A Prescriptive Adaptive Test Framework (PATFrame) for Unmanned and Autonomous Systems: A Collaboration Between MIT, USC, UT Arlington and Softstar Systems

Dr. Ricardo Valerdi
Massachusetts Institute of Technology

August 11, 2010

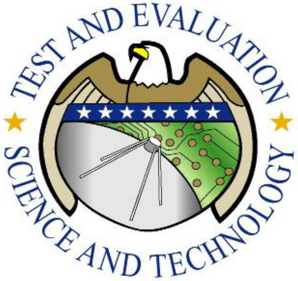
Outline

- PATFrame team
- The Challenge
- Test & evaluation decisions
- PATFrame features
- Use case

"Anything that gives us new knowledge gives us an opportunity to be more rational"

- Herb Simon

Sponsors

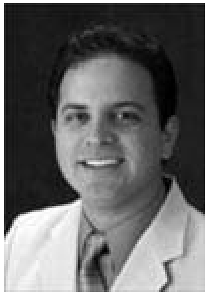


WHITE SANDS MISSILE RANGE
BIRTHPLACE OF AMERICA'S MISSILE & SPACE ACTIVITY

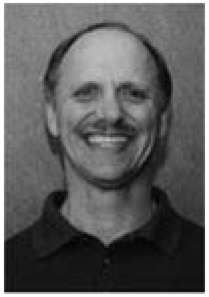
Transition Partners

PATFrame team

http://mit.edu/patframe



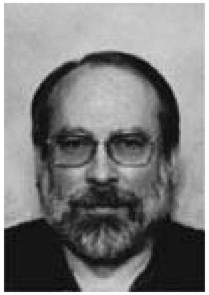
Valerdi



Kenley

Medvidovic

Ferreira

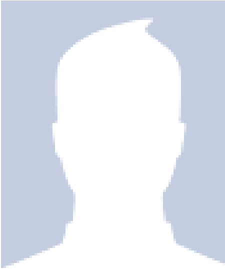


Ligett

Deonandan

Edwards

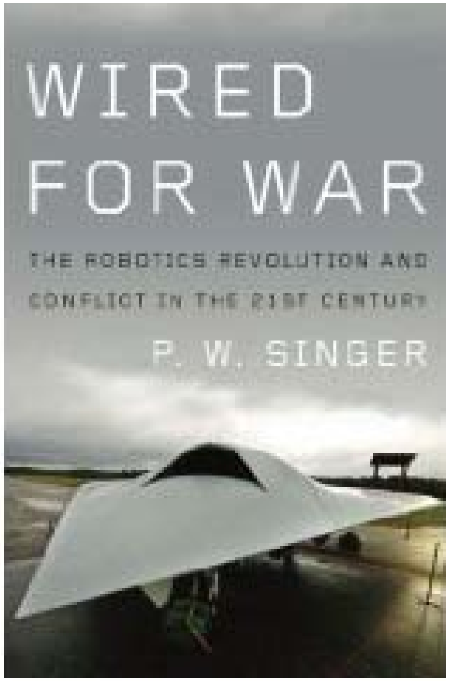
Hess



Tejeda

Cowart

The Challenge: Science Fiction to Reality



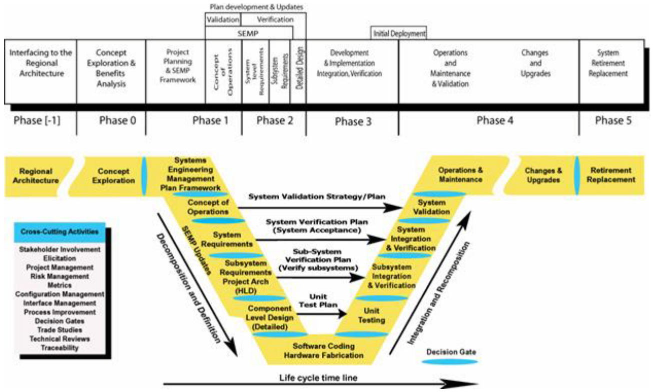
“You will be trying to apply international law written for the Second World War to *Star Trek* technology.”

Singer, P. W., *Wired For War: The Robotics Revolution and Conflict in the 21st Century* (Penguin, 2009)

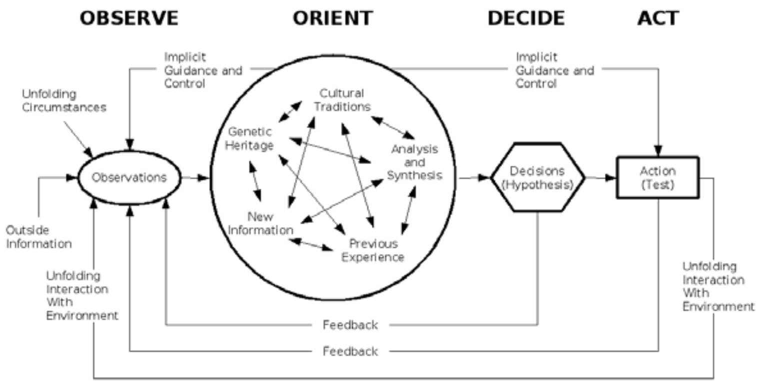
Science & Technology Background

- Current state of UAS T&E
  - UAS T&E is focused on single systems
  - One-shot planning for T&E and manual test strategy adaptation
  - Value-neutral approach to test prioritization
  - Autonomy not a key consideration
  - Function-based testing
  - Traditional acquisition process
  - Physics-based hardware-focused test prioritization and execution

- Future State of UAS T&E
  - Systems of systems introduce complex challenges
  - Accelerated & automated test planning based on rigorous methods
  - Value-based approach to test prioritization
  - Autonomy as a central challenge
  - Mission-based testing
  - Rapid acquisition process
  - Multi-attribute decision making to balance cost, risk and schedule of autonomous software-intensive systems of systems



Plan development & Updates
Validation
Verification
SEMP
Initial Deployment
Interfacing to the Regional Architecture
Concept Exploration & Benefits Analysis
Project Planning & SEMP Framework
Concept of Operations
System level Requirements
Subsystem Requirements
Detailed Design
Development & Implementation Integration, Verification
Operations and Maintenance & Validation
Changes and Upgrades
System Retirement Replacement
Phase [-1]
Phase 0
Phase 1
Phase 2
Phase 3
Phase 4
Phase 5
Regional Architecture
Concept Exploration
Systems Engineering Management Plan Framework
Concept of Operations
System Requirements
Subsystem Requirements Project Arch (HLD)
Component Level Design (Detailed)
Software Coding Hardware Fabrication
Unit Testing
Subsystem Integration & Verification
System Integration & Verification
System Validation
Operations & Maintenance
Changes & Upgrades
Retirement Replacement
System Validation Strategy/Plan
System Verification Plan (System Acceptance)
Sub-System Verification Plan (Verify subsystems)
Unit Test Plan
SEMP Updates
Decomposition and Definition
Integration and Recomposition
Cross-Cutting Activities
Stakeholder Involvement
Elicitation
Project Management
Risk Management
Metrics
Configuration Management
Interface Management
Process Improvement
Decision Gates
Trade Studies
Technical Reviews
Traceability
Decision Gate
Life cycle time line

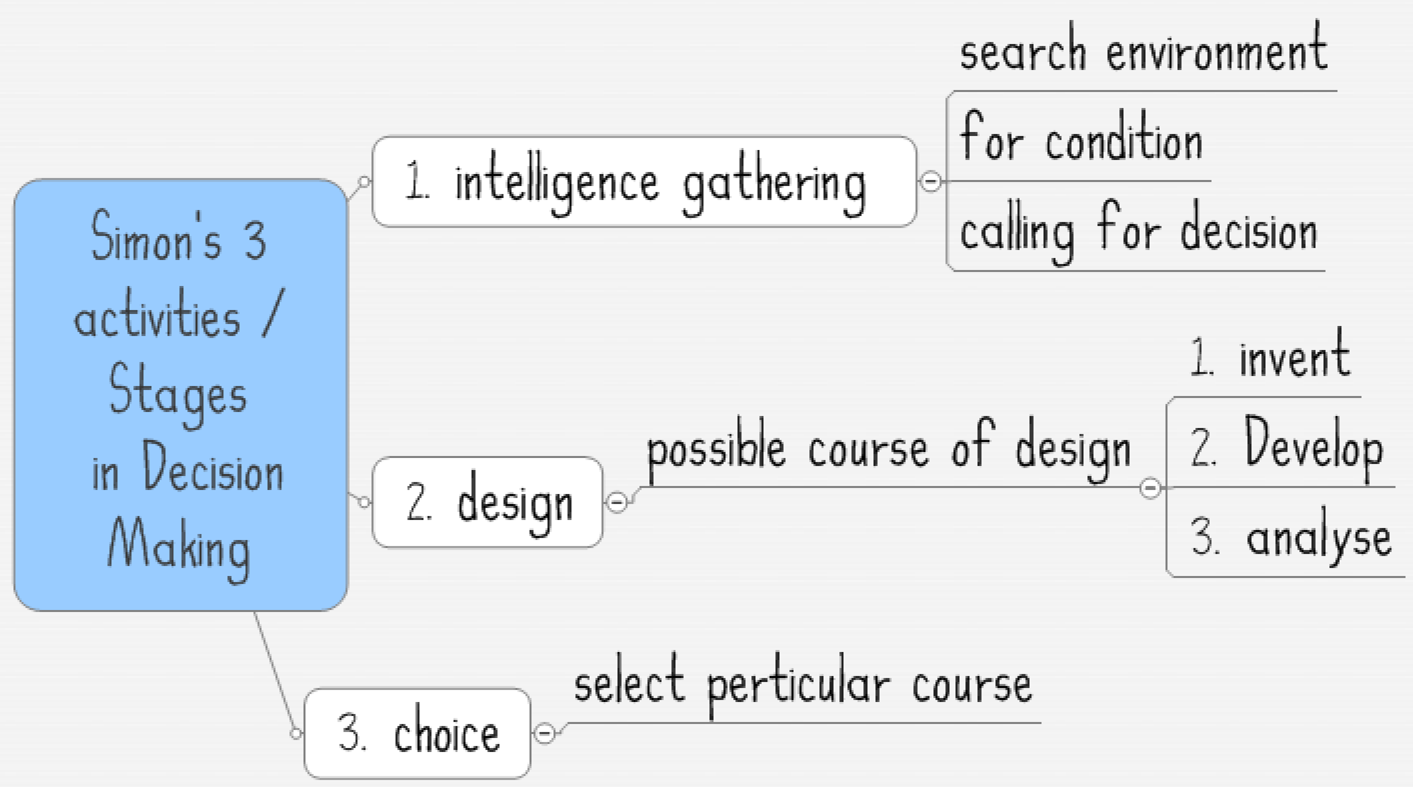


PATFrame Objective

To provide a decision support tool encompassing a prescriptive and adaptive framework for UAS SoS Testing

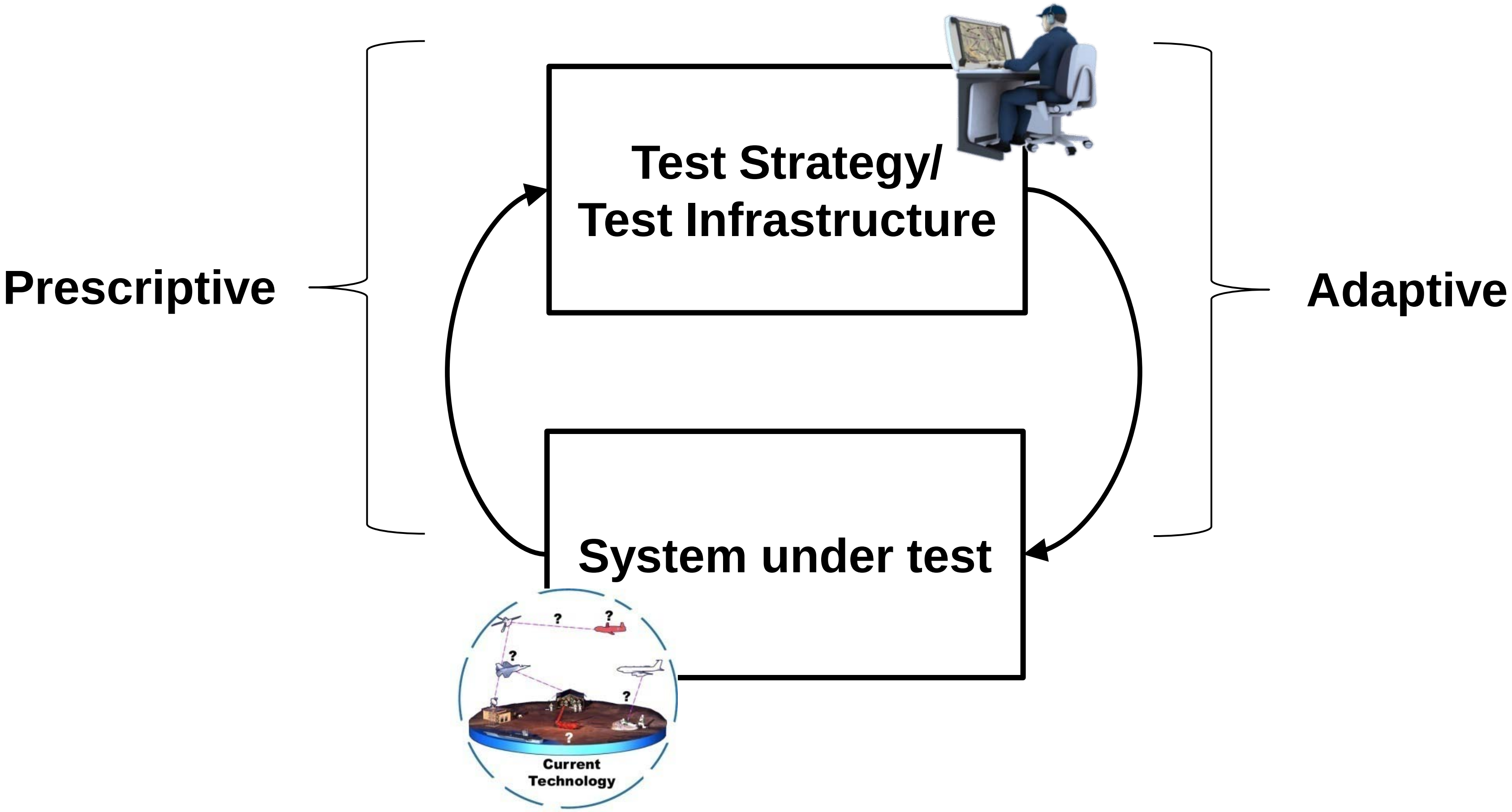
- PATFrame will be implemented using a software dashboard that will enable improved decision making for the UAS T&E community
- Focused on addressing BAA topics ***TTE-6 Prescribed System of Systems Environments*** and ***MA-6 Adaptive Architectural Frameworks***
- Three University team (MIT-USC-UTA) draws from experts in test & evaluation, decision theory, systems engineering, software architectures, robotics and modeling
- Based on Valerdi, R., Ross, A. and Rhodes, D., "A Framework for Evolving System of Systems Engineering," *CrossTalk - The Journal of Defense Software Engineering, 20(10),* 28-30, 2007.

Three Stages in Decision Making



Simon, H. (1976), *Administrative Behavior* (3rd ed.), New York: The Free Press

Prescriptive Adaptive Test Framework



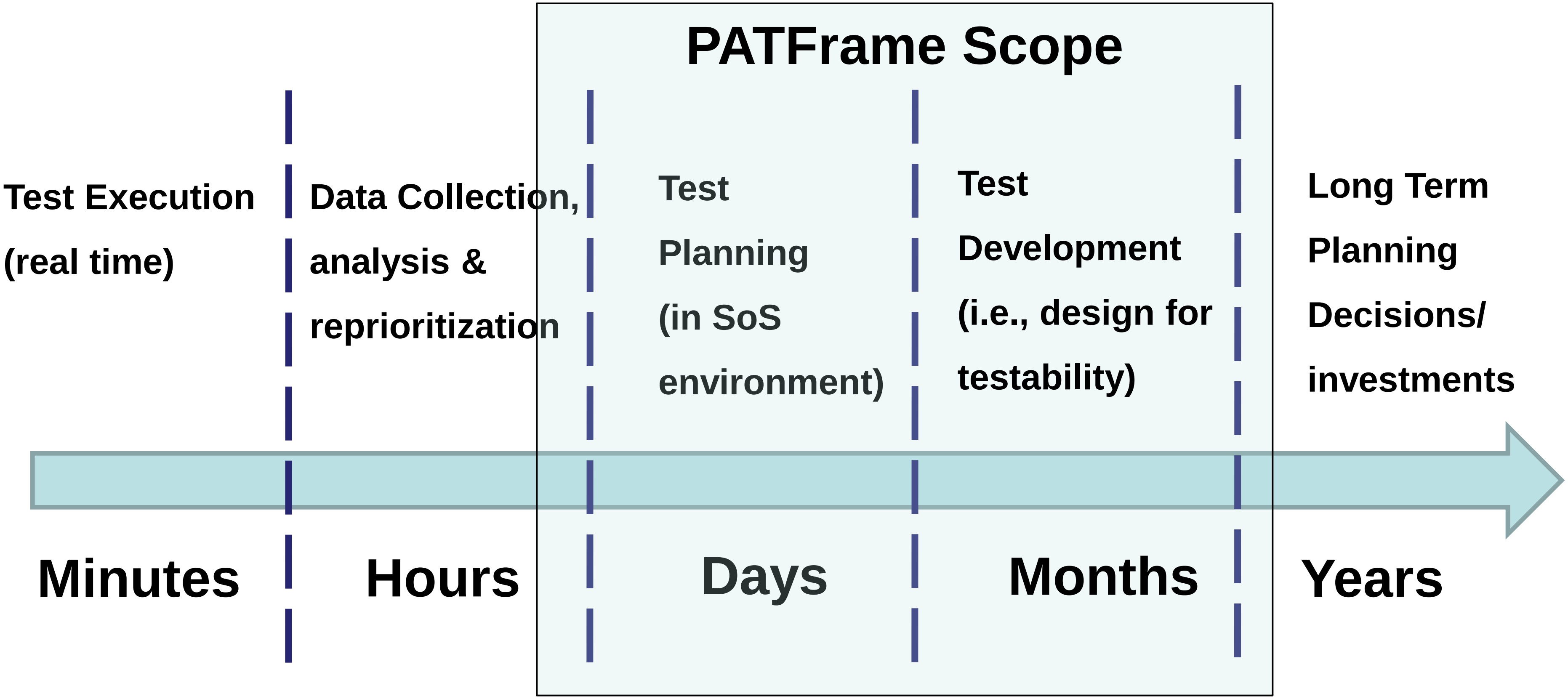
Prescriptive

Test Strategy/
Test Infrastructure

System under test

Adaptive

Time Scale for Testing Decisions



PATFrame Scope

Test Execution (real time)

Data Collection, analysis & reprioritization

Test Planning (in SoS environment)

Test Development (i.e., design for testability)

Long Term Planning Decisions/ investments

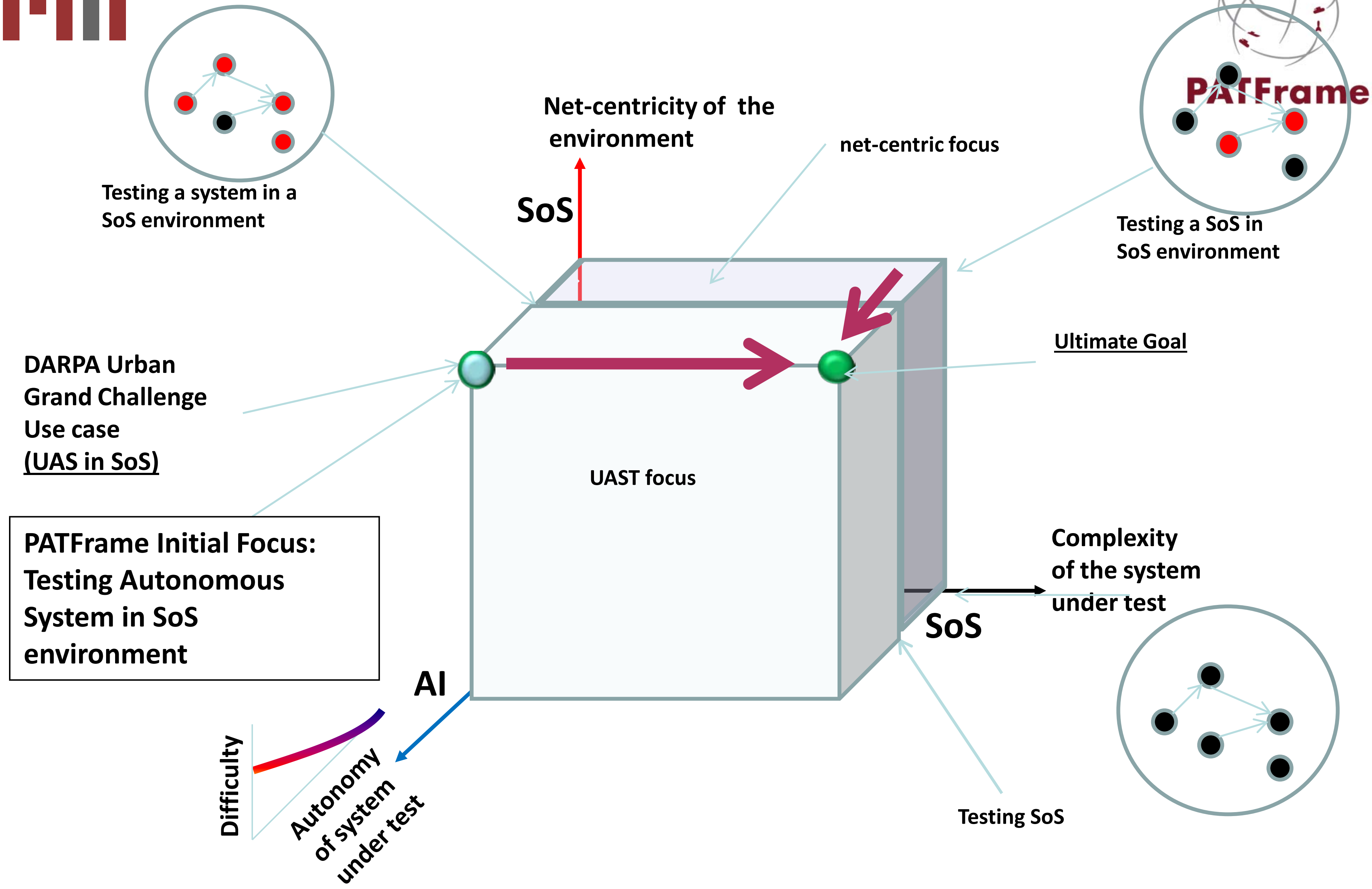
Minutes

Hours

Days

Months

Years



Testing a system in a SoS environment

Net-centricity of the environment

SoS

net-centric focus

Testing a SoS in SoS environment

DARPA Urban Grand Challenge Use case <u>(UAS in SoS)</u>

<u>Ultimate Goal</u>

UAST focus

PATFrame Initial Focus: Testing Autonomous System in SoS environment

Complexity of the system under test

SoS

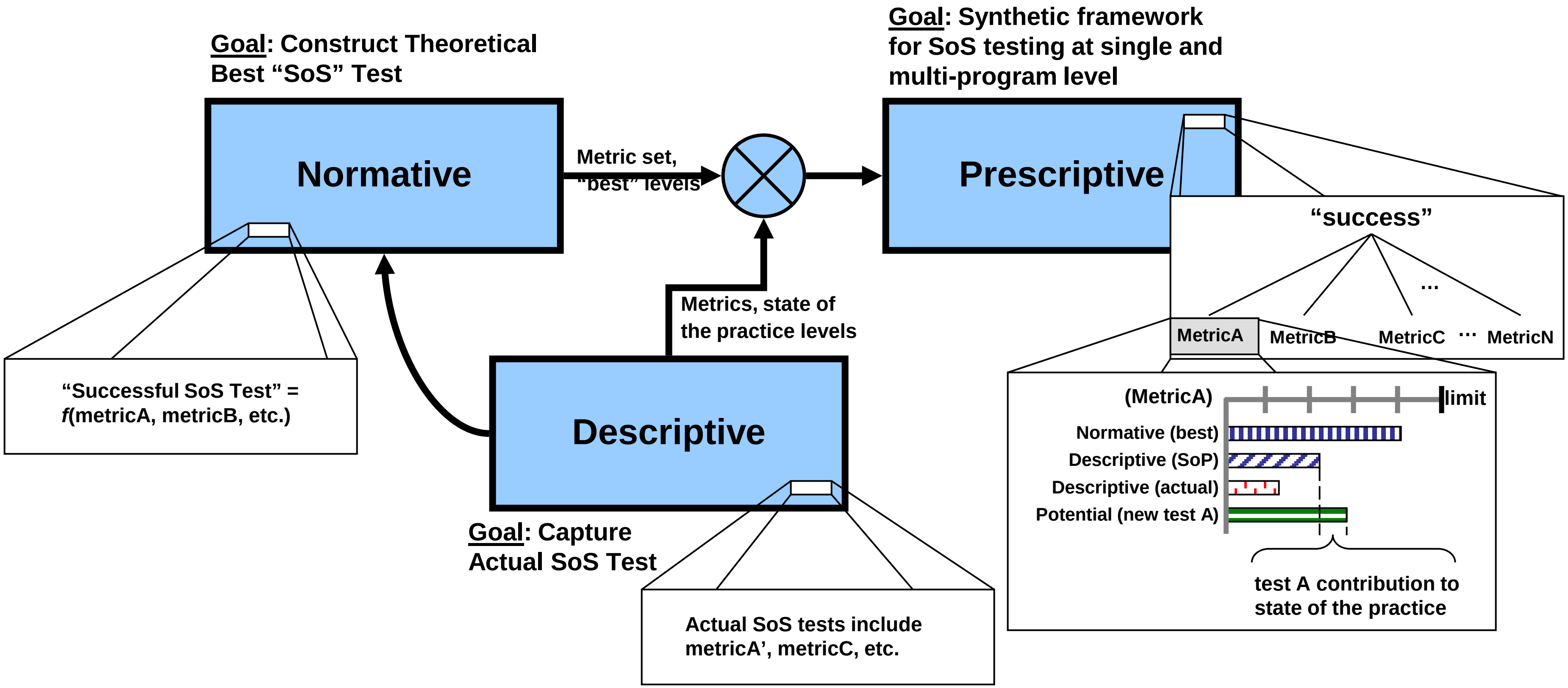
AI

Difficulty

Autonomy of system under test

Testing SoS

Prescribed System of Systems Environment


Goal: Construct Theoretical Best "SoS" Test
Normative
"Successful SoS Test" = f(metricA, metricB, etc.)
Metric set, "best" levels
Goal: Synthetic framework for SoS testing at single and multi-program level
Prescriptive
"success"
MetricA
MetricB
MetricC
…
MetricN
(MetricA)
limit
Normative (best)
Descriptive (SoP)
Descriptive (actual)
Potential (new test A)
test A contribution to state of the practice
Metrics, state of the practice levels
Descriptive
Goal: Capture Actual SoS Test
Actual SoS tests include metricA', metricC, etc.

Integrated Test Management

Primary Inputs to PATFrame

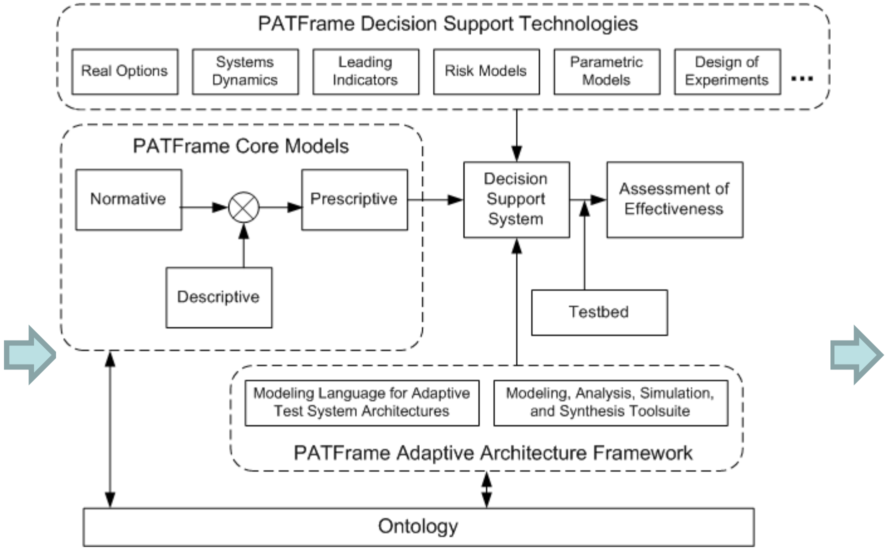
SUT requirements, capabilities, & architecture

Mission needs and goals

Test resources & associated architecture (infrastructure)

Test requirements

Schedule, resources and cost constraints


PATFrame Decision Support Technologies
Real Options
Systems Dynamics
Leading Indicators
Risk Models
Parametric Models
Design of Experiments
...
PATFrame Core Models
Normative
Prescriptive
Descriptive
Decision Support System
Assessment of Effectiveness
Testbed
Modeling Language for Adaptive Test System Architectures
Modeling, Analysis, Simulation, and Synthesis Toolsuite
PATFrame Adaptive Architecture Framework
Ontology

Primary Outputs for UASoS T&E Planners

UASoS test strategy, recommended tests & their sequencing

Feasible T&E test planning options

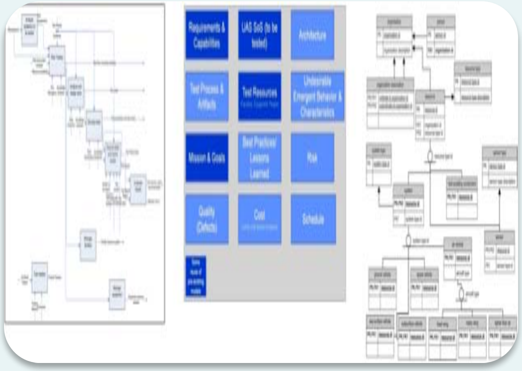
UASoS test strategy estimated cost

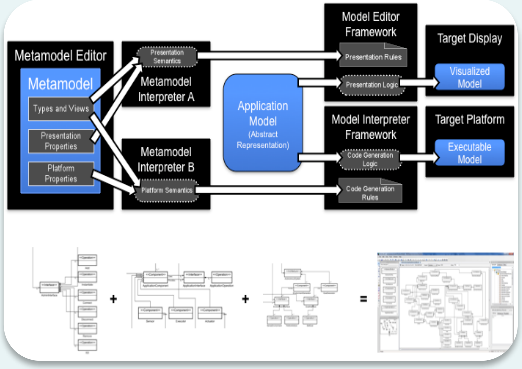
UASoS test strategy risks

Undesirable emergent behavior

Recommended and automatic adaptations to UASoS T&E issues

Current Efforts: Ontology and Architecture Frameworks





Metamodel Editor

Metamodel

Types and Views

Presentation Properties

Platform Properties

Metamodel Interpreter A

Presentation Semantics

Metamodel Interpreter B

Platform Semantics

Application Model (Abstract Representation)

Model Editor Framework

Presentation Rules

Presentation Logic

Model Interpreter Framework

Code Generation Logic

Code Generation Rules

Target Display

Visualized Model

Target Platform

Executable Model

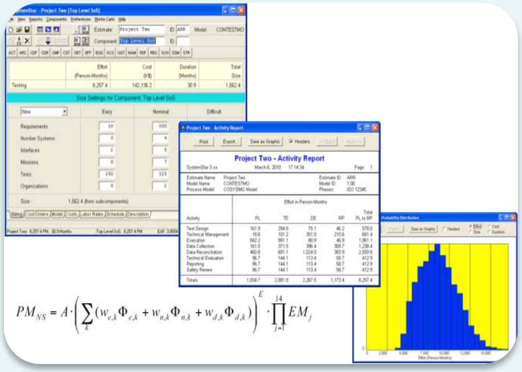
Ontology

Meta Modeling

Discrete Event Simulation

Current Efforts: Normative Models

PATFrame



stemStar - Project Two (Top Level SoS)

Estimate: Project Two ID: ARR Model: CONTESTMO

Component: Top Level SoS

	Effort (Person-Months)	Cost (K$)	Duration (Months)	Total Size
Testing	8,257.4	142,136.2	30.9	1,562.4

Size Settings for Component: Top Level SoS

New	Easy	Nominal	Difficult
Requirements	10	300	
Number Systems	0	4	
Interfaces	2	5	
Missions	0	7	
Tests	150	325	
Organizations	0	2	

Size: 1,562.4 (from subcomponents)

Project Two - Activity Report

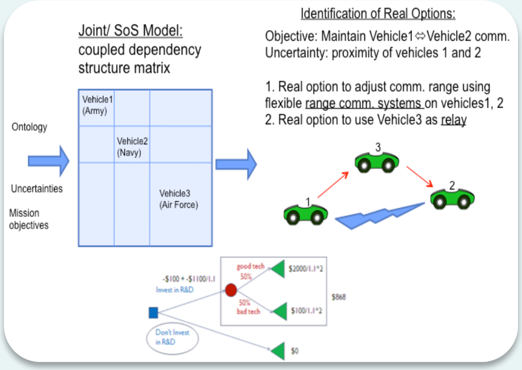
SystemStar 3.xx March 6, 2010 17:14:34 Page 1

Estimate Name: Project Two Estimate ID: ARR
Model Name: CONTESTMO Model ID: 1.00
Process Model: COSYSMO Model Phases: ISO 12345

Effort in Person-Months

Activity	PL	TE	DE	RP	Total PL to RP
Test Design	161.9	294.8	75.1	46.2	578.0
Technical Management	18.6	101.2	351.0	210.6	681.4
Execution	842.2	991.1	80.9	46.9	1,961.1
Data Collection	161.0	371.5	396.4	309.7	1,238.4
Data Reconciliation	460.8	691.1	1,024.0	383.9	2,559.8
Technical Evaluation	96.7	144.1	113.4	58.7	412.9
Reporting	96.7	144.1	113.4	58.7	412.9
Safety Review	96.7	144.1	113.4	58.7	412.9
Totals	1,934.7	2,881.8	2,267.5	1,173.4	8,257.4

$$PM_{NS} = A \cdot \left(\sum_k \left(w_{e,k}\Phi_{e,k} + w_{n,k}\Phi_{n,k} + w_{d,k}\Phi_{d,k} \right) \right)^E \cdot \prod_{j=1}^{14} EM_j$$

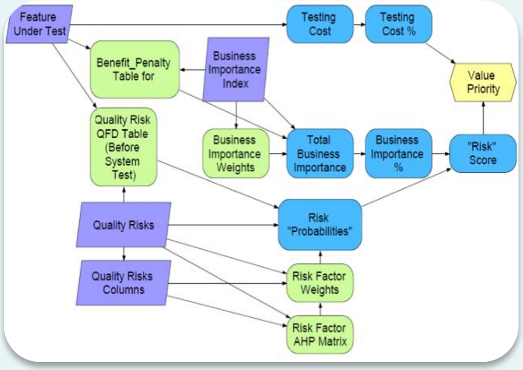


Joint/ SoS Model: coupled dependency structure matrix

Identification of Real Options:

Objective: Maintain Vehicle1⇔Vehicle2 comm.
Uncertainty: proximity of vehicles 1 and 2

1. Real option to adjust comm. range using flexible range comm. systems on vehicles1, 2
2. Real option to use Vehicle3 as relay

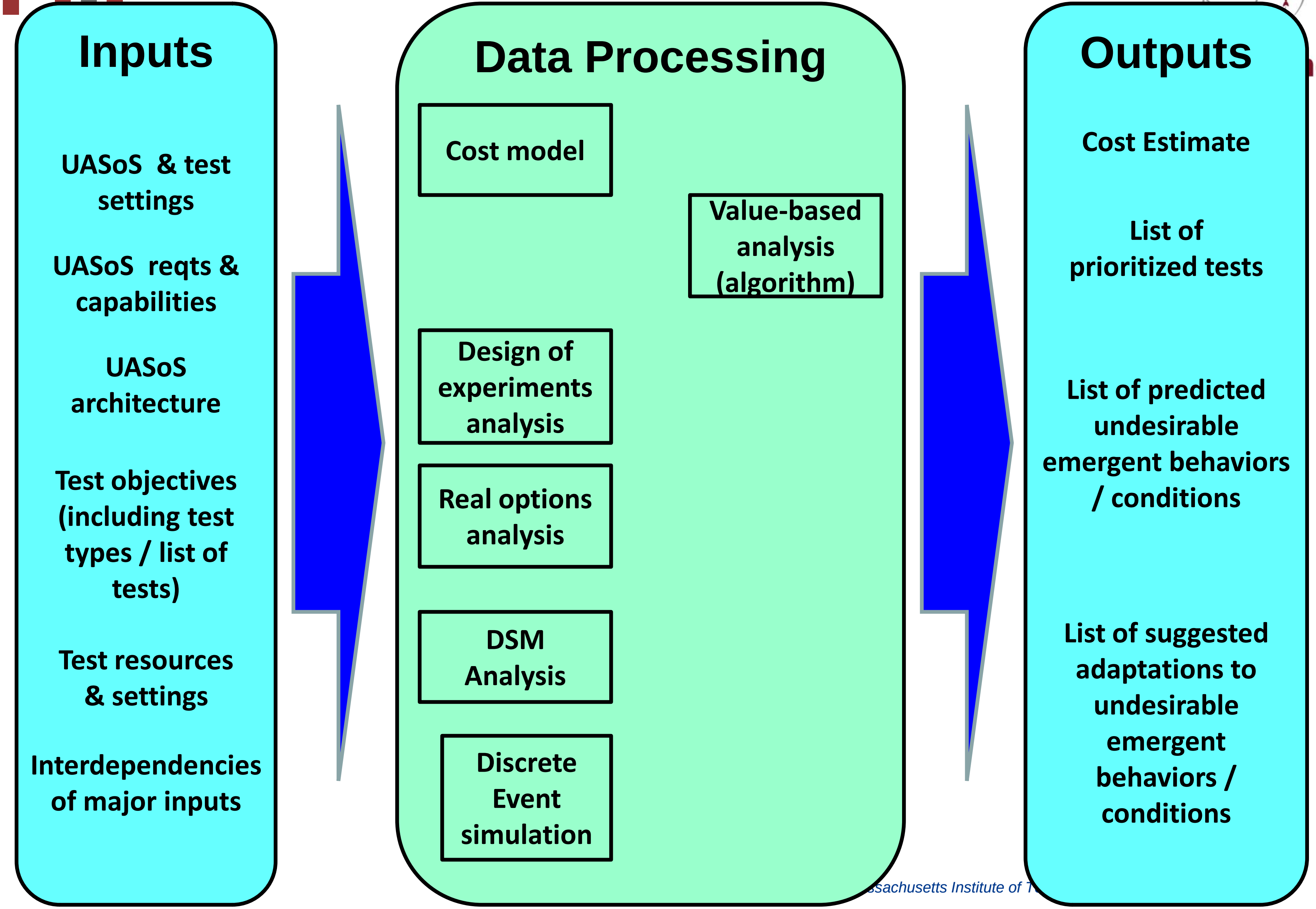


Cost Modeling

Real Options

Value-based Testing

PATFrame Workflow



Inputs

UASoS & test settings

UASoS reqts & capabilities

UASoS architecture

Test objectives (including test types / list of tests)

Test resources & settings

Interdependencies of major inputs

Data Processing

Cost model

Value-based analysis (algorithm)

Design of experiments analysis

Real options analysis

DSM Analysis

Discrete Event simulation

Outputs

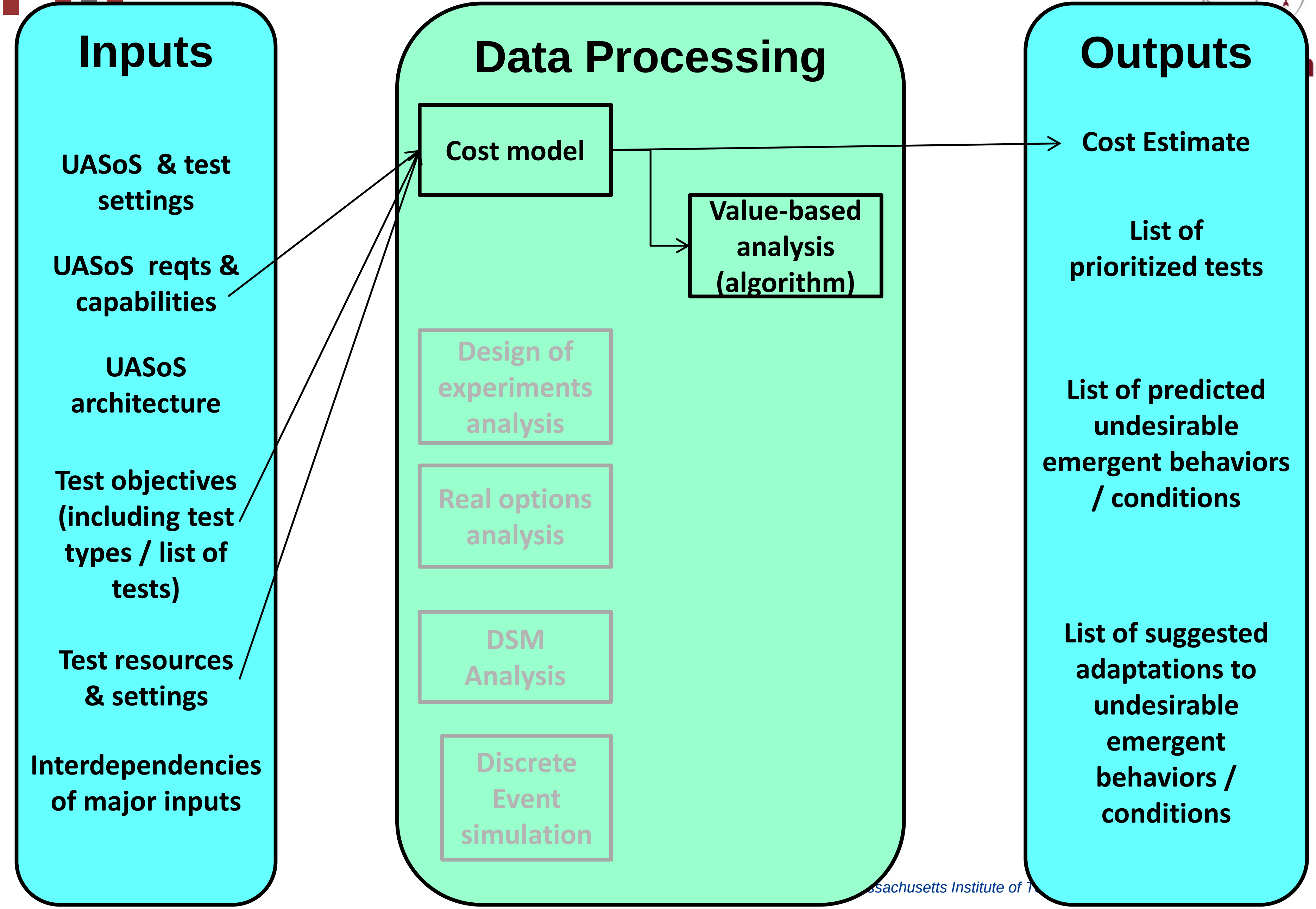
Cost Estimate

List of prioritized tests

List of predicted undesirable emergent behaviors / conditions

List of suggested adaptations to undesirable emergent behaviors / conditions

Cost Model Workflow



Inputs

UASoS & test settings

UASoS reqts & capabilities

UASoS architecture

Test objectives (including test types / list of tests)

Test resources & settings

Interdependencies of major inputs

Data Processing

Cost model

Value-based analysis (algorithm)

Design of experiments analysis

Real options analysis

DSM Analysis

Discrete Event simulation

Outputs

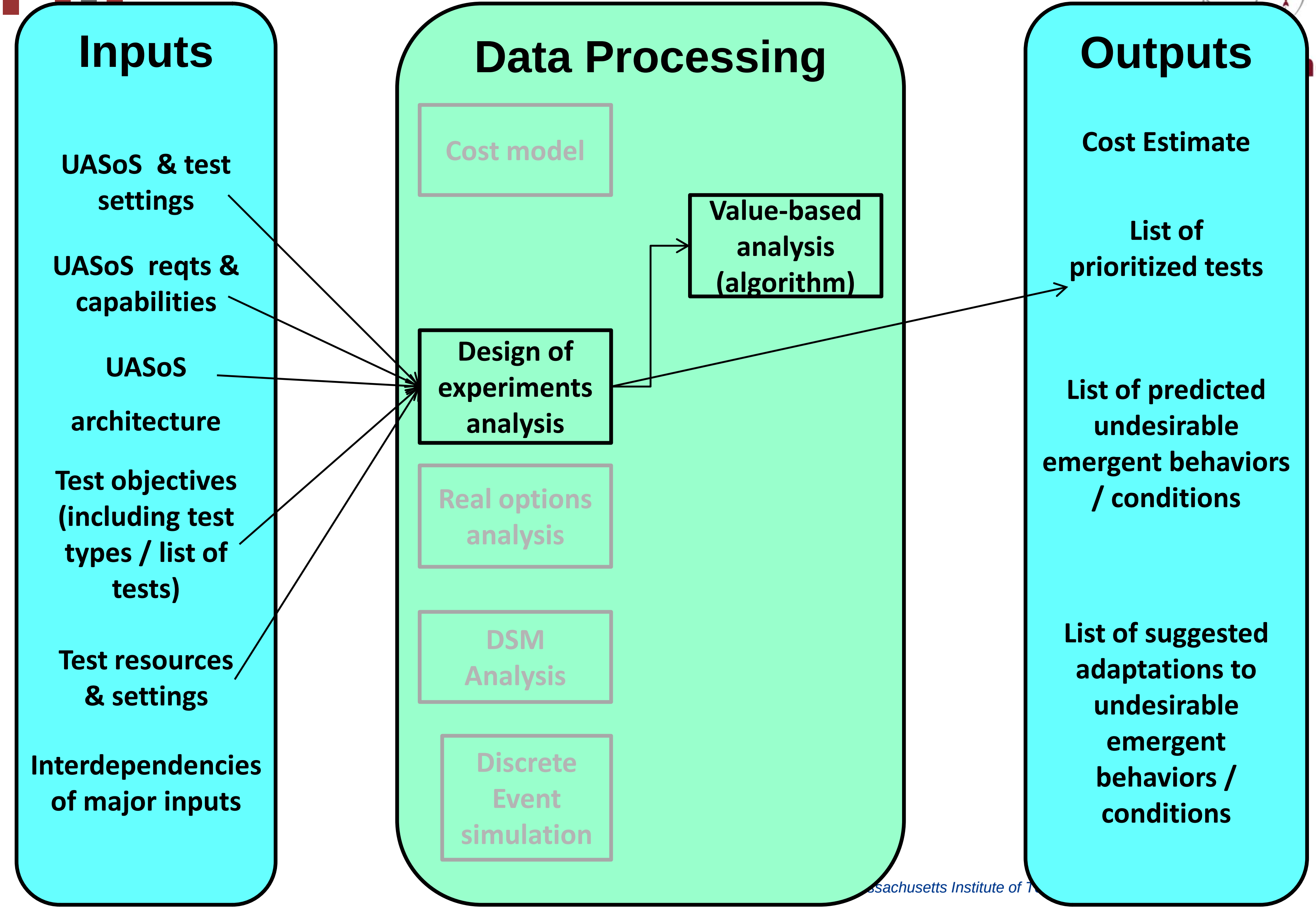
Cost Estimate

List of prioritized tests

List of predicted undesirable emergent behaviors / conditions

List of suggested adaptations to undesirable emergent behaviors / conditions

Design of Experiments Analysis Workflow



Inputs

UASoS & test settings

UASoS reqts & capabilities

UASoS architecture

Test objectives (including test types / list of tests)

Test resources & settings

Interdependencies of major inputs

Data Processing

Cost model

Value-based analysis (algorithm)

Design of experiments analysis

Real options analysis

DSM Analysis

Discrete Event simulation

Outputs

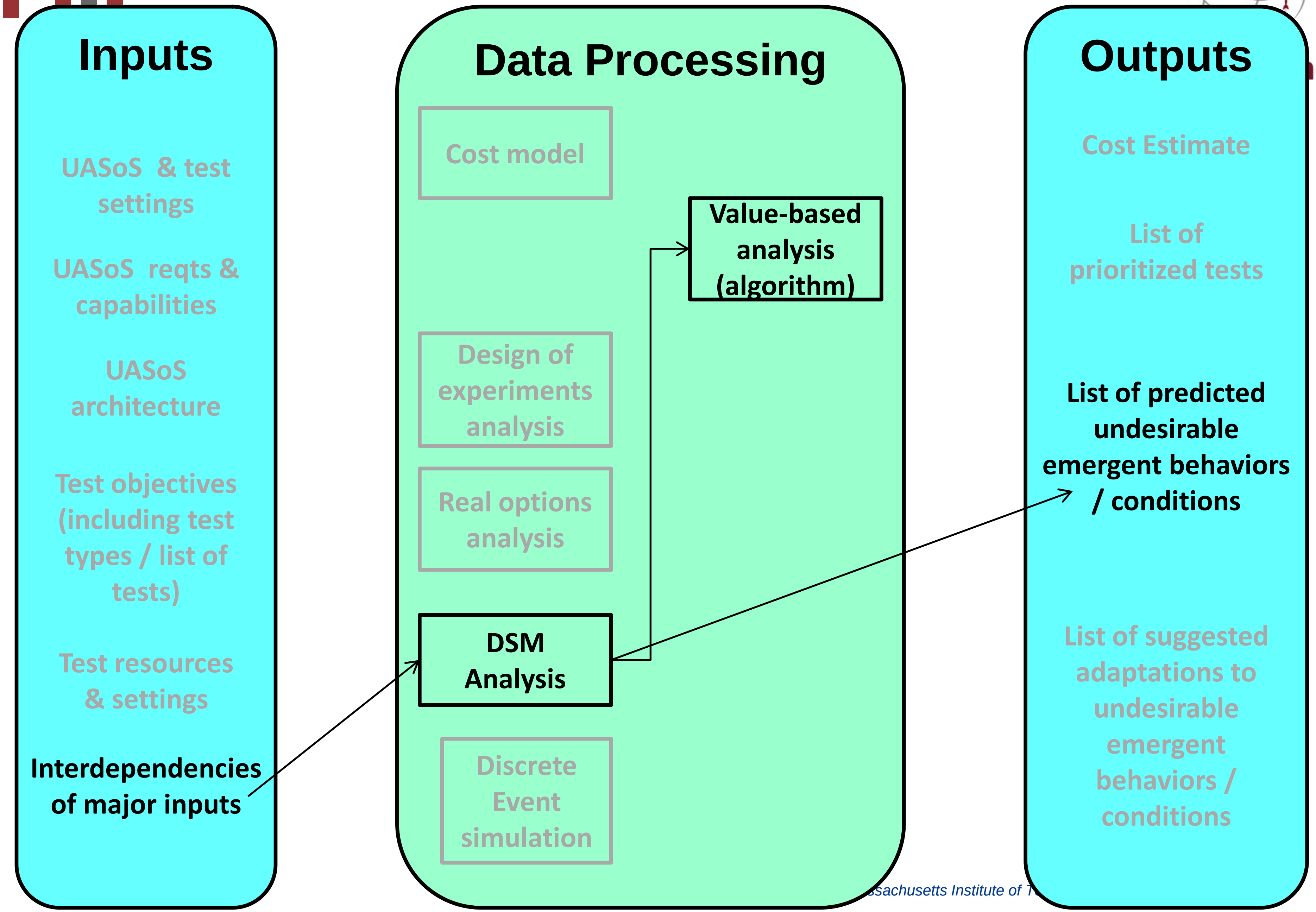
Cost Estimate

List of prioritized tests

List of predicted undesirable emergent behaviors / conditions

List of suggested adaptations to undesirable emergent behaviors / conditions

Dependency Structure Matrix (DSM) Analysis Workflow



Inputs

UASoS & test settings

UASoS reqts & capabilities

UASoS architecture

Test objectives (including test types / list of tests)

Test resources & settings

Interdependencies of major inputs

Data Processing

Cost model

Value-based analysis (algorithm)

Design of experiments analysis

Real options analysis

DSM Analysis

Discrete Event simulation

Outputs

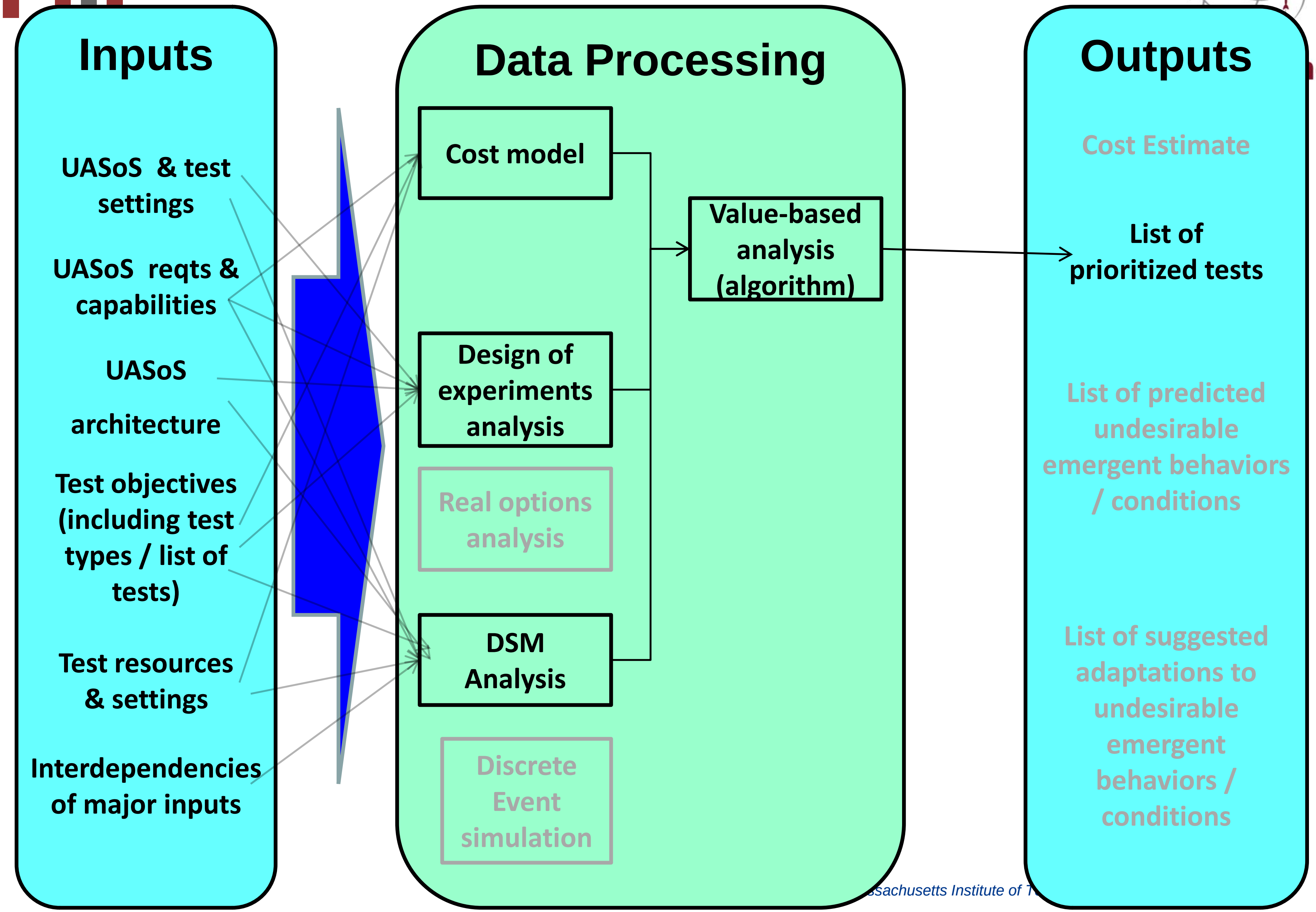
Cost Estimate

List of prioritized tests

List of predicted undesirable emergent behaviors / conditions

List of suggested adaptations to undesirable emergent behaviors / conditions

Value-based Analysis Workflow



Inputs

UASoS & test settings

UASoS reqts & capabilities

UASoS

architecture

Test objectives (including test types / list of tests)

Test resources & settings

Interdependencies of major inputs

Data Processing

Cost model

Design of experiments analysis

Real options analysis

DSM Analysis

Discrete Event simulation

Value-based analysis (algorithm)

Outputs

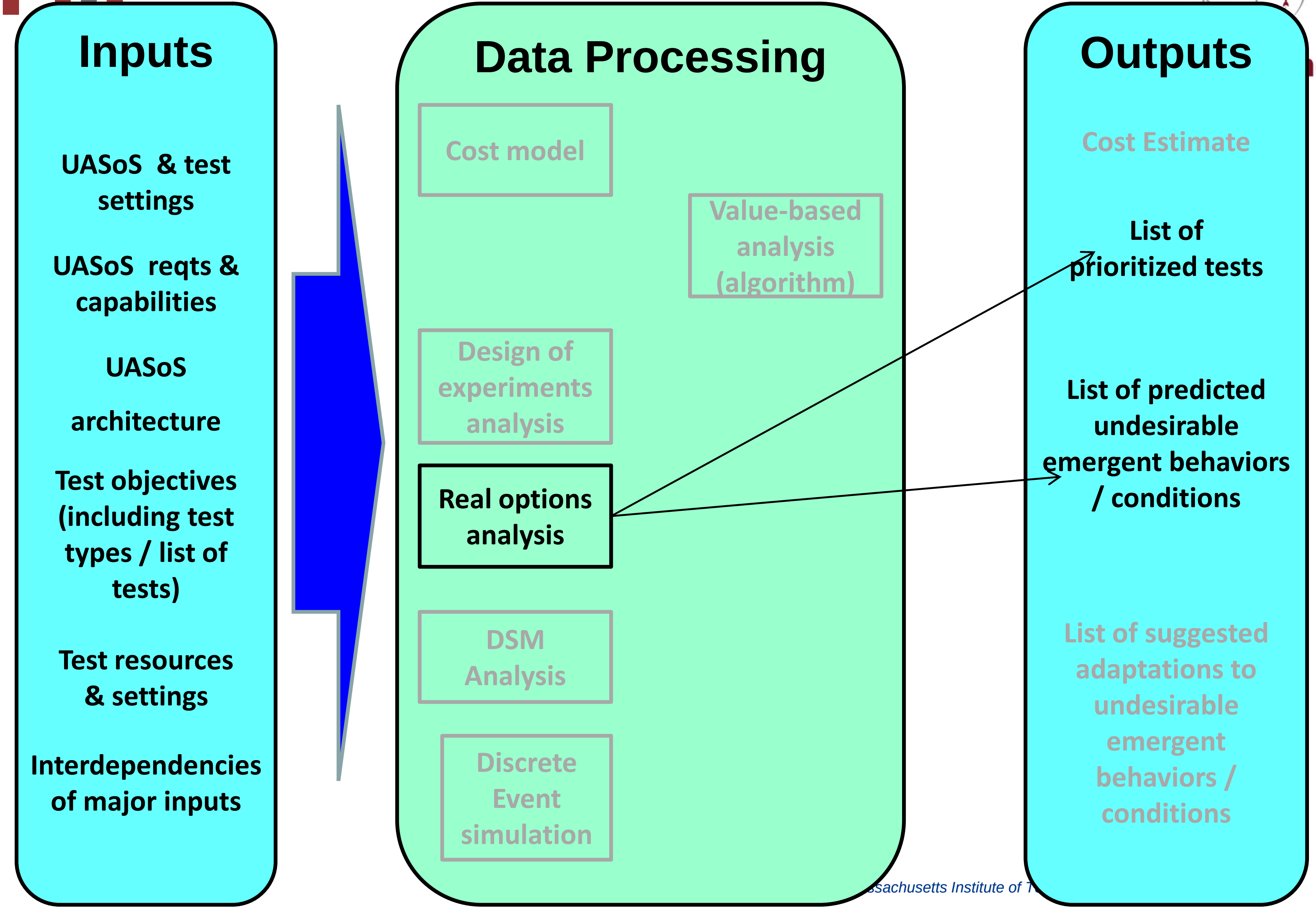
Cost Estimate

List of prioritized tests

List of predicted undesirable emergent behaviors / conditions

List of suggested adaptations to undesirable emergent behaviors / conditions

Real Options Analysis Workflow



Inputs

UASoS & test settings

UASoS reqts & capabilities

UASoS

architecture

Test objectives (including test types / list of tests)

Test resources & settings

Interdependencies of major inputs

Data Processing

Cost model

Value-based analysis (algorithm)

Design of experiments analysis

Real options analysis

DSM Analysis

Discrete Event simulation

Outputs

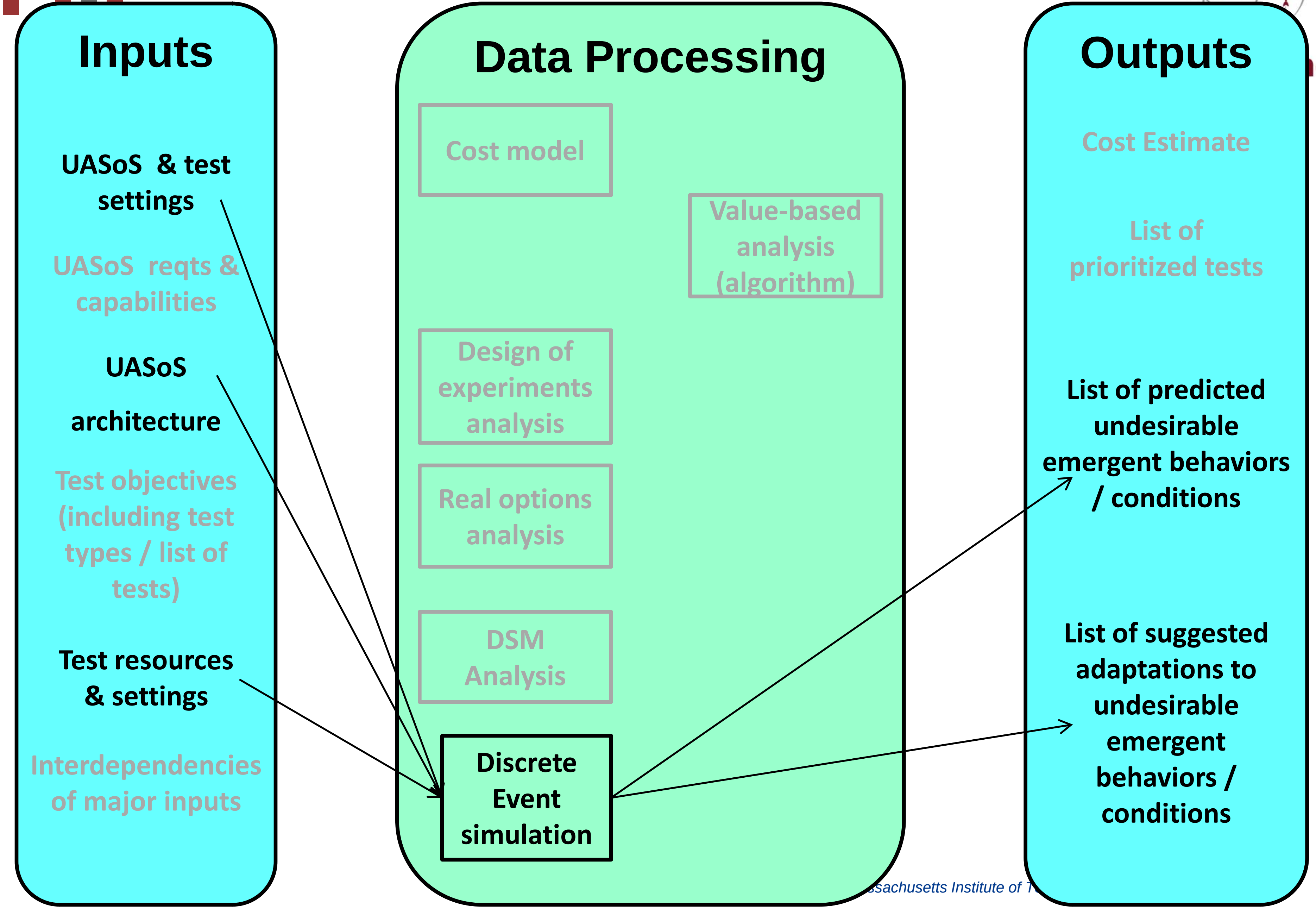
Cost Estimate

List of prioritized tests

List of predicted undesirable emergent behaviors / conditions

List of suggested adaptations to undesirable emergent behaviors / conditions

Discrete Event Simulation Workflow



Inputs

UASoS & test settings

UASoS reqts & capabilities

UASoS architecture

Test objectives (including test types / list of tests)

Test resources & settings

Interdependencies of major inputs

Data Processing

Cost model

Value-based analysis (algorithm)

Design of experiments analysis

Real options analysis

DSM Analysis

Discrete Event simulation

Outputs

Cost Estimate

List of prioritized tests

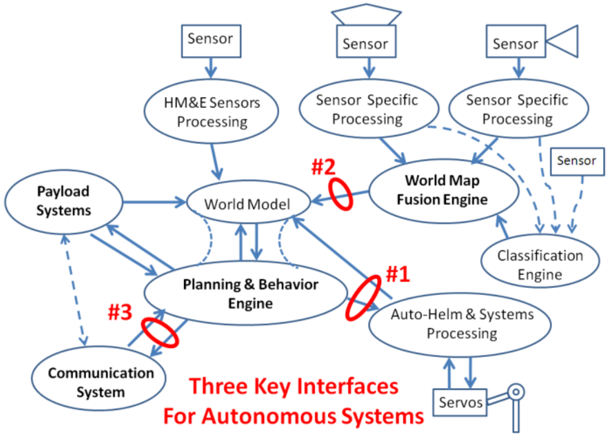
List of predicted undesirable emergent behaviors / conditions

List of suggested adaptations to undesirable emergent behaviors / conditions

Use Case 1: Define and Prioritize Tests

- Operational thread
  - NAVSEA unmanned surface vehicles (USVs) must comply with International Regulations for Preventing Collisions at Sea (COLREGS)*
- T&E thread
  - Action to avoid collision shall be: positive, obvious, made in good time**
  - Validate USVs ability to self-adapt to: learn, sense & avoid, perform automated coupling, optimize adaptive software***

Three Yellow Buoys
STOP
START



Three Key Interfaces For Autonomous Systems

*Hansen, E. C., "USV Performance Testing," April 14, 2010.

***Part B, Sect. I, Sec. 8, Convention on the International Regulations for Preventing Collisions at Sea,* IMO (The International Maritime Organisation), *1972.*

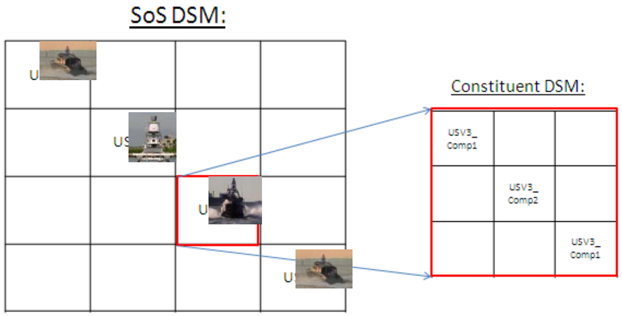
***Engineering Autonomous System Architectures.

Use Case 1: Define and Prioritize Tests

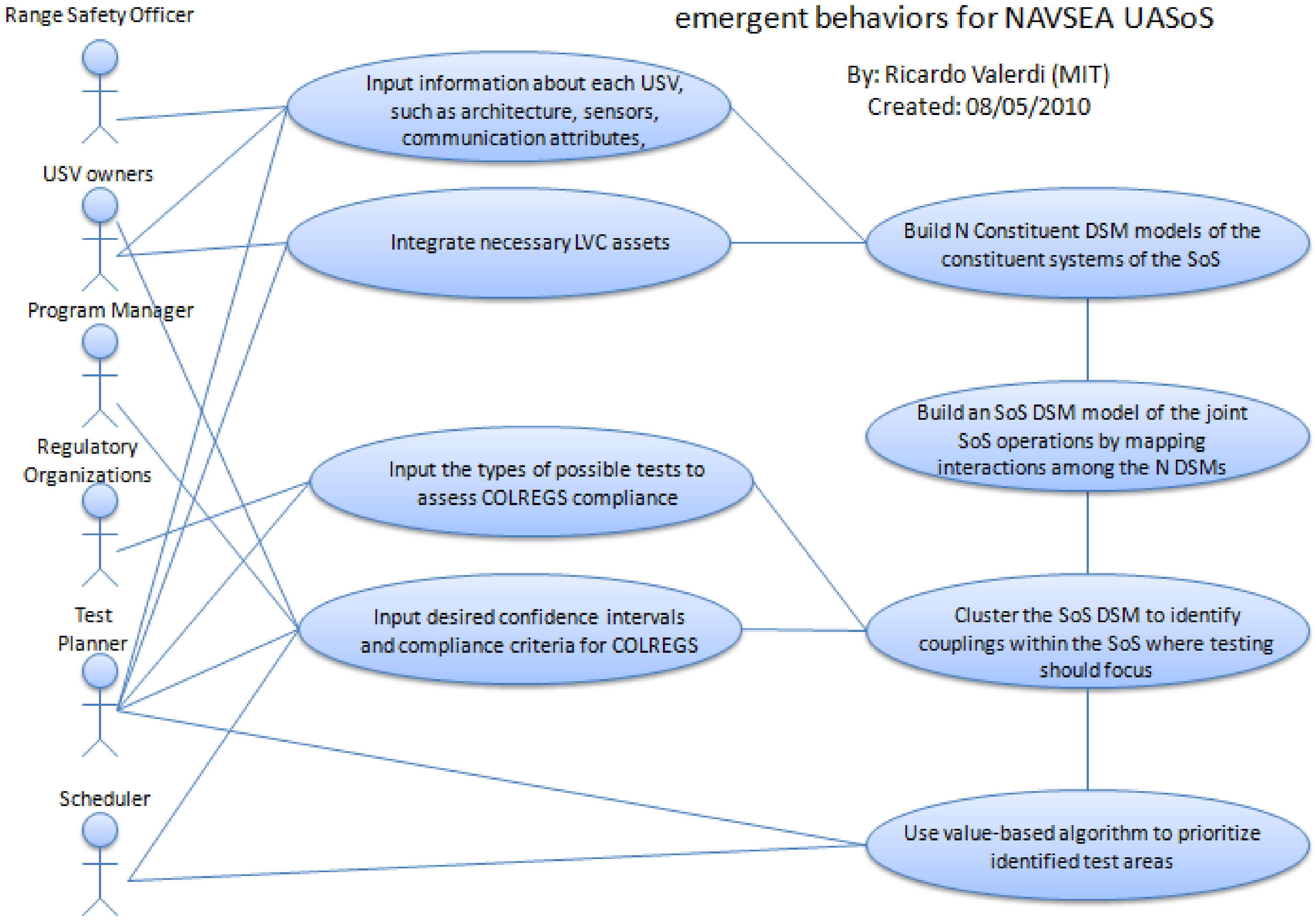
Use Case Name: Test selection and prioritization for NAVSEA UASoS
Goal: Define and prioritize a set of tests for an unmanned & autonomous SoS comprised of NAVSEA's fleet of unmanned surface vehicles (USVs)
Summary: SoS cannot be exhaustively tested, therefore tests must be chosen that provide the most value within the allocated time and budget
Actors: Test planner personnel, program manager, scheduler, range safety officer, owners of USVs, regulatory organization(s)
Components: Dependency Structure Matrix (DSM) modeling interface, DSM clustering algorithm, XTEAM meta-modeling environment, value-based testing algorithm, LVC environment
Normal Flow:

1. Input information about each USV, such as architecture, sensors, communication attributes, etc.
2. Integrate necessary LVC assets
3. Input the types of possible tests to assess COLREGS compliance
4. Input desired confidence intervals and compliance criteria for COLREGS
5. PATFrame outputs

    A prioritized set of tests to perform
    Expected level of confidence of COLREGS compliance after each test



Use Case 1: Define and Prioritize Tests



Test selection and prioritization based on emergent behaviors for NAVSEA UASoS

By: Ricardo Valerdi (MIT)
Created: 08/05/2010

Range Safety Officer
USV owners
Program Manager
Regulatory Organizations
Test Planner
Scheduler
Input information about each USV, such as architecture, sensors, communication attributes,
Integrate necessary LVC assets
Input the types of possible tests to assess COLREGS compliance
Input desired confidence intervals and compliance criteria for COLREGS
Build N Constituent DSM models of the constituent systems of the SoS
Build an SoS DSM model of the joint SoS operations by mapping interactions among the N DSMs
Cluster the SoS DSM to identify couplings within the SoS where testing should focus
Use value-based algorithm to prioritize identified test areas

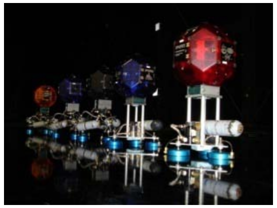
Testing to Reduce SoS Risks vs. the Risks of Performing Tests on an SoS

SoS Risks

- What are unique risks for UAS's? For UAS's operating in an SoS environment?
- How do you use testing to mitigate these risks?
- What are metrics that you are using to measure the level of risk?

Risks of Testing an SoS

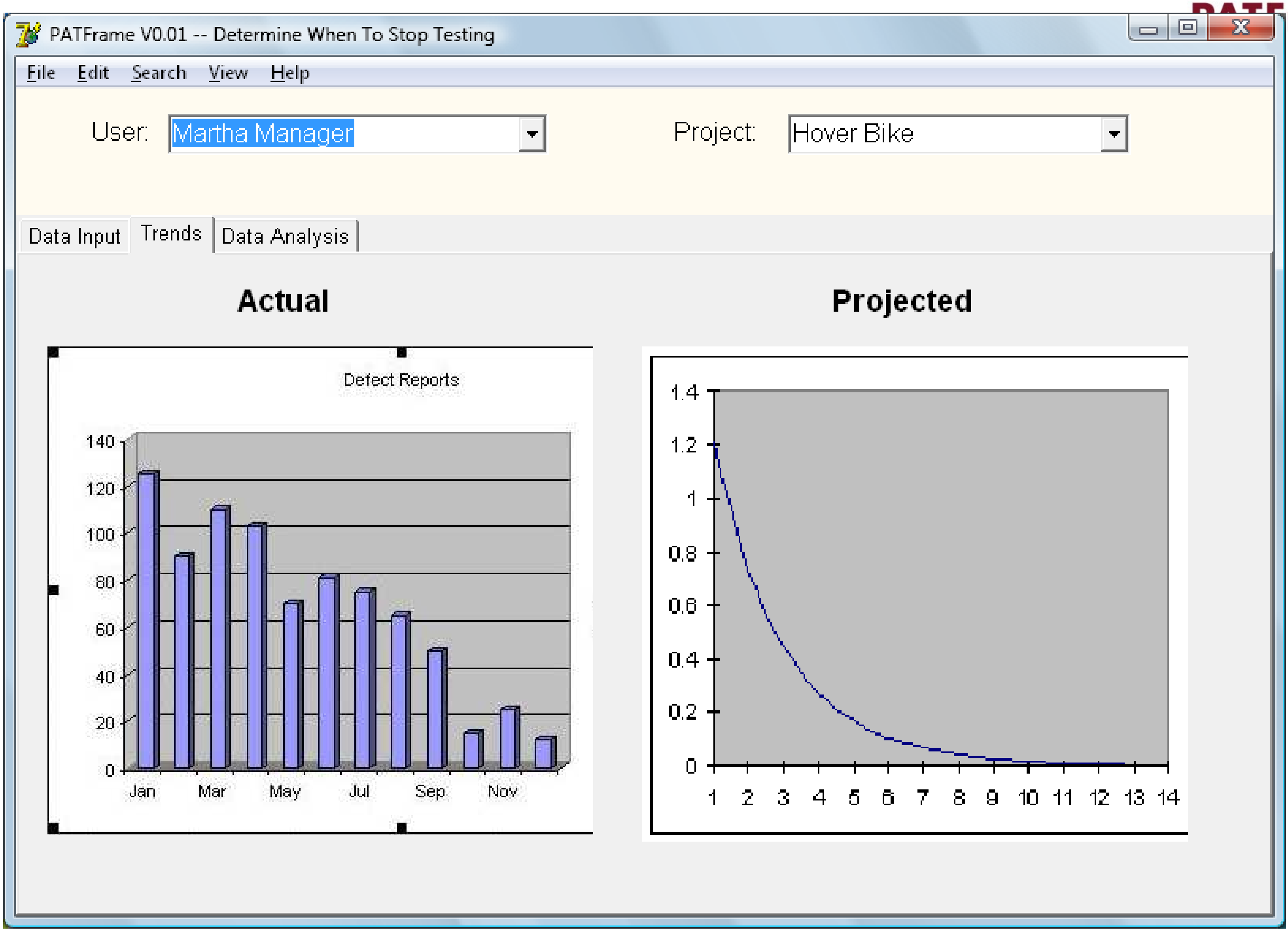
- What are unique programmatic risks that impact your ability to do testing on UAS's? To do testing on UAS's operating in an SoS environment?
- What methods do you use to mitigate these risks?
- What are the metrics that you are using to measure the level of programmatic risk in testing?



Example PATFrame Tool Concept

- Question:
  - When am I Done testing?
- Technology:
  - Defect estimation model
  - Trade Quality for Delivery Schedule
- Inputs:
  - Defects discovered
- Outputs:
  - Defects remaining, cost to quit, cost to continue

When am I done testing?



PATFrame V0.01 -- Determine When To Stop Testing

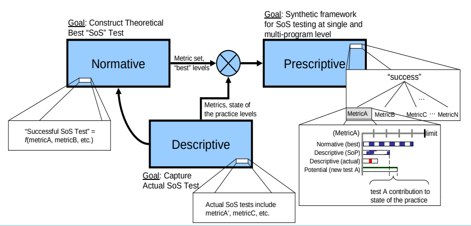
User: Martha Manager

Project: Hover Bike

Technical Specifications

	Specifications			
Parameter	**Current Performance Level**	**Current Target***	**Ultimate Goal***	**Achieved**
Automation level in predicting undesirable emergent behavior	Manual	Semi-automated (moderate – based on identified best practices and rules from SMEs)	Semi–automated (extensive – current target plus collected patterns in SUTs, test infrastructure, associated test settings)	
Probability of automatically predicting actual undesirable emergent behavior before testing (emergent behavior will occur)	No capability	0.90	0.995	
Rate of falsely predicting emergent behavior automatically and before testing (emergent behavior won't occur – false pos.)	No capability	1 per 100 scenarios	1 per 1000 scenarios	

PATFrame Technology Maturation Plan

1. Applied Research Phase (TRL 3-4)  Goal: Construct Theoretical Best "SoS" Test Normative Metric set, "best" levels Goal: Synthetic framework for SoS testing at single and multi-program level Prescriptive "success" … MetricA MetricB MetricC … MetricN "Successful SoS Test" = f(metricA, metricB, etc.) Metrics, state of the practice levels Descriptive Goal: Capture Actual SoS Test Actual SoS tests include metricA', metricC, etc. (MetricA) limit Normative (best) Descriptive (SoP) Descriptive (actual) Potential (new test A) test A contribution to state of the practice	**1A.** Define Prescribed SoS Environments	Critical UAS SoS T&E concepts are validated through a normative, descriptive and prescriptive framework.
	1B. Implement Adaptive Architectural Frameworks	Basic ideas are implemented in a prototype and integrated through a UAS SoS T&E ontology model, system dynamics model and SoS architecture notions for dynamic adaptation of SoS T&E.
2. Development Phase (TRL 4-5) 	**2A.** Develop Decision Support System	Prescriptive framework (PATFrame) is developed as a decision support system to enable better outcomes of UAS T&E.
	2B. Refine Use Cases	Use cases identified by transition partners are used to validate PATFrame in relevant environments. Simulations are performed in specific operational domains (ground, air, space, water, underwater).
3. Deployment Phase (TRL 5-6) Legend Army Air Force Navy DISA Cold Regions Test Center Tropic Regions Test Center Dugway Proving Ground Nevada Test & Training Range Utah Test & Training Range 46th Test Group Aberdeen Test Center Naval Air Warfare Center Weapons Division Naval Air Warfare Center Aircraft Division 30th Space Wing Arnold Engineering Development Center Air Force Flight Test Center Yuma Proving Ground 46th Test Wing 45th Space Wing Electronic Proving Ground High Energy Laser Systems Test Facility Pacific Missile Range Facility White Sands Missile Range Joint Interoperability Test Command Atlantic Undersea Test & Evaluation Center Reagan Test Site	**3A.** Conduct Focused Experiments	Readiness is demonstrated through focused experiments in simulated operational environments (i.e., Future Combat Systems).
	3B. Deploy Decision Support System	PATFrame documented test performance demonstrating agreement with analytical predictions deployed across facilities in the Army, Navy and Air Force engaged in UAS T&E.