work are presented.

II. RELATED WORK

Satisfying stakeholder needs is critical to the success of a project. Unfortunately, satisfying these needs often produces schedule and cost pressure on a system developer, resulting in the introduction of technical debt into the system [8]. Increasing the awareness of technical debt upon its introduction to the system can improve overall project performance [9]. de Almeida, et al. connect technical debt prioritization with business processes, and show accounting for business processes affects how technical debt is prioritized [10]. However, they do not provide a generalizable and mathematical approach to link the stakeholder needs and the system development to assist in the prioritization of development.

The LEAP process provides a novel approach for linking the delivery timeline of system capabilities to the times when the stakeholder needs the capability [1]. It can be used in iterative development scenarios or in project planning. An example of its usage for identifying technological investments is provided in [11]. The primary objective of the process is to identify technologies that may exacerbate system development schedules, resulting in the failure to meet stakeholder needs on time. The LEAP process allows technologies that contribute to the late delivery of multiple capabilities to be identified early. Therefore, the LEAP process can provide leading indicators of technical debt and the impact of technical compromises involving these technologies can be assessed. However, the LEAP process in [1] only deals in absolute delivery timelines and needs to be augmented with probabilistic delivery estimates.

Similar work has been performed by other authors investigating the impacts of rework on project schedules. Rework is associated with the repetition of tasks which were

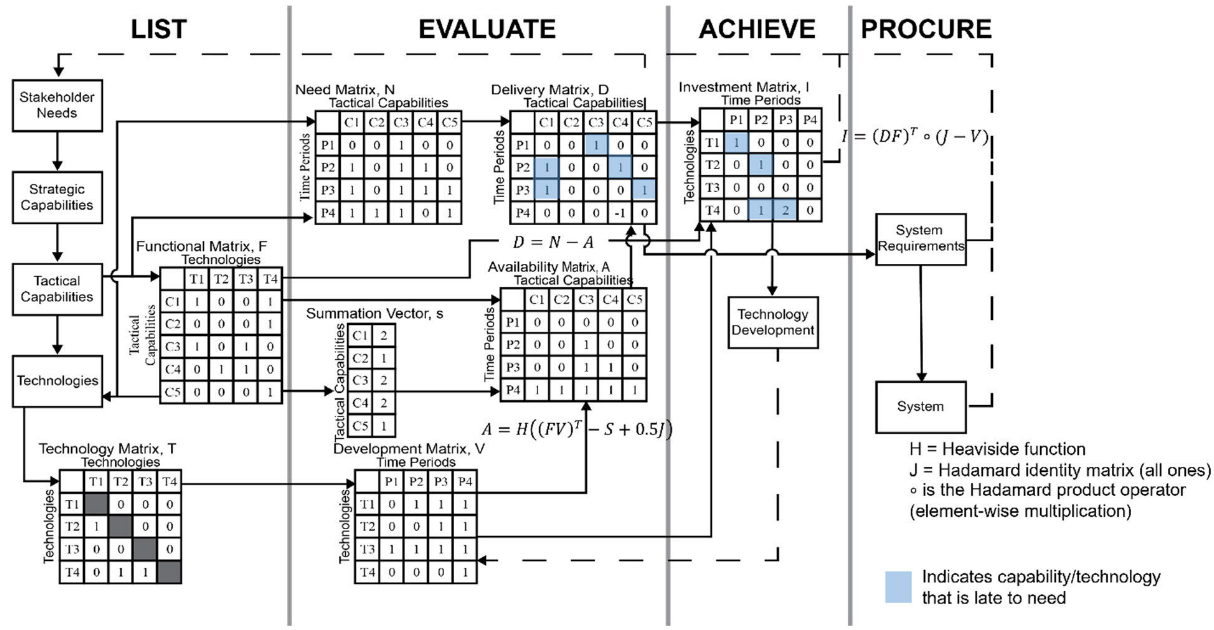


FIGURE 1. The qualitative LEAP process as defined in [1].

not performed to the required quality levels of the project [12] while technical debt is associated with the increased effort required to complete successor tasks [4]. Research on rework includes the connection between project iterations [13], [14], causes of rework [12], [15], and task and project durations [9], [16], [17]. These overlapping conditions are critical to project success, since a successful project requires acceptable performance in addition to on-time and on-budget delivery [17]. Kim incorporates rework probabilities into a linear programming solution to determine the cost of crashing schedule and the impact on total project duration [17]. Smith and Eppinger identify methods to determine which tasks are contributing the most work in iterative design [14], using off-diagonal rework probabilities [13]. While these methods allow for the successive build-up of downstream impacts, they do not account for increases to a successor task's duration based on the technical compromises made during the execution of the predecessor tasks.

Krishnan, Eppinger, and Whitney analyze the duration of successor tasks based on the overlap with predecessor tasks [16]. They assess that starting a successor task too early may increase the effort and duration of the successor task and may also result in a quality loss of the predecessor activity due to a loss of flexibility in the predecessor task. Their model attempts to determine how many iterations to perform with overlapping tasks. However, in many situations, iterations are not included in a project plan and the model does not provide methods to address the impact on the successor tasks of quality loss in a predecessor task. Maheswari and Varghese [18] address task overlaps but do not quantify the rework duration, identifying the assessment of this duration as a critical area for future work.

Ma et al. recognize that current schedule analysis tools offer only passive management capabilities for rework and that leading indicators of rework potential are required [19]. They identify rework probability, the chance of rework occurring, and rework impact, the impact of each activity, and then apply a learning curve to each iteration to measure its impact. This work is similar in concept to the LEAP method in that it attempts to predict the future impact of rework on project schedule. However, it focuses on calculating the iterations required within a project and not on the association between delivery timelines and the satisfaction of stakeholder needs.

The methods and techniques identified in this review focus on the technology delivery aspects of a project – estimating when the project will be complete. While they provide quantitative estimates, they do not directly connect the technology delivery timelines to the need dates of the project stakeholders. The original LEAP process performs this association, but is restricted to qualitative estimates. Therefore, enhancing the LEAP process by adding probabilistic methods will provide a quantitative method to mathematically associate the likelihood of capability delivery with the temporal satisfaction of stakeholder needs.

III. INCLUDING PROBABILITIES IN THE LEAP MODEL

The updates to the LEAP model presented in this section focus on including probabilities in the Development Matrix. The Development Matrix defines the timelines on which the individual technologies are developed [1]. Switching the representation of this matrix from binary values (one (1) and zero (0)) to probabilities enables a more realistic modeling of technology development.

A. MATRIX MULTIPLICATION WITH PROBABILITIES

The original LEAP model [1] uses matrix operations to identify relationships and to compute the availability of capabilities. The Availability Matrix (A), which defines whether or not a capability will be available in a specified time period, is computed by first multiplying the Functional Matrix (F) and Development Matrix (V), which gives a matrix containing the number of developed technologies that support each capability in each time period. The total number of technologies that support the capability (S) is subtracted from the product to determine if the capability is complete. Finally, the Heaviside function (H) is used to restrict the output values to be between zero (0) and one (1). The Availability Matrix calculation is shown in (1) and a complete explanation of the supporting mathematics can be found in [1].

$$A = \mathrm{H}\left((FV)^T - S + 0.5J\right) \tag{1}$$

The critical concept in the Availability Matrix calculation is the combination of the Functional and Development Matrices through matrix multiplication. The dot product of the row of one matrix and the column of the other is used to determine the count of technologies that are developed (column of the Development Matrix) that support the capability (row of the Functional Matrix). The dot product adds the products of each of the corresponding elements of the row and column vectors.

If the Development Matrix includes probabilities instead of binary values, then (1) is no longer valid. Assuming that the development of each technology is independent, then the probability of developing a capability c in a specific time period p is the product of the probabilities of developing each supporting technology t in that same time period p, as depicted in (2).

$$\mathrm{P}\left(C_p\right) = \prod\nolimits_i \mathrm{P}(t_{i,p:t_i\, supports\, c}) \tag{2}$$

The matrix multiplication FV produces a summation of independent probabilities and not the product, as shown in (3). Additionally, (3) includes all the cells of each row of the Functional Matrix in the computation. This inclusion creates a problem when $F[i,j] = 0$. When adding the products of each cell, a zero (0) value in F simply eliminates the corresponding value of V from the sum. However, when multiplying the products of corresponding cells by applying (2), a zero (0) value in F results in a zero (0) product. In the definition of the Functional Matrix, a zero (0) equates to a technology that does not support the capability, and therefore the V value should be eliminated from the product instead of the reducing the product to zero.

$$FV = \begin{bmatrix} \sum_i^n F[0,i] * V[i,0] & \cdots & \sum_i^n F[0,i] * V[i,p] \\ \vdots & \ddots & \vdots \\ \sum_i^n F[m,i] * V[i,0] & \cdots & \sum_i^n F[m,i] * V[i,p] \end{bmatrix} \tag{3}$$

Based on these observations, standard matrix operations do not meet the requirements for updating the LEAP process to include probabilities in the Development Matrix. The required function must input two vectors of the same size and compute the product of the products of corresponding elements, if, and only if, the element of one vector is non-zero.

Two separate functions are required: one that selects the elements of a vector and one that produces the multiplication of the elements in the matrix. These functions are defined in the following sections.

1) SELECTING ELEMENTS OF A VECTOR: THE K FUNCTION

A new function, called the k function, is defined in (4) to select and replace non-zero input values. It inputs three values x, y, and z. If x is not zero (0), then the k function outputs y. If x is zero (0), then the k function outputs z. The function provides a simple method to select a value based on another input. Equation 5 extends the k function to apply to vectors and (6) extends it to matrices. A capital K is used to denote the matrix version of the equation. Note that in (5) vectors $\vec{u}$ and $\vec{v}$ must be the same length and in (6) matrices U and V must have the same dimensions.

$$\mathrm{k}(x,y,z) = \begin{cases} y, & x \neq 0 \\ z, & x = 0 \end{cases} \tag{4}$$

$$\vec{\mathbf{k}}(\vec{\boldsymbol{u}},\vec{\boldsymbol{v}},z) = \left[\mathrm{k}(\vec{\boldsymbol{u}}[0],\vec{\boldsymbol{v}}[0],z) \ \ldots \ \mathrm{k}(\vec{\boldsymbol{u}}[n],\vec{\boldsymbol{v}}[n],z)\right] \tag{5}$$

$$\mathrm{K}(U,V,z) = \begin{bmatrix} \mathrm{k}(U[0,0],V[0,0],z) & \cdots & \mathrm{k}(U[0,n],V[0,n],z) \\ \vdots & \ddots & \vdots \\ \mathrm{k}(U[m,0],V[m,0],z) & \cdots & \mathrm{k}(U[m,n],V[m,n],z) \end{bmatrix} \tag{6}$$

2) MULTIPLYING MATRICES: THE K* FUNCTION

The k function provides the first step of the required multiplication process – the elimination of the zero (0) terms from one of the vectors. A second function is required to address the multiplication of the elements in two matrices instead of the summation. The k* function is defined in (7). For two vectors, it computes the product of the application of the k function to the corresponding elements of the vectors. Equation 8 shows the matrix version of the k* function, denoted with a capital K.

$$\mathrm{k}^*(\vec{\boldsymbol{u}},\vec{\boldsymbol{v}},z) = \prod\nolimits_i^n \mathrm{k}(\vec{\boldsymbol{u}}[i],\vec{\boldsymbol{v}}[i],z) \tag{7}$$

$$\mathrm{K}^*(U,V,z) = \begin{bmatrix} \mathrm{k}^*(F[0,:],V[:,0],z) & \cdots & \mathrm{k}^*(F[0,:],V[:,p],z) \\ \vdots & \ddots & \vdots \\ \mathrm{k}^*(F[m,:],V[:,0],z) & \cdots & \mathrm{k}^*(F[m,:],V[:,p],z) \end{bmatrix} \tag{8}$$

The K* function combines the Functional and Development matrices when the Development Matrix contains probabilities: it eliminates zero values in the Functional Matrix from the product and also multiplies the elements of the matrices instead of adding them.

	T1	T2	T3
C1	1	0	0
C2	1	1	0
C3	0	1	1

Functional Matrix, *F*

	P1	P2	P3
T1	0.5	0.75	1
T2	0	0.5	0.75
T3	1	1	1

Development Matrix, *V*

	P1	P2	P3
C1	0.5	0.75	1
C2	0.5	1.25	1.75
C3	1	1.5	1.75

FV

	P1	P2	P3
C1	0.5	0.75	1
C2	0	0.375	0.75
C3	0	0.5	0.75

K*(*F*,*V*,*1*)

FIGURE 2. Application of the K* function and comparison with matrix multiplication.

3) APPLICATION OF THE K* FUNCTION

The application of the K* function is shown in Figure 2. In the figure, the Development Matrix V contains the probabilities of completing each technology in each time period. If this matrix is multiplied by the Functional Matrix *F* using standard matrix multiplication, the result is the third matrix, *FV*, located in the lower left of the figure. The red cells indicate results where the probability of delivering the capability in a time period are greater than one. Examining the probability calculation can show why simply multiplying these two matrices produces these incorrect results.

The Development Matrix states that C2 is composed of two technologies, T1 and T2, as indicated by the ones (1) in the matrix. Therefore, completing C2 requires completing both T1 and T2. The probability of completing C2 in any time period is the product of the probability of completing T1 and the probability of completing T2 in that time period. In time period P2, the probability of completing C2 is 0.375, as shown in the probability tree in Figure 3. Standard matrix multiplication sums the probabilities instead of multiplying them, giving the incorrect probability of 1.25. The K* function is necessary to introduce the multiplication of probabilities into the matrix operations.

The application of the K* function results in the matrix on the lower right of Figure 2. The same cells are highlighted in red, however, they now have the actual probability values for delivering the capability in the specified time periods. The first row of the final matrix is unchanged between the standard matrix multiplication and the use of the K* function. This row represents the availability of capability C1, which, as seen in the Functional Matrix, only depends on one technology (T1). Therefore, the matrix multiplication and the application of the K* function produce the same results. The

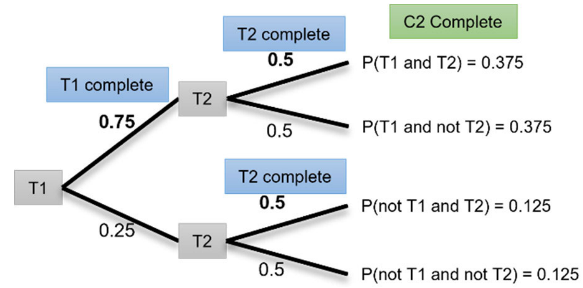


FIGURE 3. Probability tree demonstrating the need for the K* function.

yellow cells in the final matrix changed their values to zero (0). This result is due to the multiplication of probabilities instead of the summation of probabilities. Both capability C2 and capability C3 depend on technology T2. Technology T2 has a zero probability of being developed in time period P1. Therefore, when the probabilities are multiplied, there is a zero probability of completing C2 and C3 in time period P1.

The values for C2 and C3 in time periods P2 and P3 also change between the standard matrix multiplication and the use of the K* function. For the value of C2 in time period P2, standard matrix math uses the dot product of the second row of matrix F and the second column of matrix V, resulting in a summation of the probability of delivering each technology in each time period. However, in this implementation, delivering capability C2 requires technology T1 and technology T2, requiring the probabilities to be multiplied instead of summed. Therefore, the results in the FV matrix in Figure 2 are incorrect. However, the application of the K* function results in the multiplication of the probabilities of delivering the technologies, thereby producing the correct probability of delivering the capability in the time period. These cells are highlighted in red in the figure.

B. INCLUDING PROBABILITIES IN THE LEAP PROCESS

Having demonstrated the usage of the K* function to combine probabilities in matrix multiplication, the LEAP equations presented in [1] can be updated to account for the probabilistic Development Matrix. As a result of these updates the Availability and Delivery Matrices, which are the outputs of the Evaluation phase of the LEAP process, will both produce probabilistic values for capability availability and delivery. The probabilistic Delivery Matrix defines the likelihood of meeting the stakeholders' needs on time and therefore becomes a decision aid for the system developer.

1) AVAILABILITY MATRIX

The Availability Matrix determines if a capability will be available in a specific time period [1]. The K* function computes this probability when applied to the Functional and Development Matrices. Therefore, calculating the probabilistic Availability Matrix requires using the K* function as shown in (9).

$$A = \left(\mathrm{K}^{*}(F, V, 1)\right)^{T} \tag{9}$$

Within the K^* function, z is set to one (1) such that a zero (0) value in the Functional Matrix translates to one (1) in the multiplication instead of the value in the Development Matrix. This choice effectively eliminates the zero entry in the Functional Matrix from the probability computation. This behavior is desired since that capability is not dependent on the technology. The transpose of the K^* function is taken to produce an Availability Matrix with the same dimensions as the Need Matrix in [1].

2) DELIVERY MATRIX

While the Availability Matrix specifies when capabilities are available, the Delivery Matrix (D) defines whether or not the capabilities are delivered in time to meet the stakeholders' needs. In the qualitative LEAP process, the Delivery Matrix is calculated by subtracting the Availability Matrix from the Need Matrix [1]. Applying the same calculation here would result in the Delivery Matrix specifying the probability of *not* delivering the capability on time. Logically, it makes more sense to have the Delivery Matrix indicate the probability of delivering on time instead. Since the Availability Matrix contains probability values, it is necessary to distinguish between capabilities that have zero probability of being delivered on time and those that are not needed in a time period. Therefore, the Delivery Matrix is calculated using the K function on the Need and Availability Matrices as shown in (10). The z value in the K function is set to negative one (-1) to identify the time periods where a capability is not needed.

$$D = \mathrm{K}\left(N, A, -1\right) \tag{10}$$

The values in the Delivery Matrix take on different meaning than those in the qualitative LEAP process. A value that is greater than or equal to zero (0) indicates the probability of delivering a capability in the time period. A negative value indicates that the capability is not needed in that time period.

3) INVESTMENT MATRIX

The final matrix produced by the LEAP process is the Investment Matrix (I). The Investment Matrix identifies those technologies that have the greatest contributions to the late satisfaction of stakeholder needs. In the qualitative LEAP formulation, the values in the Investment Matrix represent the number of late capabilities contributed to by each technology [1]. Using the probabilistic formulation of the Development Matrix, the values in the Investment Matrix become a score – the higher the value, the larger the impact of the technology. The updated Investment Matrix equation is shown in (11).

$$I = (NF)^{T} \circ (J - V) \tag{11}$$

The Need and the Functional Matrices both contain binary values, so standard matrix multiplication is used. This product gives the number of needed capabilities affected by a specific technology. J is the Hadamard identity matrix, which is a matrix of all ones (1) [20]. Subtracting the Development Matrix, V, from J, produces a matrix of probabilities of *not* delivering technologies. The two resulting matrices are then combined element-wise using the Hadamard product (∘) [20], producing an Investment Matrix where each value is the number of affected capabilities times the probability of late delivery.

TABLE 1. Examples of investment matrix scores.

	# of Late Capabilities Impacted	Probability of Late Delivery	Score
Formula	$(NF)^T$	$(J - V)$	$(NF)^T \circ (J - V)$
Technology 1	1	0.1	0.1
Technology 2	4	0.1	0.4
Technology 3	1	0.7	0.7
Technology 4	4	0.7	2.8

Larger scores in the Investment Matrix represent a greater contribution of that technology to the late delivery of the system in the specified time period. The score is the number of affected late capabilities times the probability of late delivery of the technology. Table 1 shows examples of Investment Matrix scores including the number of impacted capabilities and the probability of late delivery.

From these examples, it can be clearly seen that the score provides additional insight into the importance of a technology. Larger scores indicate a larger potential return-on-investment (ROI) if the likelihood of delivering the technology on time can be increased. Consider a situation where a choice is made to invest in either Technology 2 or Technology 3. The qualitative LEAP model would imply that Technology 2 provides the bigger ROI as it impacts more capabilities than Technology 3. The quantitative LEAP process, on the other hand, indicates that Technology 3 provides the bigger ROI. Although it only affects one capability, it has a much higher likelihood of delivering late and therefore a correspondingly larger score.

4) ADJUSTMENTS FOR DEPENDENT TECHNOLOGIES

The above process relies on an assumption of independence between the technologies. In situations where technologies depend upon each other, the model defined above will incorrectly calculate the probabilities. This restriction is remedied by redefining the Functional Matrix. The Functional Matrix maps the capabilities to the supporting technologies. When the technologies are independent, then all technologies should be included in each row of the Functional Matrix. However, if technologies are dependent upon each other, then only the latest technology should be included in the row in the Functional Matrix. For example, consider the Functional Matrix in Figure 2. If Technology 2 is dependent upon Technology 1, then the Functional Matrix would be rewritten as shown in Figure 4, with Capability 2 only showing Technology 2 as a supporting technology. The highlighted cell indicates the change in the matrix. With this redefinition of the Functional Matrix, the capabilities are still composed of independent technologies and the rest of the analysis process is valid.

	T1	T2	T3
C1	1	0	0
C2	0	1	0
C3	0	1	1

FIGURE 4. Functional matrix accounting for technology dependencies.

	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032
T1				1	1	1	1	1	1	1	1	1
T15	1	1	1	1	1	1	1	1	1	1	1	1
T2		1	1	1	1	1	1	1	1	1	1	1
T4			1	1	1	1	1	1	1	1	1	1
T6				1	1	1	1	1	1	1	1	1
T3				1	1	1	1	1	1	1	1	1
T8										1	1	1
T10					1	1	1	1	1	1	1	1
T11		1	1	1	1	1	1	1	1	1	1	1
T12		1	1	1	1	1	1	1	1	1	1	1
T7												1
T13	1	1	1	1	1	1	1	1	1	1	1	1
T5		1	1	1	1	1	1	1	1	1	1	1
T9			1	1	1	1	1	1	1	1	1	1
T14						1	1	1	1	1	1	1
T22		1	1	1	1	1	1	1	1	1	1	1
T21		1	1	1	1	1	1	1	1	1	1	1
T16		1	1	1	1	1	1	1	1	1	1	1
T17			1	1	1	1	1	1	1	1	1	1
T18		1	1	1	1	1	1	1	1	1	1	1
T19			1	1	1	1	1	1	1	1	1	1
T20					1	1	1	1	1	1	1	1
T23		1	1	1	1	1	1	1	1	1	1	1
T24				1	1	1	1	1	1	1	1	1
T25		1	1	1	1	1	1	1	1	1	1	1
T26						1	1	1	1	1	1	1

	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032
T1	0.3	0.5	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T15	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T2	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T4	0.5	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T6	0.3	0.5	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T3	0.3	0.5	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T8	0.0	0.0	0.0	0.0	0.1	0.2	0.3	0.5	0.7	0.8	0.9	1.0
T10	0.2	0.3	0.5	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0
T11	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T12	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T7	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.2	0.3	0.5	0.7	0.8
T13	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T5	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T9	0.5	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T14	0.1	0.2	0.3	0.5	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0
T22	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T21	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T16	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T17	0.5	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T18	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T19	0.5	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T20	0.2	0.3	0.5	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0
T23	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T24	0.3	0.5	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T25	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T26	0.1	0.2	0.3	0.5	0.7	0.8	0.9	1.0	1.0	1.0	1.0	1.0

FIGURE 5. Qualitative (left) and quantitative (right) development matrices, based on [11].

C. EXAMPLE APPLICATION OF THE QUANTITATIVE LEAP PROCESS

The updates to the LEAP process are best understood through an example application. Kleinwaks, et al. [11] applied the qualitative LEAP process to the development of optical terminals at the Space Development Agency. As an example of the quantitative LEAP process, this work is modified to use notional probabilistic values in the Development Matrix. The left side of Figure 5 shows the initial qualitative Development Matrix from [11] after notional investments were made to increase the likelihood of meeting the stakeholder capabilities. The values in this matrix were based on expert judgement. In order to use the quantitative LEAP process, probability distributions for the delivery of each technology are required. These technologies represent new capabilities, each of which has its own development cycle and critical path. Since data on the duration of these projects and their critical paths do not exist, the basic assumption from the Program Evaluation and Review Technique (PERT) that the project duration is represented by a normal distribution is used [21]. Using the normal distribution requires determining the mean and the standard deviation for each technology. These were set by the same experts who estimated the original technology development timelines in [11]. For each technology, the mean was set to the first time period identified in [11], minus two years, and the standard deviation set to two years. The same distribution parameters were applied to each technology based on the expert's opinions that the uncertainties in the technology development timelines were similar for all the identified technologies. The selected distribution results in an 84% probability of delivering at the times identified in the qualitative analysis. The probability of delivering each technology in each time period is computed from the distribution. The resulting probabilistic Development Matrix is shown on the right side of Figure 5, where the colors go from red (low probability of delivering) to green (high probability of delivering).

	C1	C2	C3	C4	C5	C6	C7
2021	0	0	0	0	0	0	0
2022	-1	0	0	0	0	0	0
2023	0	0	0	0	0	0	0
2024	0	0	0	0	0	0	0
2025	0	0	0	0	-1	0	0
2026	0	0	0	-1	-1	0	0
2027	0	0	0	-1	-1	0	0
2028	0	0	0	0	-1	0	1
2029	0	0	0	0	-1	0	1
2030	0	0	0	0	-1	0	1
2031	0	0	0	0	-1	0	1
2032	0	0	0	0	-1	0	0

	C1	C2	C3	C4	C5	C6	C7
2021	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
2022	-1.00	-1.00	-1.00	-1.00	0.50	-1.00	-1.00
2023	0.72	0.42	-1.00	-1.00	0.76	-1.00	-1.00
2024	0.90	0.73	-1.00	-1.00	0.91	0.51	-1.00
2025	0.97	0.91	0.79	-1.00	-1.00	0.77	-1.00
2026	0.99	0.98	0.92	-1.00	-1.00	0.92	-1.00
2027	1.00	1.00	0.97	-1.00	-1.00	0.98	-1.00
2028	1.00	1.00	0.99	0.98	-1.00	1.00	0.15
2029	1.00	1.00	1.00	0.99	-1.00	1.00	0.31
2030	1.00	1.00	1.00	1.00	-1.00	1.00	0.50
2031	1.00	1.00	1.00	1.00	-1.00	1.00	0.69
2032	1.00	1.00	1.00	1.00	-1.00	1.00	0.84

FIGURE 6. Qualitative (left) and quantitative (right) delivery matrices, based on [11].

The probabilistic Development Matrix is used in the quantitative LEAP process to determine the likelihood of delivering the capabilities in each time period. Figure 6 shows the Delivery Matrix from [11] on the left and the probabilistic Delivery Matrix on the right. The qualitative LEAP Delivery Matrix uses zero (0) to indicate that the capability is either on time or not needed in a specific time period and one (1) to indicate that the capability is late. In Figure 6, late capabilities are highlighted in red in the qualitative Delivery Matrix. The quantitative LEAP Delivery Matrix gives the probability of the capability being ready in a time period it is needed, or a negative one (-1) if the capability is not needed. In the quantitative Delivery Matrix in Figure 6, the color scale goes from low likelihood of delivering a needed capability (red) to a high likelihood of delivering the needed capability (green). White cells indicate when the capability is not needed.

In the qualitative Delivery Matrix, capability C7 is late marked as late to need in 2028 (a red 1). However, in the quantitative matrix, the probability of delivering C7 in 2028 is 0.15. While this probability is small, there is still a chance of delivering the capability on time. There is a greater than 50% chance that C7 is delivered in 2031, while the qualitative matrix says that it still will not be ready. Other capabilities, such as C5, may be late (a 50% chance of delivering in 2022) at their first needed time period, a factor which is missed in the qualitative LEAP matrix. The movement away from the binary nature of the qualitative LEAP model increases the fidelity and the realism of the Delivery Matrix. The quantitative model clearly distinguishes between time periods where the capability is delivered on time and when it is not required. For example, the quantitative model identifies that capability C1 is required in 2023 and that there is a 72% probability of it being delivered on time. The qualitative model shows a zero (0) in the entry for C1 in 2023, which is interpreted as either delivering on time or not being needed. The increased fidelity of the model makes the quantitative LEAP model more effective in predicting outcomes for the stakeholders.

	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032
T1	0	0	0	0	0	0	0	0	0	0	0	0
T15	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	0	0	0	0	0	0	0	0	0	0
T8	0	0	0	0	0	0	0	1	1	0	0	0
T10	0	0	0	0	0	0	0	0	0	0	0	0
T11	0	0	0	0	0	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	0	0	0	0	0
T7	0	0	0	0	0	0	0	1	1	1	1	0
T13	0	0	0	0	0	0	0	0	0	0	0	0
T5	0	0	0	0	0	0	0	0	0	0	0	0
T9	0	0	0	0	0	0	0	0	0	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0	0
T22	0	0	0	0	0	0	0	0	0	0	0	0
T21	0	0	0	0	0	0	0	0	0	0	0	0
T16	0	0	0	0	0	0	0	0	0	0	0	0
T17	0	0	0	0	0	0	0	0	0	0	0	0
T18	0	0	0	0	0	0	0	0	0	0	0	0
T19	0	0	0	0	0	0	0	0	0	0	0	0
T20	0	0	0	0	0	0	0	0	0	0	0	0
T23	0	0	0	0	0	0	0	0	0	0	0	0
T24	0	0	0	0	0	0	0	0	0	0	0	0
T25	0	0	0	0	0	0	0	0	0	0	0	0
T26	0	0	0	0	0	0	0	0	0	0	0	0

	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032
T1	0.0	0.0	0.0	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T2	0.0	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T4	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T6	0.0	0.0	0.0	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T3	0.0	0.0	0.0	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.3	0.2	0.1	0.0
T10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T11	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T12	0.0	0.2	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.8	0.7	0.5	0.3	0.2
T13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T5	0.0	0.2	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T22	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T21	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T16	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T17	0.0	0.0	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T18	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T19	0.0	0.0	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T20	0.0	0.0	0.0	0.0	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0
T23	0.0	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T26	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

FIGURE 7. **Qualitative (left) and quantitative (right) investment matrices, based on [11].**

The final calculation in the LEAP model is to determine the Investment Matrix, which highlights which technologies are contributing to the late delivery of capabilities in each time period. In the qualitative model, shown on the left of Figure 7, technologies T8 and T7 are identified as each contributing to a late capability starting in 2028 (shown as red boxes). The quantitative model, shown on the right side of Figure 7, shows the investment 'score' for each of the technologies in each time period with low values in green and high values in red.

The values in the quantitative Investment Matrix take on a slightly different meaning from those in the qualitative model. In the qualitative model, the Investment Matrix values indicate the count of the late capabilities that depend on the technology [1]. In the quantitative model, the value is a score that represents how important the technology is in driving late capability deliveries in the time period. For example, consider technologies T8 and T7 in 2028. In the qualitative model, they both have the same value (1) in the Investment Matrix since they each contribute to the late delivery of a single capability. In the quantitative model, the value for T8 is 0.5 and the value for T7 is 0.841 in 2028. The higher score for T7 indicates a higher potential ROI if the probability of delivering the technology on time could be increased. The cost to increase the delivery probability would need to be accounted for in any ROI calculation, but that is out of the scope of this paper.

D. ACCOUNTING FOR TECHNICAL DEBT IN THE QUANTITATIVE LEAP PROCESS

Including probabilities in the LEAP process enhances its ability to identify the technologies that can be potential sources of technical debt within a system development. Kleinwaks, Batchelor, and Bradley [4] use the technical debt metaphor to reflect the long-term system impacts of short-term decisions. The LEAP framework enables a system developer to rapidly assess the potential for long-term impacts of short-term decisions that impact the development of critical technologies and capabilities. For example, system developers are often faced with choices on the sequencing of technology development due to cost, schedule, and performance limitations. Often, a particular technology is delayed because it is viewed as less valuable, even though it may be necessary for later development tasks.

In the LEAP model, technologies with the high potential for technical debt manifest themselves in the Investment Matrix. Higher scores in the Investment Matrix indicate increased dependencies on on-time delivery and therefore the potential for impacts due to the presence of technical debt. With the probabilistic nature of the quantitative LEAP model, these relationships become clearer as the potential late delivery of a technology can be assessed, including its cascading impacts on the delivery of capabilities.

IV. LIMITATIONS

The research presented in this paper presents a unique perspective on predicting the likelihood of temporal satisfaction of stakeholder needs based on assessment of technology development timelines. However, as presented, this work contains several limitations that can be expanded through continued research into this topic. First, the research relies on the accurate decomposition of a system into capabilities and technologies and the association of stakeholder need dates with those capabilities. If the system decomposition is not done properly, then the results of the analysis can be skewed. Similarly, the calculation of the probabilities contained within the Development Matrix have a heavy influence on the results of the computation – if these probabilities do not reflect realistic assessments, then an inaccurate prediction of the ability to satisfy the stakeholder needs will be generated. In the example provided in this paper, a normal distribution was used to estimate the development timelines of the technologies. Other distribution choices, or other parameter assessments by experts, can alter the end results indicating stakeholder satisfaction. For example, if a series of triangular distributions were used instead of a normal distribution, then the values in the Delivery Matrix would also change. The end results are only as good as the estimates used to set up the problem. The quantitative LEAP process can provide the likelihood of delivering capabilities to meet stakeholder needs, but it is limited by the fidelity of the input information. Similarly, the ability of the quantitative LEAP process to predict the impact of technical debt on the system development requires the development of an outside method to relate the presence of technical debt to technology development timelines.

V. CONCLUSION AND FUTURE WORK

Including probabilities within the LEAP framework enables a more realistic assessment of the ability of the system to deliver in time to meet the stakeholders' needs. This research updates the LEAP process defined in [1] to account for a probabilistic Development Matrix and to propagate those probabilities through the system. This update is critical to better align the LEAP process to real-world systems. Real systems do not guarantee system delivery in a specific time period and the ability to estimate the likelihood of delivery allows for higher fidelity modeling.

The user of the quantitative LEAP process can more accurately assess the potential for achieving technology development by determining the increase in the likelihood of delivering a capability. An achievement initiative may speed up technology development, but does not guarantee that the technology will be achieved in a specific time frame. The quantitative LEAP process enables the modeling of this change in probability, instead of an assumption of complete success. This change in probability can also be mapped more directly to an implementation cost, enabling a calculation of the ROI for these decisions. The score in the Investment Matrix provides a more refined estimate of the impact of a technology's late delivery, highlighting the potential for higher ROI.

These updates to the process represent the second step in defining a full process for accounting for technical debt within system development planning, as defined in [1]. Similar to the qualitative model, the quantitative LEAP framework highlights technologies with the potential for introducing technical debt into the system. Combining the quantitative LEAP framework with a scheduling model that accounts for technical debt will highlight the downstream impacts of the technical debt on the delivery of system capabilities. By modeling the impact of the technical debt of one technology on its successor technologies in the development cycle, the probabilities of delivering each technology in a defined time period can be estimated. These estimates, when included in the quantitative LEAP process, will provide insight to system stakeholders to enable proper investment decisions to limit the risk of late deliveries. Further verification and validation of the process includes implementing the quantitative LEAP process on additional real-world systems to identify insights provided by the process to assist users in delivering capabilities on time.

Additional work can be performed to enhance the analysis process and expand beyond the basic matrix-based system employed in this research paper. Numerous mathematical methods exist to handle uncertainty, including fuzzy sets and fuzzy logic [22] and multi-criteria decision making [23]. These methods need to be reviewed for their application to this specific problem and for their potential benefit in further enhancing the LEAP process.

REFERENCES

[1] H. Kleinwaks, A. Batchelor, T. Bradley, M. Rich, and J. F. Turner, "LEAP—A process for identifying potential technical debt in iterative system development," *INCOSE Int. Symp.*, vol. 33, no. 1, pp. 535–553, 2023.

[2] T. S. Schmidt, S. Weiss, and K. Paetzold, "Expected vs. real effects of agile development of physical products: Apportioning the hype," in *Proc. 15th Int. Design Conf.*, 2018, pp. 2121–2132.

[3] D. M. Tate, "Acquisition cycle time: Defining the problem (Revised)," Inst. Defense Analyses, Alexandria, VA, USA, Tech. Rep. IDA Document NS D-5762, 2016.

[4] H. Kleinwaks, A. Batchelor, and T. H. Bradley, "Technical debt in systems engineering—A systematic literature review," *Syst. Eng.*, vol. 26, no. 5, pp. 675–687, Sep. 2023.

[5] H. Kleinwaks, A. Batchelor, and T. H. Bradley, "Ontology for technical debt in systems engineering," *IEEE Open J. Syst. Eng.*, vol. 1, pp. 111–122, 2023.

[6] F. T. Anbari, "Earned value project management method and extensions," *Project Manage. J.*, vol. 34, no. 4, pp. 12–23, Dec. 2003.

[7] D. T. Hulett, "Advanced quantitative schedule risk analysis," Hulett & Associate LLC, Los Angeles, CA, USA, White Paper, 2004. [Online]. Available: https://www.ipecsystems.com.my/wp-content/uploads/2020/01/Advanced-Quantitaive-Schedule_Risk_Analysis.pdf

[8] H. Kleinwaks, A. Batchelor, and T. H. Bradley, "An empirical survey on the prevalence of technical debt in systems engineering," in *Proc. INCOSE Int. Symp.*, vol. 33, no. 1, 2023, pp. 1640–1658.

[9] T. Walworth, M. Yearworth, J. Davis, and P. Davies, "Early estimation of project performance: The application of a system dynamics rework model," in *Proc. IEEE Int. Syst. Conf. (SysCon)*, Apr. 2013, pp. 204–208.

[10] R. R. de Almeida, U. Kulesza, C. Treude, D. C. Feitosa, and A. H. G. Lima, "Aligning technical debt prioritization with business objectives: A multiple-case study," in *Proc. IEEE Int. Conf. Softw. Maintenance Evol. (ICSME)*, Sep. 2018, pp. 655–664.

[11] H. Kleinwaks, M. Rich, M. C. Butterfield, and J. F. Turner, "LEAPing ahead—The space development agency's method for planning for the future," *INCOSE Int. Symp.*, vol. 33, no. 1, pp. 925–942, 2023.

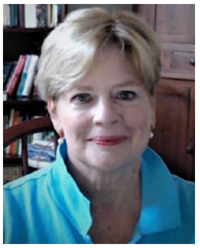
[12] P. E. D. Love, "Influence of project type and procurement method on rework costs in building construction projects," *J. Construct. Eng. Manage.*, vol. 128, no. 1, pp. 18–29, Feb. 2002.

[13] R. P. Smith and S. D. Eppinger, "A predictive model of sequential iteration in engineering design," *Manage. Sci.*, vol. 43, no. 8, pp. 1104–1120, Aug. 1997.

[14] R. P. Smith and S. D. Eppinger, "Identifying controlling features of engineering design iteration," *Manage. Sci.*, vol. 43, no. 3, pp. 276–293, Mar. 1997.

[15] P. E. D. Love, F. Ackermann, J. Smith, Z. Irani, and D. J. Edwards, ‘‘Making sense of rework causation in offshore hydrocarbon projects,’’ *Project Manage. J.*, vol. 47, no. 4, pp. 16–28, Aug. 2016.
[16] V. Krishnan, S. D. Eppinger, and D. E. Whitney, ‘‘A model-based framework to overlap product development activities,’’ *Manage. Sci.*, vol. 43, no. 4, pp. 437–451, Apr. 1997.
[17] J. Kim, C. Kang, and I. Hwang, ‘‘A practical approach to project scheduling: Considering the potential quality loss cost in the time–cost tradeoff problem,’’ *Int. J. Project Manage.*, vol. 30, no. 2, pp. 264–272, Feb. 2012.
[18] J. U. Maheswari and K. Varghese, ‘‘Project scheduling using dependency structure matrix,’’ *Int. J. Project Manage.*, vol. 23, no. 3, pp. 223–230, Apr. 2005.
[19] G. Ma, J. Jia, T. Zhu, and S. Jiang, ‘‘A critical design structure method for project schedule development under rework risks,’’ *Sustainability*, vol. 11, no. 24, p. 7229, Dec. 2019.
[20] E. Million. (2007). ‘‘The Hadamard product. [Online]. Available: http://buzzard.pugetsound.edu/courses/2007spring/projects/million-paper.pdf
[21] M. Hajdu and O. Bokor, ‘‘The effects of different activity distributions on project duration in PERT networks,’’ *Procedia Social Behav. Sci.*, vol. 119, pp. 766–775, Mar. 2014.
[22] L. A. Zadeh, ‘‘Fuzzy sets as a basis for a theory of possibility,’’ *Fuzzy Sets Syst.*, vol. 100, pp. 9–34, Jan. 1999.
[23] M. Aruldoss, T. M. Lakshmi, and V. P. Venkatesan, ‘‘A survey on multi criteria decision making methods and its applications,’’ *Amer. J. Inf. Syst.*, vol. 1, no. 1, pp. 31–43, 2013.

HOWARD KLEINWAKS received the Bachelor of Science and Master of Science degrees in aerospace engineering from MIT and the Doctor of Engineering degree in systems engineering from Colorado State University, Fort Collins, CO, USA. He is currently researching the management of technical debt in iterative systems development. He is a Program Director in the Strategic Space Business Unit at Modern Technology Solutions Inc. (MTSI). He is also a Project Management Professional (PMP), Certified Systems Engineering Professional (CSEP), and Professional Scrum Master. He has over 15 years of experience in the aerospace and systems engineering fields.



ANN BATCHELOR is currently a Professor in systems engineering with Colorado State University, teaching engineering project and program management, systems requirements engineering, and engineering risk assessment. She has more than 20 years of industrial experience in technical management, systems engineering, production, manufacturing, lean engineering, life cycle management, test and analysis, proposal and project management, and technical writing. She has held roles, including a Chief Scientist, a Systems Engineer, the Director of Engineering, and the Director of Program Management. She is also a past certified Program Management Professional (PMP), a Military Sensing Fellow (DOD Informational and Analysis Center for Military Sensing), and the former President-Elect of the INCOSE Atlanta Chapter.

THOMAS H. BRADLEY (Member, IEEE) received the B.S. degree in mechanical engineering from the University of California at Davis and the Ph.D. degree in mechanical engineering from the Georgia Institute of Technology. He is currently the Woodward Foundation Professor and the Head of the Department of Systems Engineering, Colorado State University. He conducts research and teaches a variety of courses in system engineering, multidisciplinary optimization, and design. His research interests include applications in automotive and aerospace system design, energy system management, and lifecycle assessment. He is a member of INCOSE, SAE, ASME, and AIAA.