33rd Annual INCOSE international symposium
hybrid event
Honolulu, HI, USA
July 15 - 20, 2023

LEAP – A Process for Identifying Potential Technical Debt in Iterative System Development

Howard Kleinwaks
Colorado State University
Modern Technology Solutions, Inc
Space Development Agency

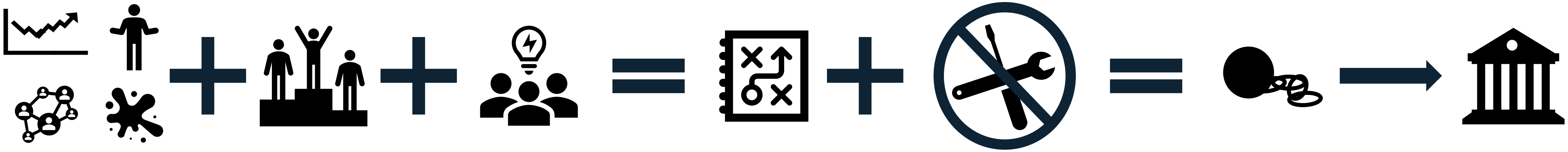
Ann Batchelor
Colorado State University

Dr. Thomas Bradley
Colorado State University

Matthew Rich
GoLion Company
Space Development Agency

Dr. J. F. Turner
Space Development Agency

Introduction



Volatile, Uncertain, Complex, and Ambiguous Environment[4]

Competition[6]

Stakeholder Pressure[3]

Shortcuts[3]

Under-prioritization of non-functional requirements[5]

Technical Debt[7]

Technical Bankruptcy

Development Environment

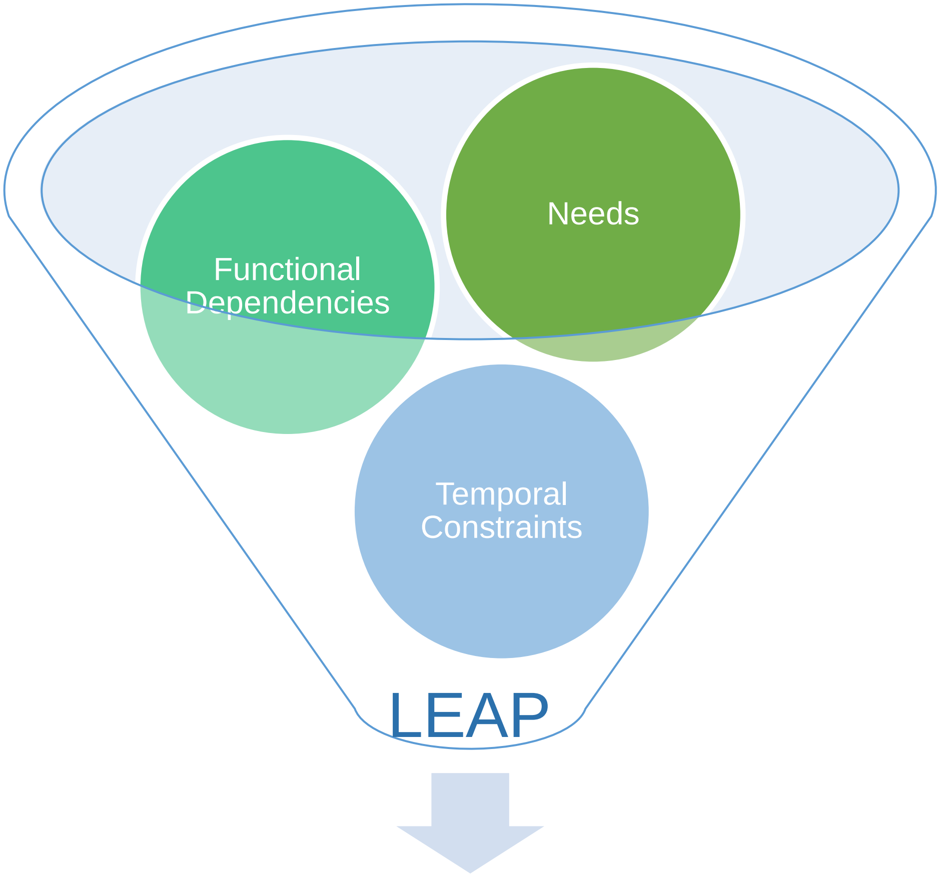
Development Decisions

Consequences

- **Technical debt:** technical compromises that yield short-term benefit but may hurt the long-term health of a system[1]
  - Technical debt originated in software engineering, but is not consistently used or defined within systems engineering[2]
- Technical bankruptcy is the state where the system can no longer proceed due to large amounts of technical debt

Background and Motivation

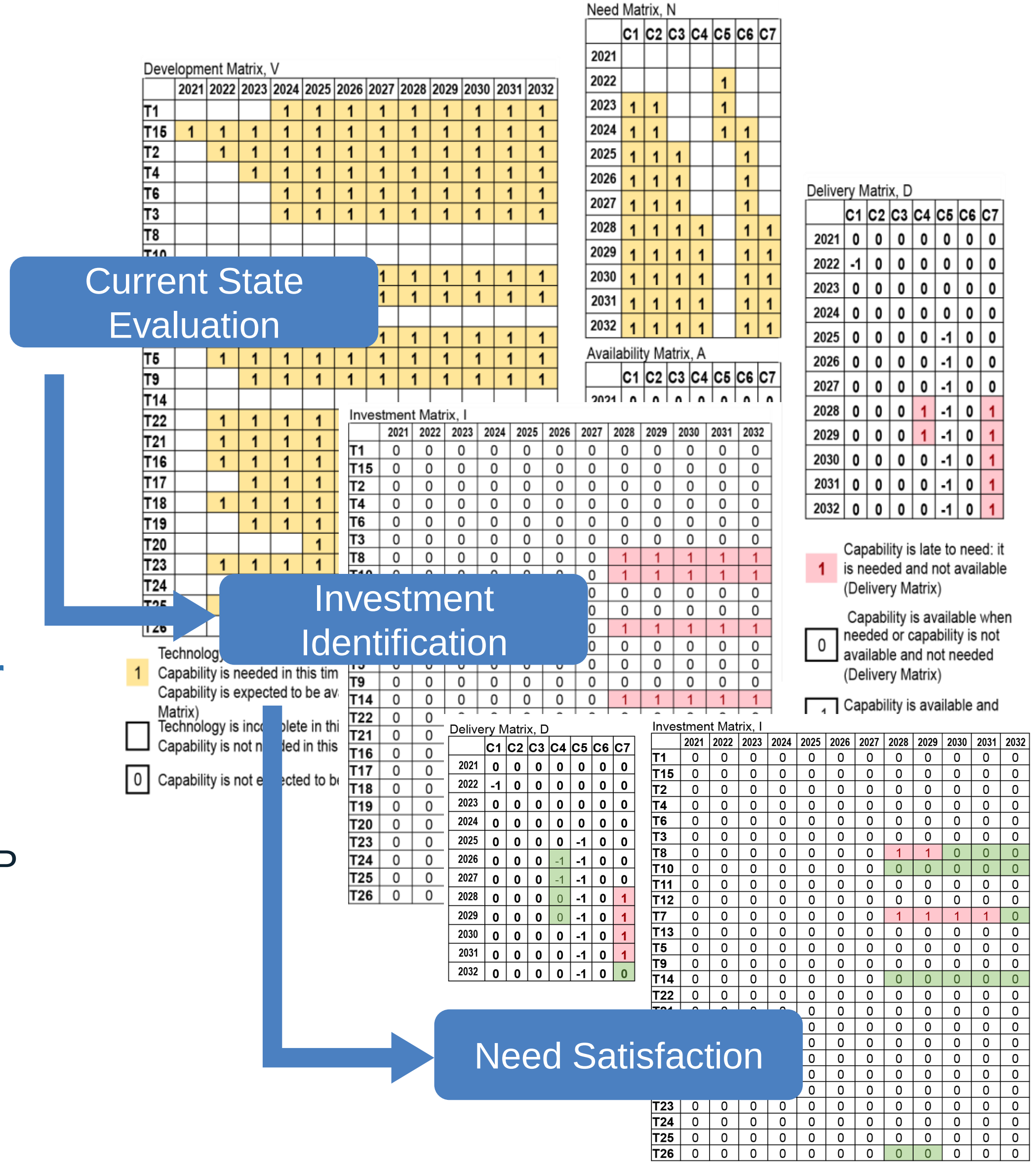
- In iterative development, it is often difficult to identify the long term impact of short term decisions[9], resulting in technical debt
  - Selection of 'easiest' components to deliver value makes a system difficult to change[8]
  - Schedule pressures make fixed time iterations especially susceptible to technical debt[10]
  - Tradeoff exists between early value creation and potential for future rework[11]
- Methods for managing technical debt focus on identification of existing technical debt and repayment during future iterations
  - **Do not provide proactive assessments of the ability to meet the stakeholders' needs at the time the decision is made**
- List, Evaluate, Achieve, Procure (LEAP) process
  - Process is designed to identify technologies that contribute to capability deliveries in both the functional and temporal dimensions and to predict the ability to satisfy stakeholder needs based on technology development timelines



Stakeholder Satisfaction

Related Work

- Release planning methods try to optimize the features that are included in each release[10]
  - Weighted shortest job first[12]
  - Model uncertainty with fixed time release cycles[10]
  - Minimization of total cost and maximization of early value[13]
  - Optimal release to repay technical debt[14]
- These models do not
  - Propagate delays induced by technical debt into the rest of the development cycle
  - Enable proactive technical debt identification
  - Map value delivery directly to the stakeholder's needs
- The LEAP process provides a method to **map the temporal and functional delivery of value to the satisfaction of stakeholder needs** and to identify technologies with a high potential for introducing technical debt into the system
- The research is divided into two related presentations at IS 2023
  - This presentation provides the details on the mechanics of the LEAP process
  - *LEAPing Ahead: The Space Development Agency's Method for Planning for the Future*[15] provides an example application of the process at the Space Development Agency to identify critical technologies in need of accelerated development



Figures from Kleinwaks, et al[15]

The LEAP Process

List

Decompose needs into strategic and tactical capabilities

Define enabling technologies

Evaluate

Determine the capability need dates and technology development timelines.

Mathematically determine if capabilities meet need dates

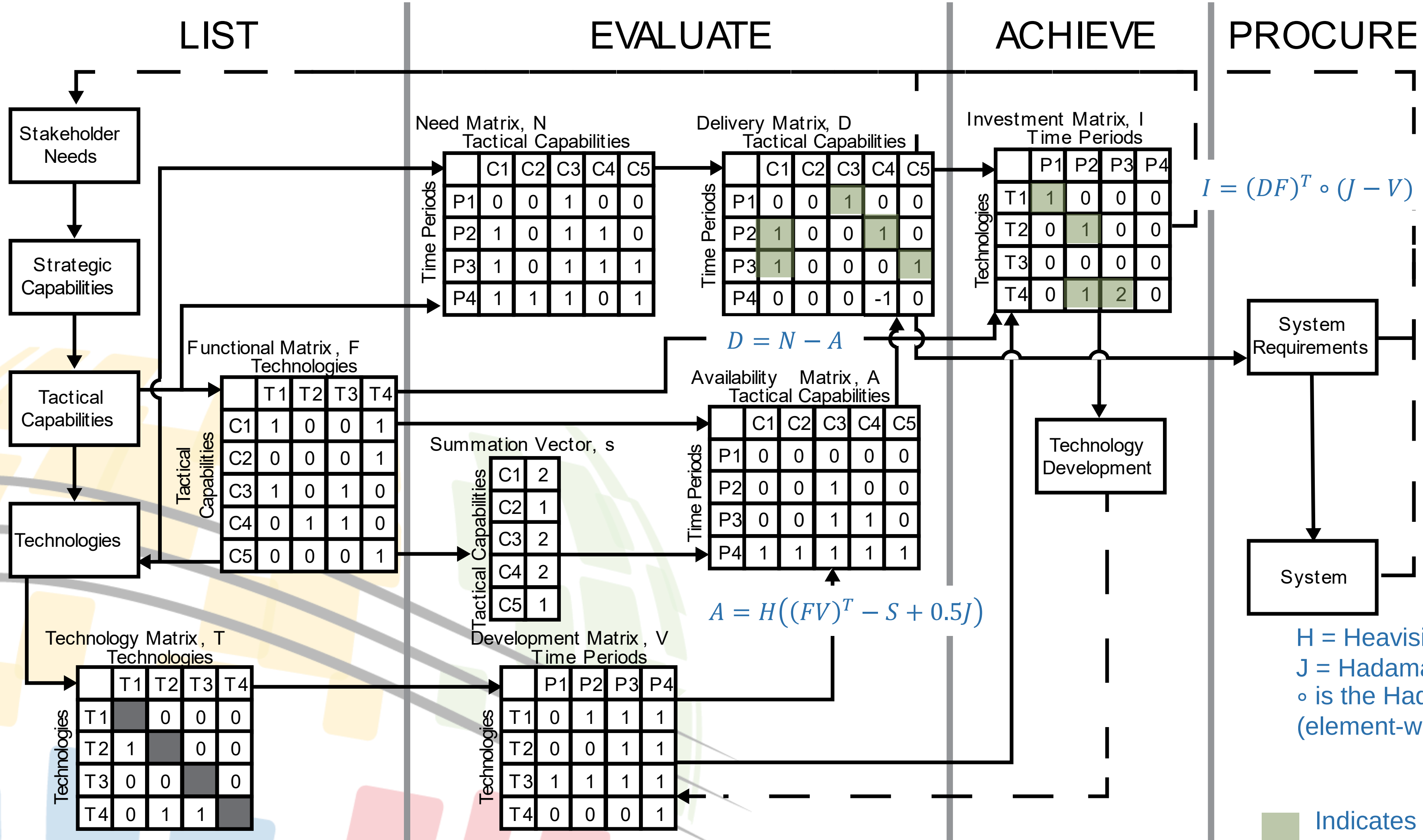
Achieve

Identify technology investments that produce largest return on investment

Procure

Release a system that provides one or more tactical or strategic capabilities

LEAP Process Overview



Need Matrix, N
Tactical Capabilities
Time Periods

	C1	C2	C3	C4	C5
P1	0	0	1	0	0
P2	1	0	1	1	0
P3	1	0	1	1	1
P4	1	1	1	0	1

Delivery Matrix, D
Tactical Capabilities
Time Periods

	C1	C2	C3	C4	C5
P1	0	0	1	0	0
P2	1	0	0	1	0
P3	1	0	0	0	1
P4	0	0	0	-1	0

Investment Matrix, I
Time Periods
Technologies

	P1	P2	P3	P4
T1	1	0	0	0
T2	0	1	0	0
T3	0	0	0	0
T4	0	1	2	0

$$I = (DF)^T \circ (J - V)$$

$$D = N - A$$

Functional Matrix, F
Technologies
Tactical Capabilities

	T1	T2	T3	T4
C1	1	0	0	1
C2	0	0	0	1
C3	1	0	1	0
C4	0	1	1	0
C5	0	0	0	1

Availability Matrix, A
Tactical Capabilities
Time Periods

	C1	C2	C3	C4	C5
P1	0	0	0	0	0
P2	0	0	1	0	0
P3	0	0	1	1	0
P4	1	1	1	1	1

Summation Vector, s
Tactical Capabilities

C1	2
C2	1
C3	2
C4	2
C5	1

$$A = H\left((FV)^T - S + 0.5J\right)$$

Technology Matrix, T
Technologies
Technologies

	T1	T2	T3	T4
T1		0	0	0
T2	1		0	0
T3	0	0		0
T4	0	1	1	

Development Matrix, V
Time Periods
Technologies

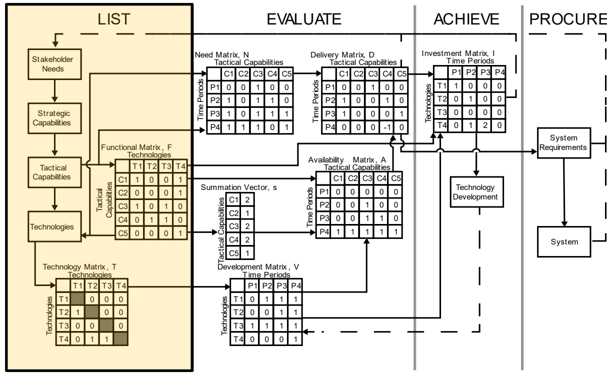
	P1	P2	P3	P4
T1	0	1	1	1
T2	0	0	1	1
T3	1	1	1	1
T4	0	0	0	1

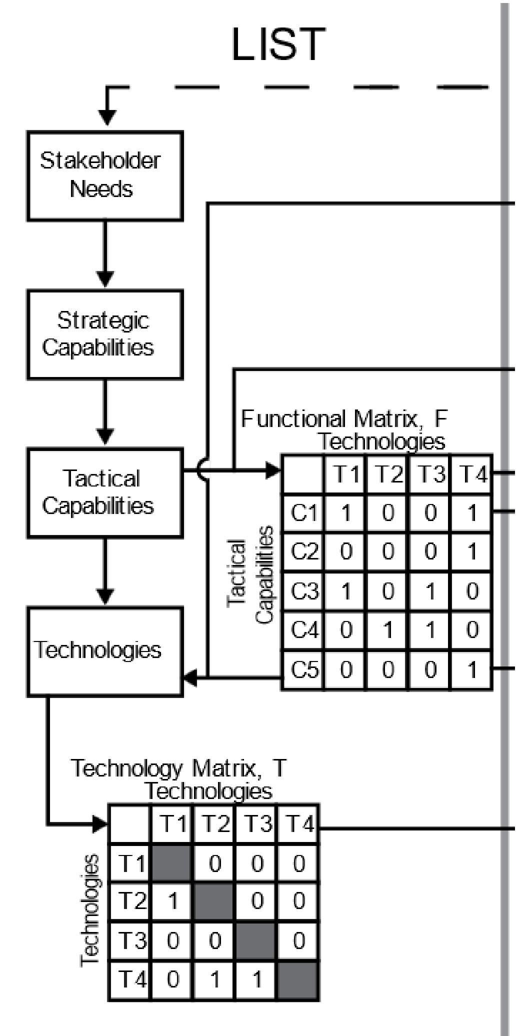
H = Heaviside function
J = Hadamard identity matrix (all ones)
∘ is the Hadamard product operator (element-wise multiplication)

🟩 Indicates capability/technology that is late to need

List Phase

- **List** establishes the system definition
  - Identification of the capabilities from the stakeholder needs
  - Functional breakdown of the capabilities into technologies
  - Sequencing of technology development
- Two major products:
  - Functional Matrix (F): maps the capabilities to the technologies that support them
  - Technology Matrix (T): design structure matrix that identifies dependencies between the technologies





Functional Matrix, F

Technologies

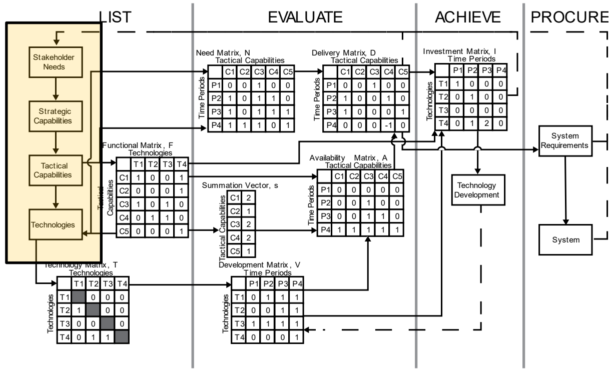
Tactical Capabilities	T1	T2	T3	T4
C1	1	0	0	1
C2	0	0	0	1
C3	1	0	1	0
C4	0	1	1	0
C5	0	0	0	1

Technology Matrix, T

Technologies

Technologies	T1	T2	T3	T4
T1	▨	0	0	0
T2	1	▨	0	0
T3	0	0	▨	0
T4	0	1	1	▨

List: Need Decomposition



Need Matrix, N

	C1	C2	C3	C4	C5
P1	0	0	1	0	0
P2	1	0	1	1	0
P3	1	0	1	1	1
P4	1	1	1	0	1

Delivery Matrix, D

	C1	C2	C3	C4	C5
P1	0	0	1	0	0
P2	1	0	0	1	0
P3	1	0	0	0	1
P4	0	0	0	-1	0

Investment Matrix, I

	P1	P2	P3	P4
T1	1	0	0	0
T2	0	1	0	0
T3	0	0	0	0
T4	0	1	2	0

Functional Matrix, F

	T1	T2	T3	T4
C1	1	0	0	1
C2	0	0	0	1
C3	1	0	1	0
C4	0	1	1	0
C5	0	0	0	1

Summation Vector, s

C1	2
C2	1
C3	2
C4	2
C5	1

Availability Matrix, A

	C1	C2	C3	C4	C5
P1	0	0	0	0	0
P2	0	0	1	0	0
P3	0	0	1	1	0
P4	1	1	1	1	1

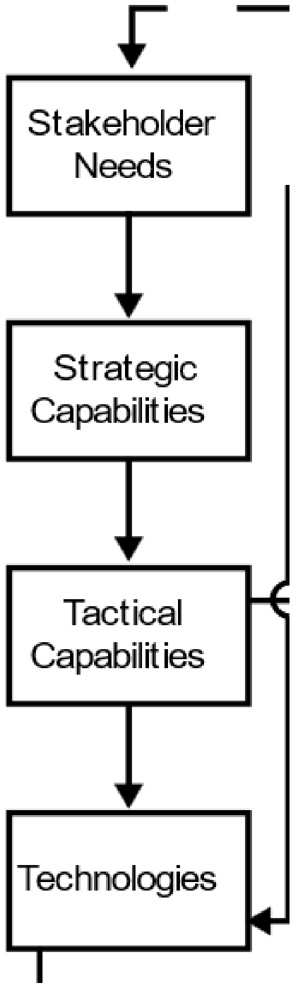
Technology Matrix, T

	T1	T2	T3	T4
T1	▨	0	0	0
T2	1	▨	0	0
T3	0	0	▨	0
T4	0	1	1	▨

Development Matrix, V

	P1	P2	P3	P4
T1	0	1	1	1
T2	0	0	1	1
T3	1	1	1	1
T4	0	0	0	1

- Stakeholder needs decompose into:
  - Strategic Capabilities: ability of a system to perform in its intended environment and meet the intent of its users
  - Tactical Capabilities: part of a strategic capability that delivers value to the user
  - Technology: Methods and devices (hardware or software) resulting from the practical application of knowledge



Stakeholder Needs

Strategic Capabilities

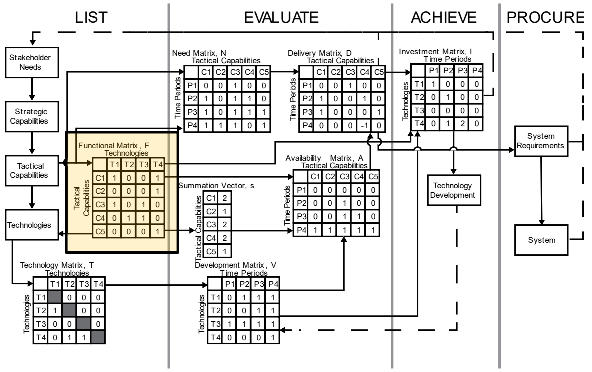
Tactical Capabilities

Technologies

Tactical Capability is ***required*** at a specific point in time

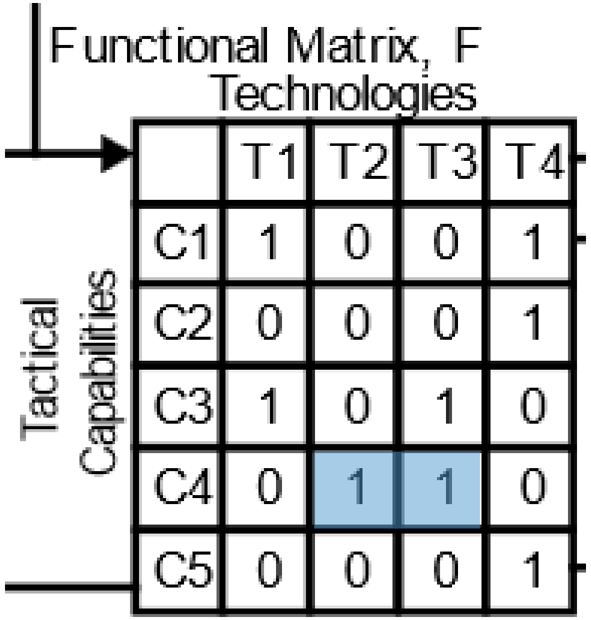
Technology is ***delivered*** at a specific point in time

List: Functional Matrix



- Maps Tactical Capabilities to the Technologies that support them
  - 1 indicates that a capability is supported by the technology
  - 0 indicates that the capability is not dependent on the technology

Throughout this presentation, we will trace tactical capability C4 through the LEAP process


Functional Matrix, F

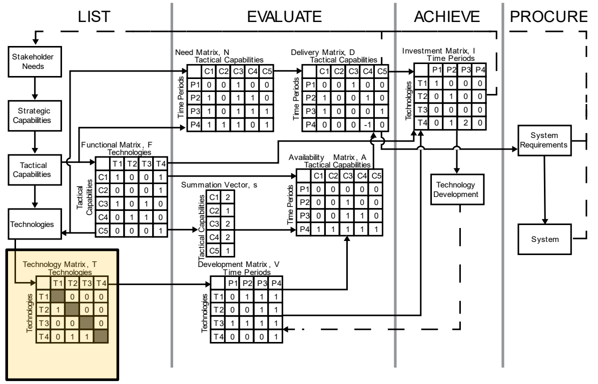
Technologies

Tactical Capabilities

	T1	T2	T3	T4
C1	1	0	0	1
C2	0	0	0	1
C3	1	0	1	0
C4	0	1	1	0
C5	0	0	0	1

C4 depends on T2 and T3

List: Technology Matrix



Need Matrix, N
Tactical Capabilities

Time Periods	C1	C2	C3	C4	C5
P1	0	0	1	0	0
P2	1	0	1	1	0
P3	1	0	1	1	1
P4	1	1	1	0	1

Delivery Matrix, D
Tactical Capabilities

Time Periods	C1	C2	C3	C4	C5
P1	0	0	1	0	0
P2	1	0	0	1	0
P3	1	0	0	0	1
P4	0	0	0	-1	0

Investment Matrix, I
Time Periods

Technologies	P1	P2	P3	P4
T1	1	0	0	0
T2	0	1	0	0
T3	0	0	0	0
T4	0	1	2	0

Functional Matrix, F
Technologies

Tactical Capabilities	T1	T2	T3	T4
C1	1	0	0	1
C2	0	0	0	1
C3	1	0	1	0
C4	0	1	1	0
C5	0	0	0	1

Summation Vector, s

Tactical Capabilities	
C1	2
C2	1
C3	2
C4	2
C5	1

Availability Matrix, A
Tactical Capabilities

Time Periods	C1	C2	C3	C4	C5
P1	0	0	0	0	0
P2	0	0	1	0	0
P3	0	0	1	1	0
P4	1	1	1	1	1

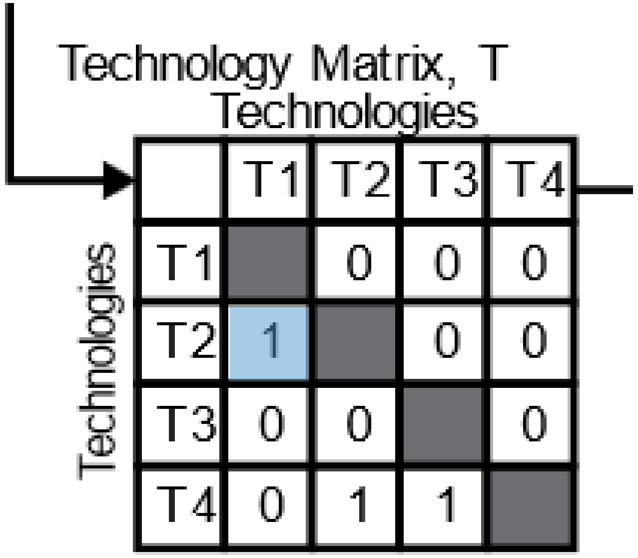
Technology Matrix, T
Technologies

Technologies	T1	T2	T3	T4
T1		0	0	0
T2	1		0	0
T3	0	0		0
T4	0	1	1	

Development Matrix, V
Time Periods

Technologies	P1	P2	P3	P4
T1	0	1	1	1
T2	0	0	1	1
T3	1	1	1	1
T4	0	0	0	1

- Implemented as a design structure matrix
- Shows interdependencies between technologies
  - 1 indicates that the technology in the row is dependent upon the technology in the column
- Rows are arranged in rough chronological order[16]
  - Entries below the diagonal represent forward information flow: T1 flows information to T2, T2 depends on T1
  - Entries above the diagonal represent feedback mechanisms
- Can use standard DSM partitioning techniques to rearrange the matrix and to find optimal work flows[16]



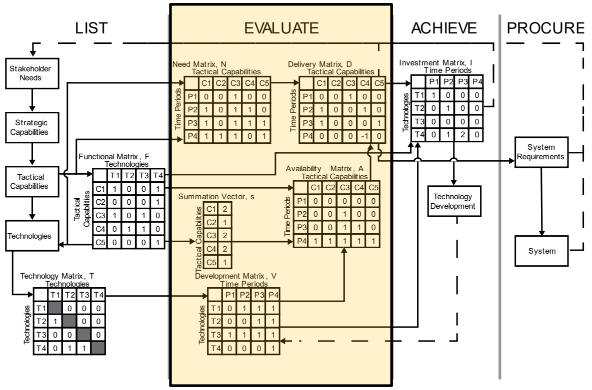
Technology Matrix, T
Technologies

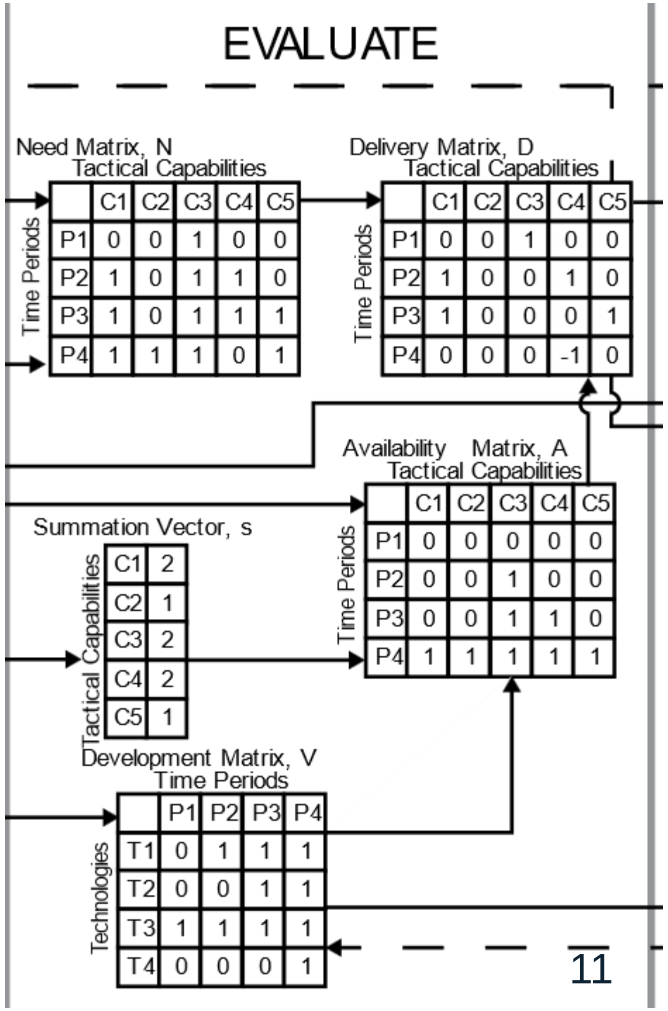
Technologies	T1	T2	T3	T4
T1		0	0	0
T2	1		0	0
T3	0	0		0
T4	0	1	1	

T2 depends on T1
T3 has no dependencies

Evaluate Phase

- **Evaluate** assesses the capabilities and the technologies
- Four products:
  - – Input matrices:
    - Development Matrix (V)
    - Need Matrix (N)
  - – Calculated matrices
    - Availability Matrix (A)
    - Delivery Matrix (D)





Need Matrix, N
Tactical Capabilities
Time Periods

	C1	C2	C3	C4	C5
P1	0	0	1	0	0
P2	1	0	1	1	0
P3	1	0	1	1	1
P4	1	1	1	0	1

Delivery Matrix, D
Tactical Capabilities
Time Periods

	C1	C2	C3	C4	C5
P1	0	0	1	0	0
P2	1	0	0	1	0
P3	1	0	0	0	1
P4	0	0	0	-1	0

Availability Matrix, A
Tactical Capabilities
Time Periods

	C1	C2	C3	C4	C5
P1	0	0	0	0	0
P2	0	0	1	0	0
P3	0	0	1	1	0
P4	1	1	1	1	1

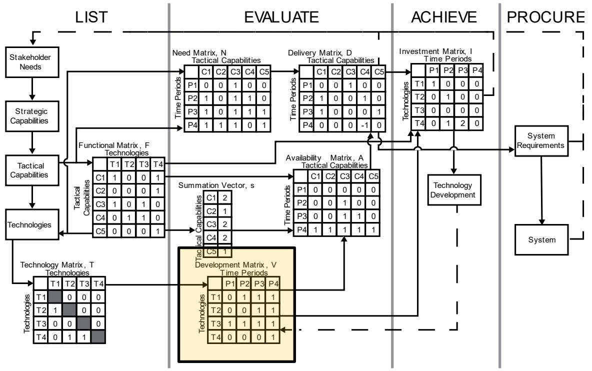
Summation Vector, s
Tactical Capabilities

C1	2
C2	1
C3	2
C4	2
C5	1

Development Matrix, V
Time Periods
Technologies

	P1	P2	P3	P4
T1	0	1	1	1
T2	0	0	1	1
T3	1	1	1	1
T4	0	0	0	1

Evaluate: Development Matrix



Need Matrix, N
Tactical Capabilities

Time Periods	C1	C2	C3	C4	C5
P1	0	0	1	0	0
P2	1	0	1	1	0
P3	1	0	1	1	1
P4	1	1	1	0	1

Delivery Matrix, D
Tactical Capabilities

Time Periods	C1	C2	C3	C4	C5
P1	0	0	1	0	0
P2	1	0	0	1	0
P3	1	0	0	0	1
P4	0	0	0	-1	0

Investment Matrix, I
Time Periods

Technologies	P1	P2	P3	P4
T1	1	0	0	0
T2	0	1	0	0
T3	0	0	0	0
T4	0	1	2	0

Functional Matrix, F
Technologies

Tactical Capabilities	T1	T2	T3	T4
C1	1	0	0	1
C2	0	0	0	1
C3	1	0	1	0
C4	0	1	1	0
C5	0	0	0	1

Summation Vector, s

Tactical Capabilities	
C1	2
C2	1
C3	2
C4	2
C5	1

Availability Matrix, A
Tactical Capabilities

Time Periods	C1	C2	C3	C4	C5
P1	0	0	0	0	0
P2	0	0	1	0	0
P3	0	0	1	1	0
P4	1	1	1	1	1

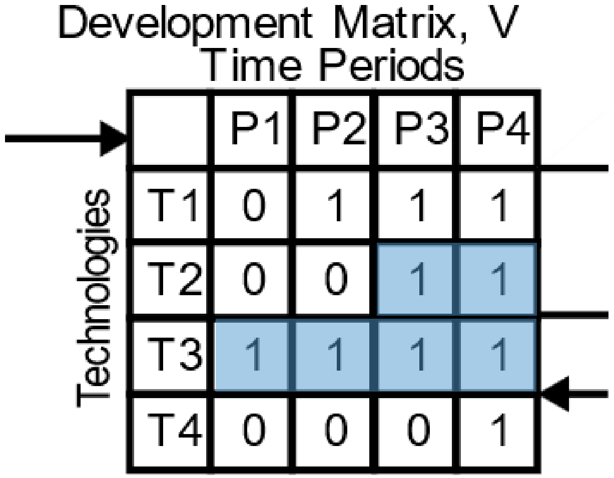
Technology Matrix, T
Technologies

Technologies	T1	T2	T3	T4
T1	■	0	0	0
T2	1	■	0	0
T3	0	0	■	0
T4	0	1	1	■

Development Matrix, V
Time Periods

Technologies	P1	P2	P3	P4
T1	0	1	1	1
T2	0	0	1	1
T3	1	1	1	1
T4	0	0	0	1

- Development Matrix
  - maps the technologies to the time periods where they are expected to be ready
  - 1 is entered in all time periods that the technology is developed
  - 0 indicates that the technology is not developed in that time period
  - Binary value: technology either is or is not ready in the time period

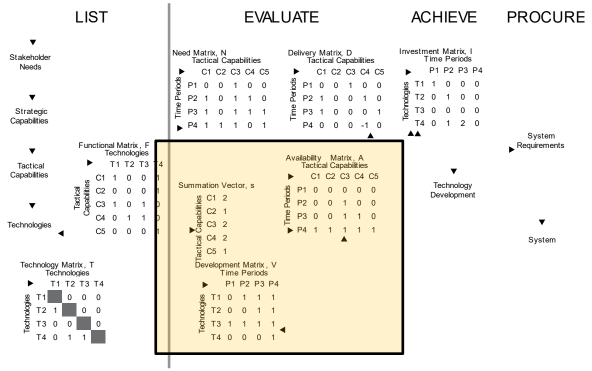


Development Matrix, V
Time Periods

Technologies	P1	P2	P3	P4
T1	0	1	1	1
T2	0	0	1	1
T3	1	1	1	1
T4	0	0	0	1

T2 will be developed in P3
T3 will be developed in P1

Evaluate: Availability Matrix



Need Matrix, N
Tactical Capabilities

Time Periods	C1	C2	C3	C4	C5
P1	0	0	1	0	0
P2	1	0	1	1	0
P3	1	0	1	1	1
P4	1	1	1	0	1

Delivery Matrix, D
Tactical Capabilities

Time Periods	C1	C2	C3	C4	C5
P1	0	0	1	0	0
P2	1	0	0	1	0
P3	1	0	0	0	1
P4	0	0	0	-1	0

Investment Matrix, I
Time Periods

Technologies	P1	P2	P3	P4
T1	1	0	0	0
T2	0	1	0	0
T3	0	0	0	0
T4	0	1	2	0

Functional Matrix, F
Technologies

Tactical Capabilities	T1	T2	T3	T4
C1	1	0	0	1
C2	0	0	0	1
C3	1	0	1	0
C4	0	1	1	0
C5	0	0	0	1

Summation Vector, s

Tactical Capabilities	
C1	2
C2	1
C3	2
C4	2
C5	1

Availability Matrix, A
Tactical Capabilities

Time Periods	C1	C2	C3	C4	C5
P1	0	0	0	0	0
P2	0	0	1	0	0
P3	0	0	1	1	0
P4	1	1	1	1	1

Technology Matrix, T
Technologies

Technologies	T1	T2	T3	T4
T1		0	0	0
T2	1		0	0
T3	0	0		0
T4	0	1	1	

Development Matrix, V
Time Periods

Technologies	P1	P2	P3	P4
T1	0	1	1	1
T2	0	0	1	1
T3	1	1	1	1
T4	0	0	0	1

- Availability Matrix identifies in which time period each capability is expected to be available
- Multiply the Functional Matrix (F) by the Development Matrix (V)
  - 1 in the **Functional Matrix** cell: technology (the column) is **required** by a capability (the row)
  - 1 in the **Development Matrix** cell: technology (the row) is **available** in a time period (the column)
  - Matrix multiplication (F*V) gives the count of the number of technologies that support a capability and are available in the time period
- Subtract the number of required technologies per capability
  - Summation Matrix (S): how many technologies are required for each capability
    - Calculated by summing the columns of F
- Apply the Heaviside function
  - Sets values to either 0 (capability is not available) or 1 (capability is available) for each time period

Time period 0

In time period 0, technology 2 is developed (1) and technology 0 is not (0)

$$F[2,:]\cdot V[:,0] = [1 \quad 0 \quad 1 \quad 0]\cdot\begin{bmatrix}0\\0\\1\\0\end{bmatrix} = 1$$

Capability 2

Technologies 0 and 2 support capability 2

1 technology that supports capability 2 is available in time period 0

Available number of technologies that support capability in each time period

Transpose to align matrix dimensions

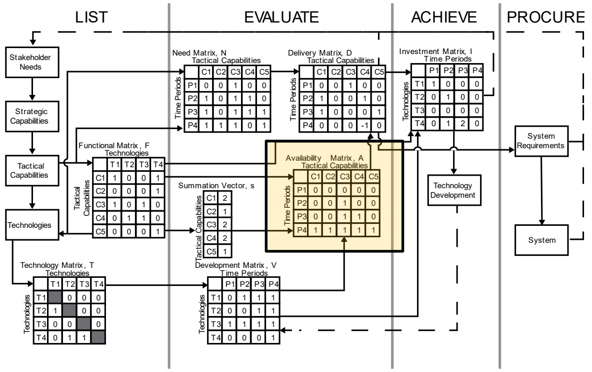
$$A = H\left((FV)^T - S + 0.5J\right)$$

Heaviside Function

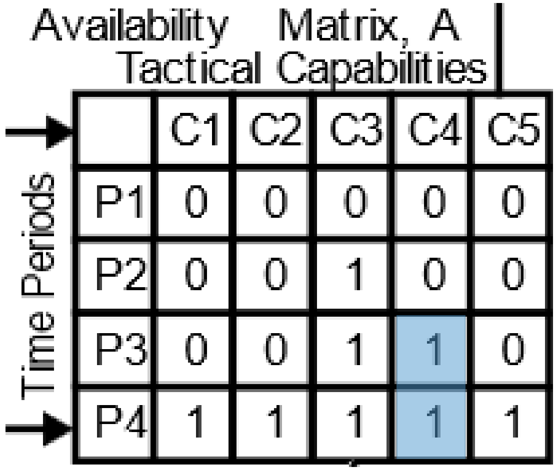
Total number of technologies required per capability

0.5J offset for application of Heaviside function

Evaluate: Availability Matrix



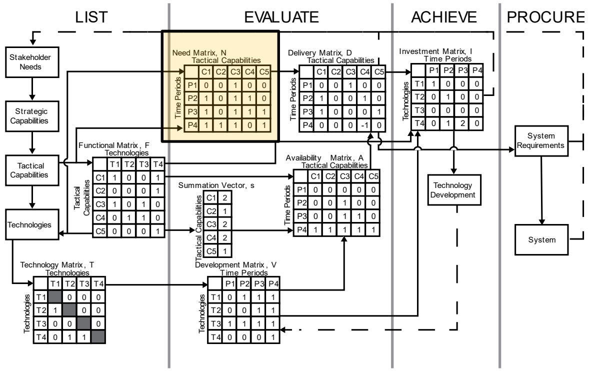
- Availability Matrix shows which capabilities are expected to be available in each time period
  - 1 indicates that the capability is expected to be available
  - 0 indicates that the capability is not expected to be available
- Availability Matrix considers the ability to deliver capabilities based on the functional breakdown and temporal delivery of technologies
- It is independent of the stakeholder's needs



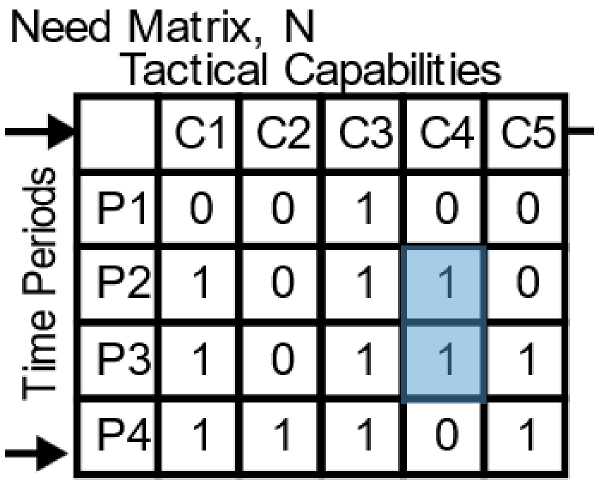
	C1	C2	C3	C4	C5
P1	0	0	0	0	0
P2	0	0	1	0	0
P3	0	0	1	1	0
P4	1	1	1	1	1

C4 is available in P3

Evaluate: Need Matrix



- Need Matrix: maps the tactical capabilities to the time periods in which they are needed
  - Capabilities may not be needed in all time periods
  - Need for a capability may change over time
  - 1 indicates that the capability is needed in that time period
  - 0 indicates that the capability is not needed in that time period



Need Matrix, N

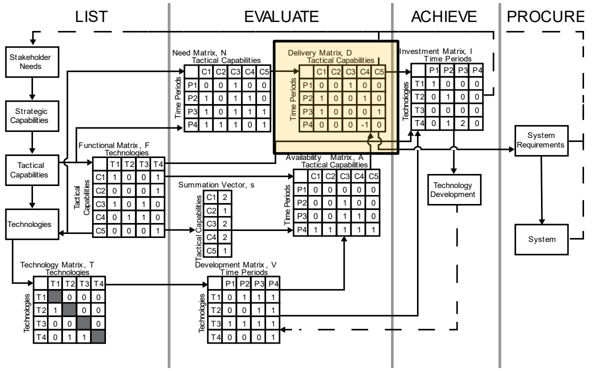
Tactical Capabilities

Time Periods

	C1	C2	C3	C4	C5
P1	0	0	1	0	0
P2	1	0	1	1	0
P3	1	0	1	1	1
P4	1	1	1	0	1

C4 is needed in P2 and P3

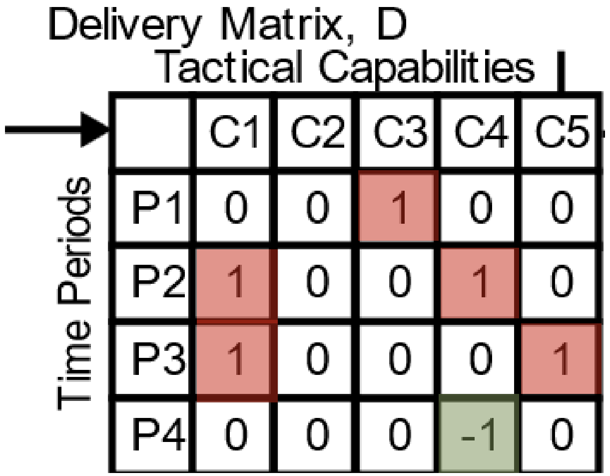
Evaluate: Delivery Matrix



- Delivery Matrix: identifies which capabilities are delivered late to need
- Subtract the Availability Matrix (A) from the Need Matrix (N)

$$D = N - A$$

  - Both A and N contain 1 or 0 and have time periods in rows and capabilities in columns.
- Delivery Matrix provides information about the timeliness of capabilities compared to the needs
  - On time delivery: 0 indicates that the capability is **needed and available** or is **not needed and not available** in the time period
  - Late delivery: one 1 indicates that the capability is **needed and not available** in the time period
  - Early delivery: -1 indicates that the capability is **available but not needed** in the time period



Delivery Matrix, D

Tactical Capabilities

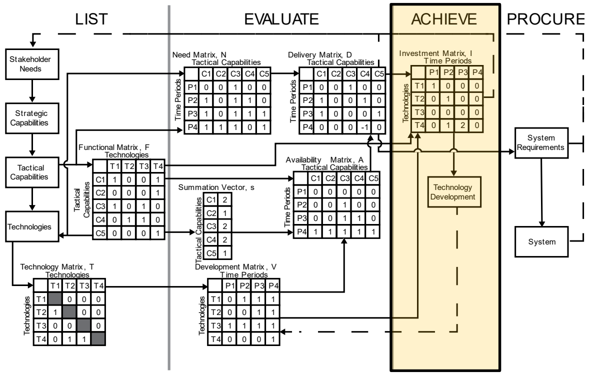
Time Periods

	C1	C2	C3	C4	C5
P1	0	0	1	0	0
P2	1	0	0	1	0
P3	1	0	0	0	1
P4	0	0	0	-1	0

☐ On time 🟩 Early or not needed 🟥 Late

C4 is late in P2 and available but not needed in P4

Achieve Phase



Need Matrix, N

Tactical Capabilities

Time Periods	C1	C2	C3	C4	C5
P1	0	0	1	0	0
P2	1	0	1	1	0
P3	1	0	1	1	1
P4	1	1	1	0	1

Delivery Matrix, D

Tactical Capabilities

Time Periods	C1	C2	C3	C4	C5
P1	0	0	1	0	0
P2	1	0	0	1	0
P3	1	0	0	0	1
P4	0	0	0	-1	0

Investment Matrix, I

Time Periods

Technologies	P1	P2	P3	P4
T1	1	0	0	0
T2	0	1	0	0
T3	0	0	0	0
T4	0	1	2	0

Functional Matrix, F

Technologies

Tactical Capabilities	T1	T2	T3	T4
C1	1	0	0	1
C2	0	0	0	1
C3	1	0	1	0
C4	0	1	1	0
C5	0	0	0	1

Summation Vector, s

Tactical Capabilities	
C1	2
C2	1
C3	2
C4	2
C5	1

Availability Matrix, A

Tactical Capabilities

Time Periods	C1	C2	C3	C4	C5
P1	0	0	0	0	0
P2	0	0	1	0	0
P3	0	0	1	1	0
P4	1	1	1	1	1

Technology Matrix, T

Technologies

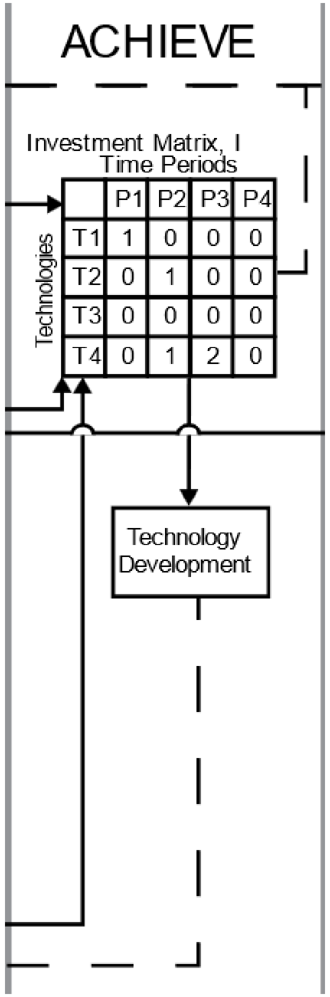
Technologies	T1	T2	T3	T4
T1	■	0	0	0
T2	1	■	0	0
T3	0	0	■	0
T4	0	1	1	■

Development Matrix, V

Time Periods

Technologies	P1	P2	P3	P4
T1	0	1	1	1
T2	0	0	1	1
T3	1	1	1	1
T4	0	0	0	1

- **Achieve** determines where the largest technical debt impacts are and where the system developer should invest to reduce those impacts
- One product:
  - Investment Matrix

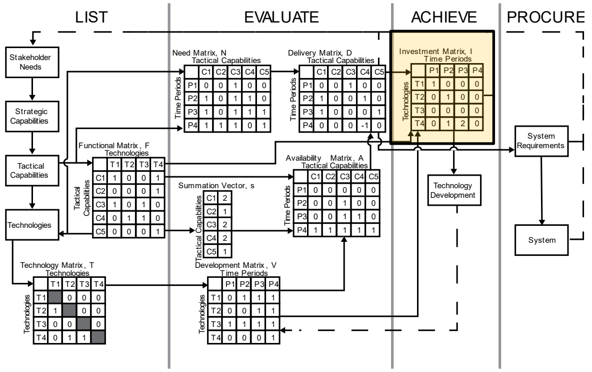


Investment Matrix, I

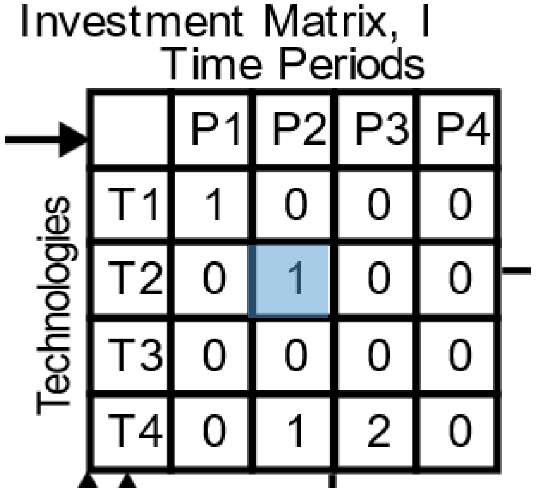
Time Periods

Technologies	P1	P2	P3	P4
T1	1	0	0	0
T2	0	1	0	0
T3	0	0	0	0
T4	0	1	2	0

Achieve: Investment Matrix



- Investment Matrix: maps technologies to the negative impact they have in each time period. Larger numbers indicate:
  - Larger impact due to the technology not being available
  - Technologies that have more benefit if their development timelines can be reduced or if the probability of success can be increased
- Matrix multiplication (D*F) determines the number of late capabilities influenced by each technology in each time period
- Investment Matrix produces the following results:
  - Positive values: the number of capabilities that the technology is causing to be late in that time period
  - Negative values: the technology is contributing to a capability that is delivered early
  - Zero values: the technology supports on-time capability delivery that time period



Investment Matrix, I
Time Periods

Technologies	P1	P2	P3	P4
T1	1	0	0	0
T2	0	1	0	0
T3	0	0	0	0
T4	0	1	2	0

T2 causes C4 to be late in P2

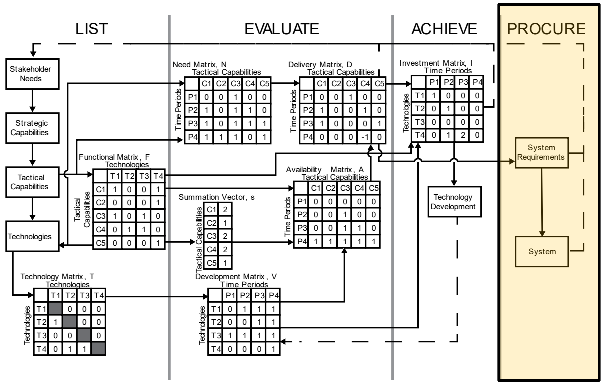
Transpose to align matrix dimensions

$$I = (DF)^T \circ (J - V)$$

Number of late capabilities influenced by each technology

Remove delivered technologies from the count

Procure Phase



Need Matrix, N
Tactical Capabilities / Time Periods

	C1	C2	C3	C4	C5
P1	0	0	1	0	0
P2	1	0	1	1	0
P3	1	0	1	1	1
P4	1	1	1	0	1

Delivery Matrix, D
Tactical Capabilities / Time Periods

	C1	C2	C3	C4	C5
P1	0	0	1	0	0
P2	1	0	0	1	0
P3	1	0	0	0	1
P4	0	0	0	-1	0

Investment Matrix, I
Time Periods / Technologies

	P1	P2	P3	P4
T1	1	0	0	0
T2	0	1	0	0
T3	0	0	0	0
T4	0	1	2	0

Functional Matrix, F
Technologies / Tactical Capabilities

	T1	T2	T3	T4
C1	1	0	0	1
C2	0	0	0	1
C3	1	0	1	0
C4	0	1	1	0
C5	0	0	0	1

Summation Vector, s
Tactical Capabilities

C1	2
C2	1
C3	2
C4	2
C5	1

Availability Matrix, A
Tactical Capabilities / Time Periods

	C1	C2	C3	C4	C5
P1	0	0	0	0	0
P2	0	0	1	0	0
P3	0	0	1	1	0
P4	1	1	1	1	1

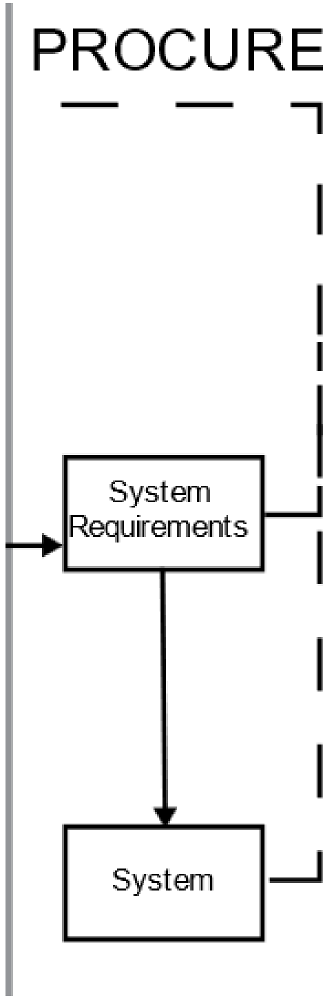
Technology Matrix, T
Technologies / Technologies

	T1	T2	T3	T4
T1	■	0	0	0
T2	1	■	0	0
T3	0	0	■	0
T4	0	1	1	■

Development Matrix, V
Time Periods / Technologies

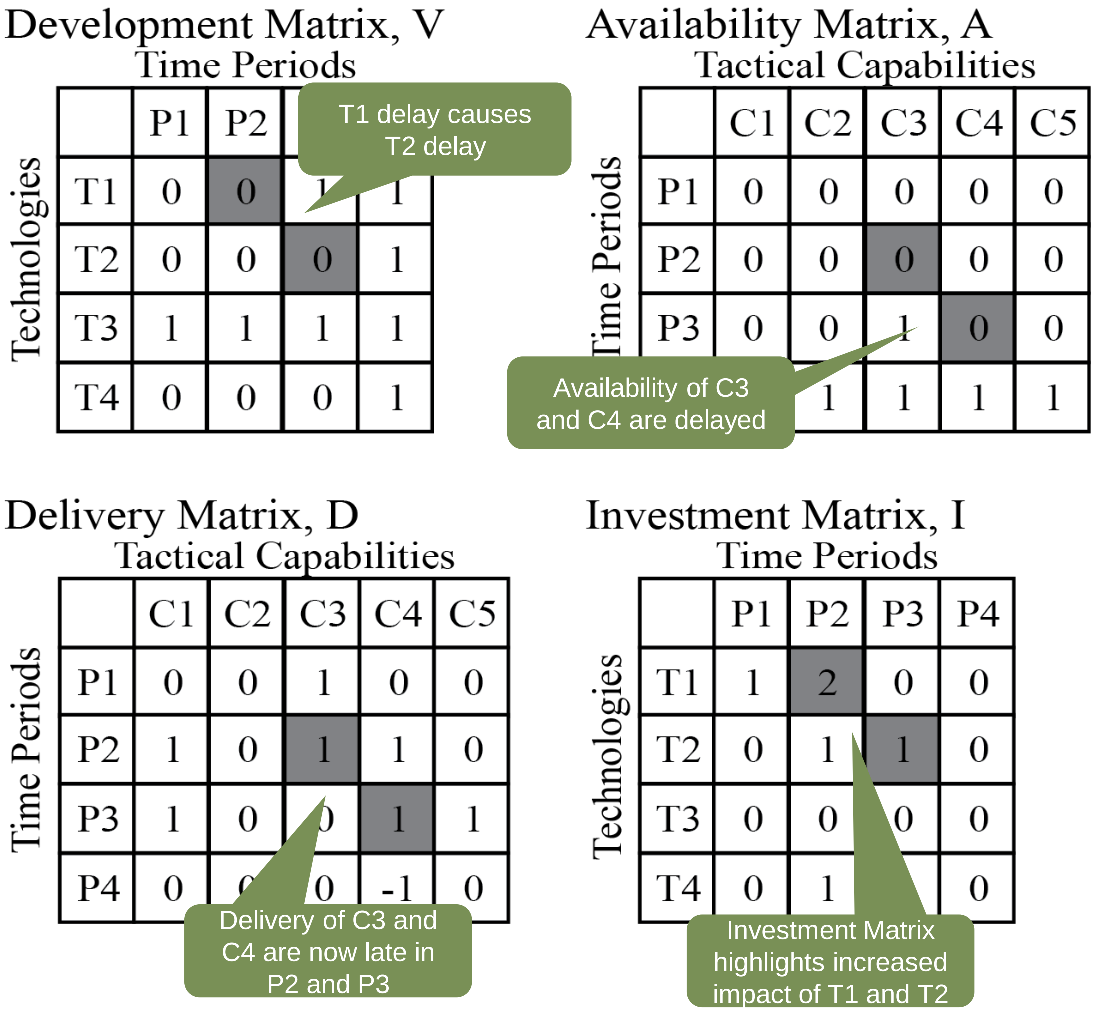
	P1	P2	P3	P4
T1	0	1	1	1
T2	0	0	1	1
T3	1	1	1	1
T4	0	0	0	1

- **Procure** determines technologies to include in the system procurement and feeds the development of the system
  - Iterative usage of the process allows the results of the procure phase to be passed back to the list stage
  - Procurement helps set development timelines for technologies included in the procurement



Assessing Technical Debt with LEAP

- Technical debt can appear due to a release not delivering planned capabilities or a technology not being ready in time for use in a release
- LEAP can indicate impacts of a late delivery through the Investment Matrix
    - Provides an estimate of technical debt by showing the larger impact of delays due to a single technology
- Consider case where development of technology T1 is delayed from P1 to P2
    - LEAP process shows the cascading impacts on later delivery of capabilities and therefore the late satisfaction of stakeholder needs



Development Matrix, V

Time Periods / Technologies

	P1	P2		
T1	0	0	1	1
T2	0	0	0	1
T3	1	1	1	1
T4	0	0	0	1

T1 delay causes T2 delay

Availability Matrix, A

Tactical Capabilities / Time Periods

	C1	C2	C3	C4	C5
P1	0	0	0	0	0
P2	0	0	0	0	0
P3	0	0	1	0	0
		1	1	1	1

Availability of C3 and C4 are delayed

Delivery Matrix, D

Tactical Capabilities / Time Periods

	C1	C2	C3	C4	C5
P1	0	0	1	0	0
P2	1	0	1	1	0
P3	1	0	0	1	1
P4	0	0	0	-1	0

Delivery of C3 and C4 are now late in P2 and P3

Investment Matrix, I

Time Periods / Technologies

	P1	P2	P3	P4
T1	1	2	0	0
T2	0	1	1	0
T3	0	0	0	0
T4	0	1		0

Investment Matrix highlights increased impact of T1 and T2

Changes are indicated in gray squares

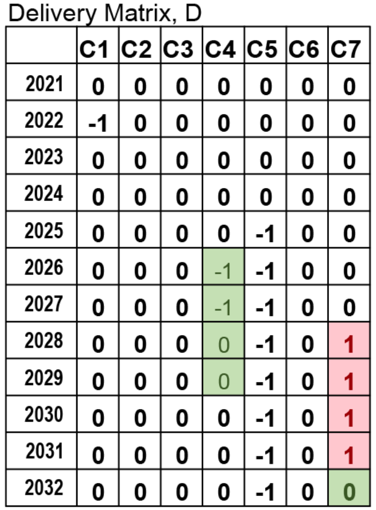
LEAP Use Cases

- Iterative development planning
  - LEAP can assess which capabilities are delivered late to need and identify which technologies should be invested in to reduce the risk of late delivery
  - Iterative use of LEAP with the procurement phase can improve each iteration
- Milestone achievement
  - By identifying the capabilities required at each milestone, LEAP can identify which technologies put meeting the milestones at risk
- Analysis of alternative investments
  - Using the Investment Matrix as a guide, the impact of accelerating or decelerating the pace of technology development on the end system can be identified
- Identification of sources and causes of technical debt
  - Technologies that are delivered late to need are potential sources of technical debt – the decision to not implement that technology can cause rework on the other parts of the system when it is finally implemented

For a more detailed use case see Kleinwaks, et al. *LEAPing Ahead – The Space Development Agency's Method for Planning for the Future*[16]

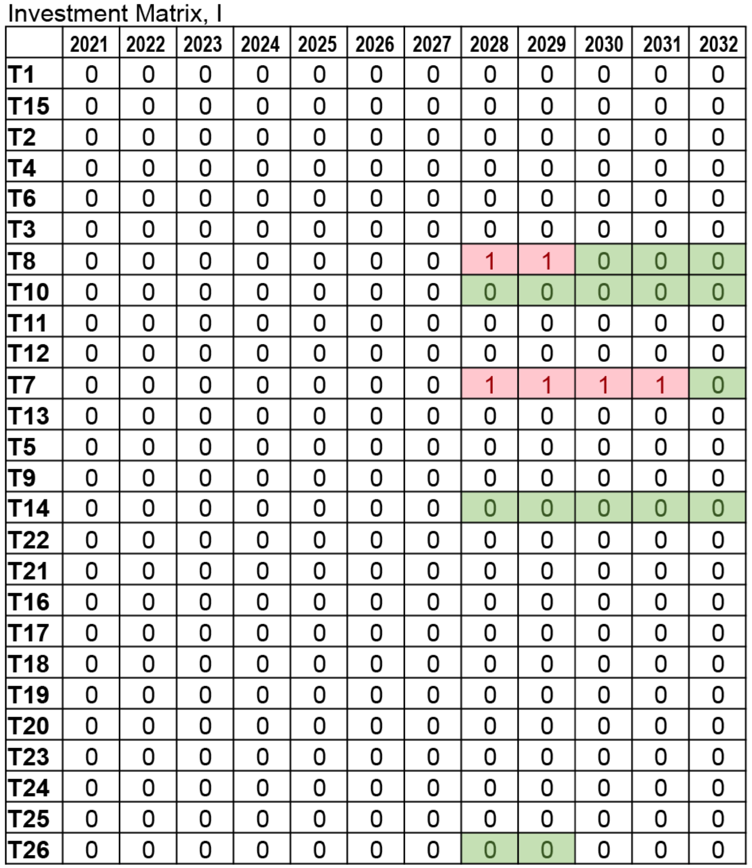
Discussion

- Usage at the Space Development Agency[15]
  - Enabled estimate of return on investment by showing accelerated delivery of capability to stakeholders
  - Identification of stakeholder needs that can be reevaluated in the next iteration
- Comparison to other iterative development processes
  - Agile: Time boxed iterations with variable scope, priority on early value delivery
  - Spiral: Evidence-based risk assessments guide development spirals
  - LEAP complements these processes by providing an objective assessment of on-time satisfaction of stakeholder needs
- Challenges with LEAP
  - Process currently deals in “absolutes”
  - Challenging to distinguish between capabilities and technologies



Delivery Matrix, D

	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



Investment Matrix, I

	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

Delivery and Investment Matrices in the Space Development Agency’s development of optical communication terminals, showing change in deliveries (green) due to technology investment. Figures from Kleinwaks, et al.[15]

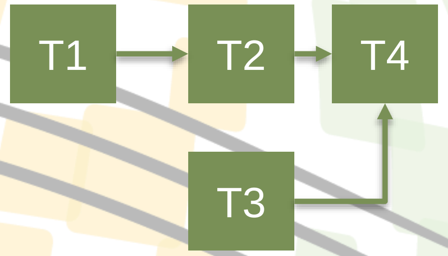
Future Work

- Add a prioritization matrix to enable ranking of needs
- Add probabilistic estimates of technology delivery
- Directly link the Technology Matrix and the Development Matrix
- Continued verification and validation

Technology Matrix

	T1	T2	T3	T4
T1	▨	0	0	0
T2	1	▨	0	0
T3	0	0	▨	0
T4	0	1	1	▨

Development Schedule



Probabilistic Development Matrix

	P1	P2	P3	P4
T1	0.47	1	1	1
T2	0.04	0.89	0.99	1
T3	0.45	1	1	1
T4	0	0.40	0.98	1

Probabilistic Delivery Matrix

	C1	C2	C3	C4	C5
P1	-1	-1	0.21	-1	-1
P2	0.40	-1	1	0.89	-1
P3	0.98	-1	1	0.99	0.98
P4	1	1	1	-1	1

Conclusion

- LEAP provides a mechanism to identify the future impact of technology development on system capability
  - Can guide the choices and schedules for investments in technology development
  - Can be used to assist in release planning to minimize NRE and risk on iteratively developed programs

Questions

Contact Information:

Howard Kleinwaks
Doctor of Engineering Candidate, Systems Engineering
Colorado State University
Modern Technology Solutions, Inc.
howard.kleinwaks@colostate.edu

Ann Batchelor
Colorado State University
ann.batchelor@colostate.edu

Dr. Thomas Bradley
Colorado State University
thomas.bradley@colostate.edu

References

1. Kleinwaks, H, A Batchelor, and T Bradley 2023, “Technical Debt in Systems Engineering – A Systematic Literature Review.” Submitted to *Systems Engineering*.
2. Kleinwaks, H, A Batchelor, and T Bradley 2023, “An Empirical Survey on the Prevalence of Technical Debt in Systems Engineering.” In Press, *33rd Annual INCOSE International Symposium*
3. Lane, JA, S Koolmanojwong, and BW Boehm 2013, ‘4.6.3 Affordable Systems: Balancing the Capability, Schedule, Flexibility, and Technical Debt Tradespace’, *INCOSE International Symposium*, pp. 1385-1399.
4. Schmidt, TS, S Weiss, and K Paetzold 2018, ‘Expected vs. Real Effects of Agile Development of Physical Products: Apportioning the Hype’, *DS 92: Proceedings of the DESIGN 2018 15th International Design Conference*, pp. 2121-2132.
5. Robiolo, G, E Scott, S Matalonga, and M Felderer 2019, ‘Technical Debt and Waste in Non-functional Requirements Documentation: An Exploratory Study’, *International Conference on Product-Focused Software Process Improvement*, pp. 220-235.
6. Koolmanojwong, S and JA Lane 2013, ‘Enablers and Inhibitors of Expediting Systems Engineering’, *Procedia Computer Science*, vol. 16, pp. 483-491.
7. Nord, RL, I Ozkaya, P Kruchten, and M Gonzalez-Rojas 2012, ‘In Search of a Metric for Managing Architectural Technical Debt’, *2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture,* pp. 91-100.
8. Boehm, BW, JA Lane, S Koolmanojwong, and R Turner 2014, *The Incremental Commitment Spiral Model: Principles and Practices for Successful Systems and Software*, Addison-Wesley, Upper Saddle River, NJ (US).
9. Forsberg, K and H Mooz 1995, ‘Application of the 'Vee' to Incremental and Evolutionary Development.’ *INCOSE International Symposium*, vol. 5, no. 1, pp. 848-855.
10. Oni, O and E Letier 2021, ‘Analyzing Uncertainty in Release Planning: A Method and Experiment for Fixed-Date Release Cycles’, *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 31, no. 2, pp. 1-39.
11. Tate, DM 2016, ‘Acquisition Cycle Time: Defining the Problem (Revised)’. Institute for Defense Analyses.
12. Scaled Agile, Inc. 2021, ‘Weighted Shortest Job First’, 10 February 2021, viewed 2 November 2022, <https://www.scaledagileframework.com/wsjf/>.
13. Sangwan, RS, A Negahban, RL Nord, and I Ozkaya 2020, ‘Optimization of Software Release Planning Considering Architectural Dependencies, Cost, and Value’, *IEEE Transactions on Software Engineering,* vol. 48, no. 4, pp. 1369-1384.
14. Schmid, K 2013, ‘A Formal Approach to Technical Debt Decision Making’, *Proceedings of the 9th international ACM Sigsoft conference on quality of software architectures*, pp. 153-162.
15. Kleinwaks, H, M Rich, M Butterfield, and F Turner 2023, “LEAPing Ahead – The Space Development Agency’s Method for Planning for the Future”, *33rd Annual INCOSE International Symposium.*
16. Browning, TR 1998, ‘Use of Dependency Structure Matrices for Product Development Cycle Time Reduction’, *Proceedings of the Fifth ISPE International Conference on Concurrent Engineering: Research and Applications*, Tokyo, Japan.

33rd Annual **INCOSE**
international symposium

hybrid event

Honolulu, HI, USA
July 15 - 20, 2023

www.incose.org/symp2023