



NASA's Complex Autonomous Systems Assurance Technical Challenge

4 March 2021

Presented by
Dr. Misty Davies (DPM)

...with input and help from many others



System-Wide Safety

Machine-Learning Enabled Components

The drive towards ‘increasingly autonomous’ systems in Aviation is being driven by:

- The growing complexity of the vehicles to address growing number of decisions to be made by operators (Boeing 737 MCAS)
- An explosion of unmanned missions (small drone operations, cargo delivery)
- A shortage of traditionally-trained pilots (simplified vehicle operations)

MLEC

Introducing the machine-learning enabled component (MLEC) to aviation

And this trend is being fueled by:

- Increased processor capability (Moore’s Law)
- Massive data repositories (The unreasonable effectiveness of data)
- Increased battery density (Electrically-powered flight more feasible)



Assuring MLEC Challenge, Part I



Each MLEC is primarily statistical in nature

- Often non-deterministic
- Very large state spaces
- Lack of explanations for decisions
- Lack of assurance that we have sufficient data to meet space of requirements (underfitting)

Contrast this with:

MLEC

- Assurance processes requiring every possible path is explored during testing



Assuring MLEC Challenge, Part II



System (+Mission, + Environment)

MLEC

Assurance (and resulting certification) for aviation is NOT gained at the MLEC level.

Overall assurance must be argued at the systems level, and is heavily dependent on the mission and the environment.

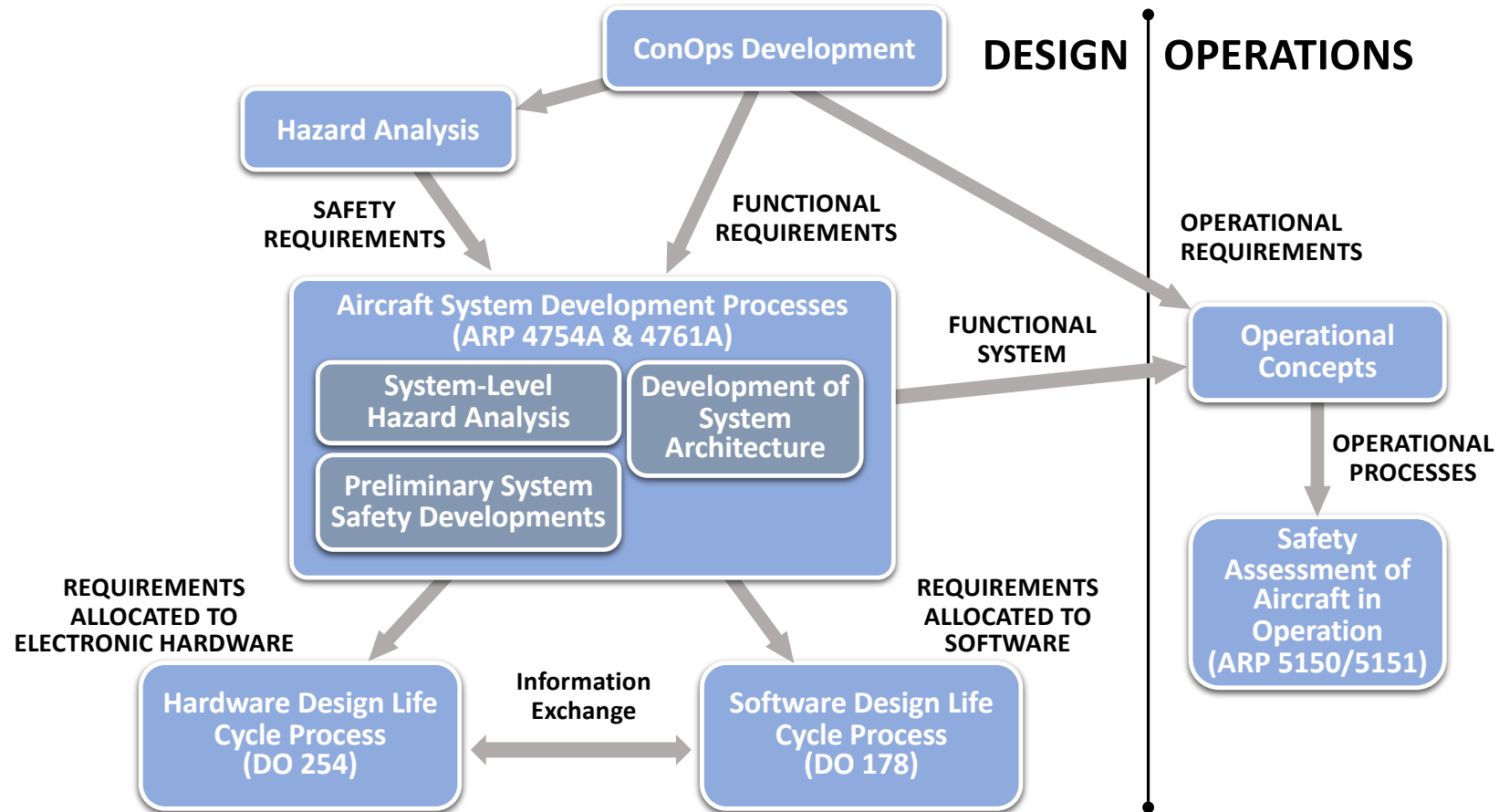
Currently, these arguments are based primarily on having followed processes.

The processes have been put in place to avoid mistakes we have seen in the past.

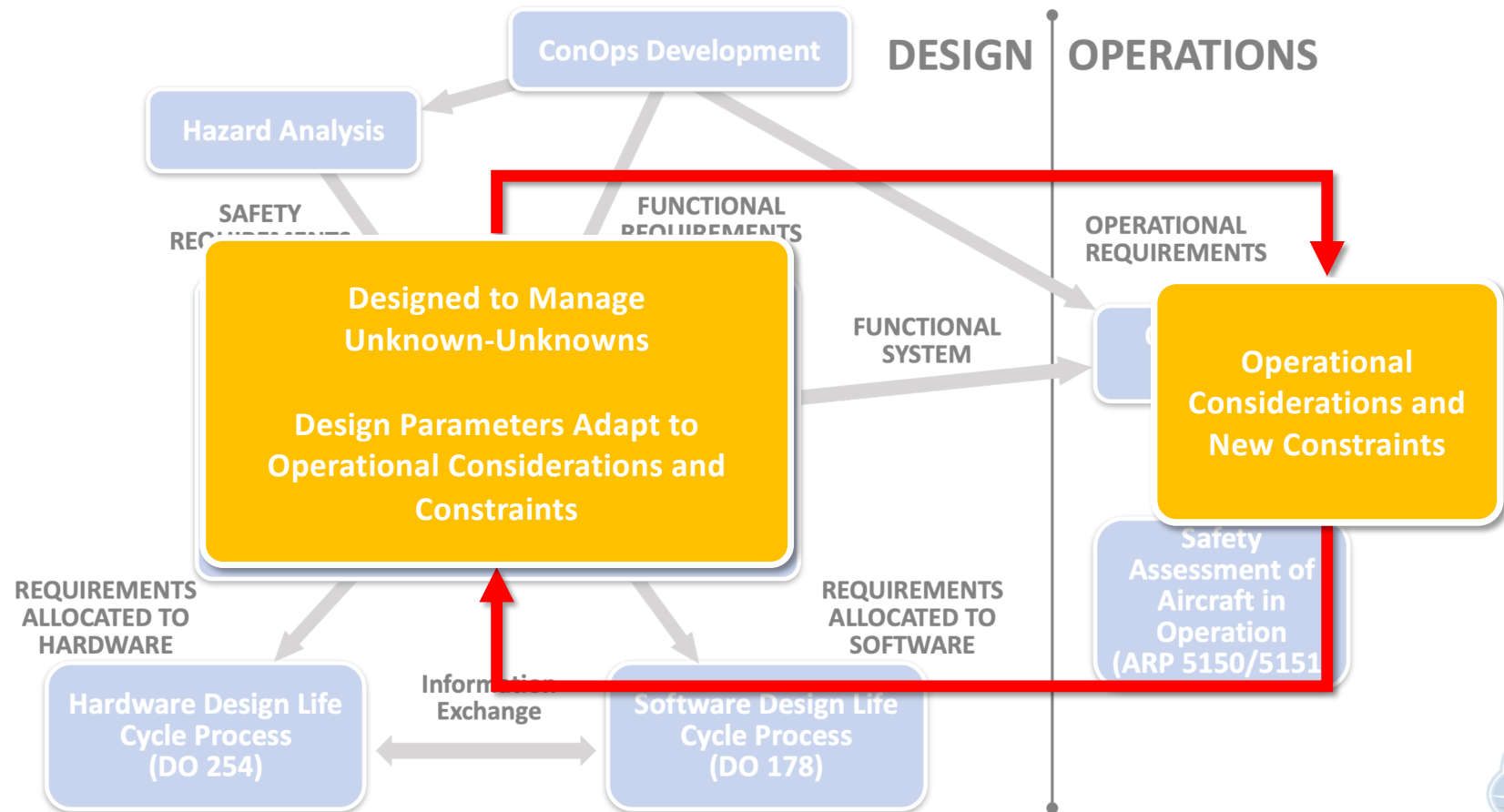


System-Wide Safety

Design Safety vs. Operational Safety Today



Design Safety vs. Operational Safety Tomorrow



Step One: Know Your MLEC!



Some MLEC more amenable to assurance

- Deterministic during operation (trained offline)
- Resulting outputs can be mathematically bounded given bounded inputs
- Some give very good confidence estimates in results as part of the algorithm

MLEC

More research is necessary to understand MLEC that are appropriate for use in safety-critical applications and the limits/bounds of those MLEC.



NASA Work: Know Your MLEC! (1 of 2)



Verification of Autonomous Cyber-Physical Systems: Cesar Munoz, Aaron Dutle, J Tanner Slagel, Mariano Moscato (NIA)
Formal verification of safety-critical algorithms for autonomous unmanned aircraft systems (working with **National Institutes of Aerospace**)

Verification of Numerical Intensive Systems: Cesar Munoz, Aaron Dutle, Laura Titolo (NIA), Mariano Moscato (NIA)
Verification of floating-point behavior of numerical intensive systems, such as those used in machine learning. (working with **National Institutes of Aerospace**)

Next Generation of Verification Systems: Cesar Munoz, J Tanner Slagel, Paolo Masci (NIA), Mariano Moscato (NIA)
Studies the use of machine learning in formal verification and the integration of formal models in software and hardware in the loop simulation platforms. (working with **National Institutes of Aerospace**)

Assured Machine Learning in Autonomous Systems: Cesar Munoz, J Tanner Slagel, Aaron Dutle, Swee Balachandran (NIA)
Investigates the safe use of ML techniques in ICAROUS. (working with **National Institutes of Aerospace**)

Deep Neural Network analysis: Corina Pasareanu (KBR), Divya Gopinath (KBR)
Formal analysis of deep neural networks and their activation patterns

NASA Work: Know Your MLEC! (2 of 2)



Plan Execution Safety Analysis: Cesar Munoz, Marco Feliu (NIA)

Investigating the encoding of ICAROUS autonomous plans in PLEXIL for verification. (working with **National Institutes of Aerospace**)

Stress testing for autonomy: Ritchie Lee

Apply stress testing to real projects with MLEC component (vision, obstacle avoidance) including industry and in-house case studies (**GE, Boeing**)

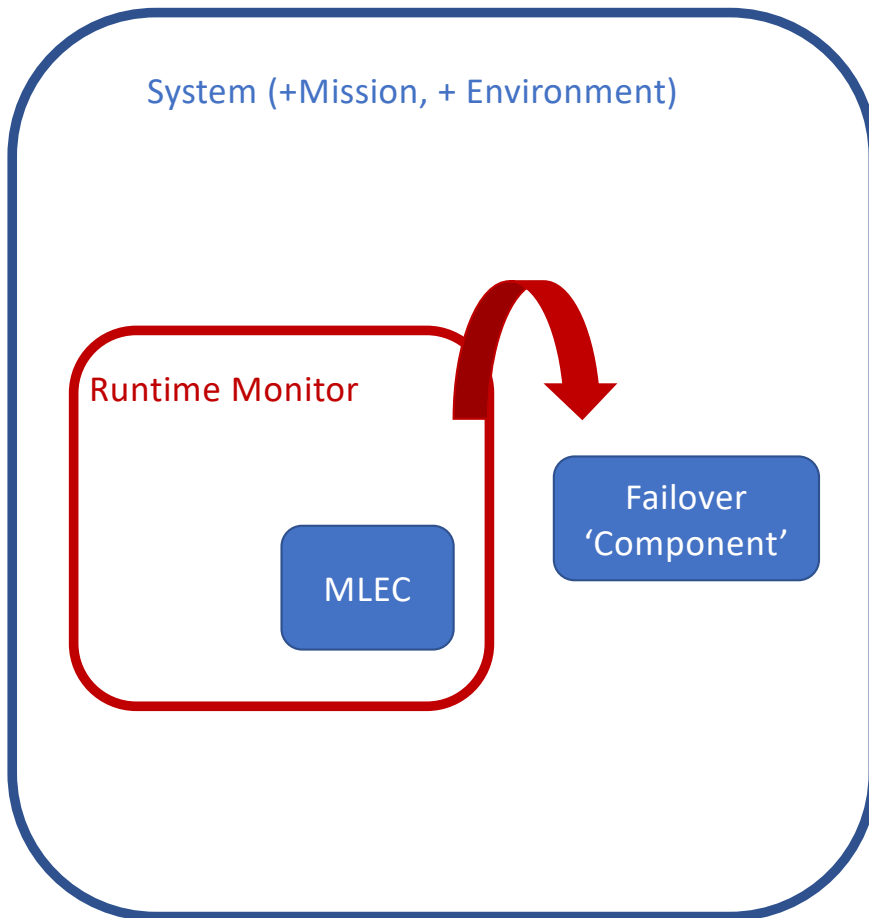
ML explainability: Adrian Agogino

Explain decisions made by neural networks. Evaluation is done using an image processing example (following a path). (**GE, Boeing**)

MLEC

More research is necessary to understand MLEC that are appropriate for use in safety-critical applications and the limits/bounds of those MLEC.

Step Two: Enforce the Limits



Highly-assured runtime monitor to

- Know when MLEC has exceeded its bounds (both inputs and outputs)
- Change mode of operation to a new mode of operation within safe limits

Note: the failover operation is likely less capable, meaning degraded mission capability

(e.g. SAFEGUARD, autonomous surface operations example)

More research is necessary to understand how to assure the runtime monitors to the required levels.



NASA Work: Enforce the Limits



Run Time Assurance (Component and Architecture Levels): Alwyn Goodloe, Ivan Perez (NIA), Wilfredo Torres, Cesar Munoz, Natasha Neogi

Explores and analyzes the use of run time monitors and run time assurance procedures as alternate assurance concepts. These tools may be used to check and/or arrest anomalous behavior of machine learning enabled components MLEC, highly automated or autonomous flight capabilities (supervised or unsupervised), other novel components or questionable architectural constructs that may manifest during the deployment phase applicable to eVTOL and UAM communities. An analysis has also been provided on distributed shields or a shielded learning approach. (working with **Galois, National Institutes of Aerospace, Collins Aerospace, Electron International, Barron Associates, Architecture Technology Corporation, UT Austin**)

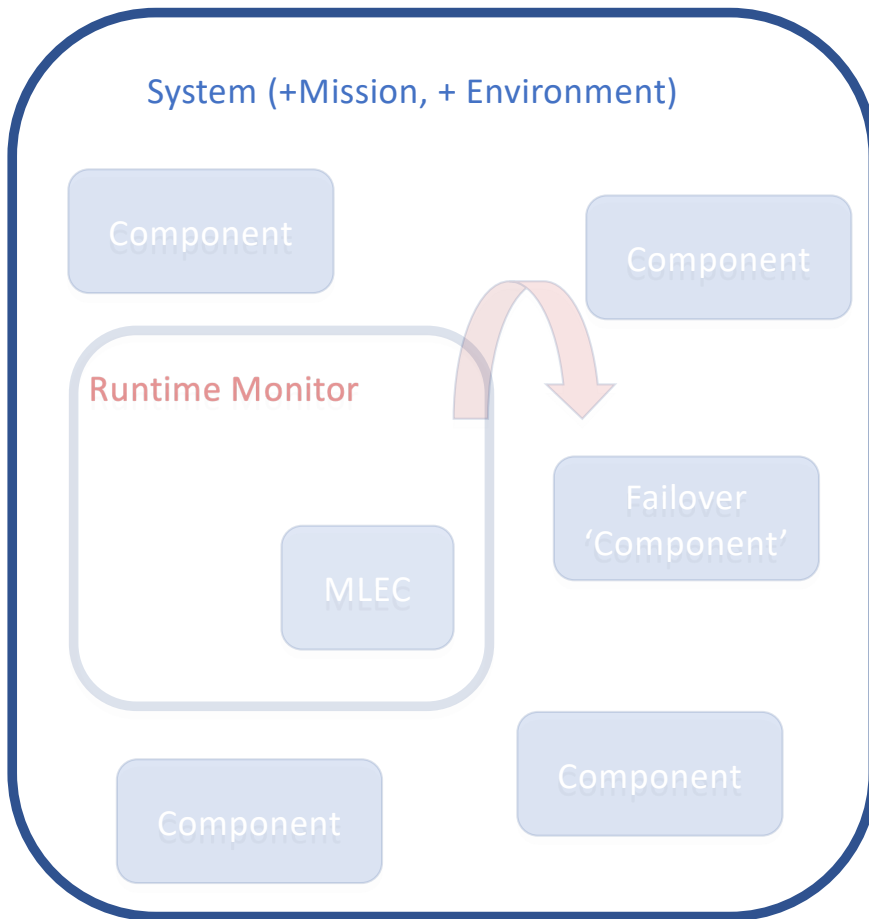
Run Time Monitoring for Autonomy: Aaron Dutle, Esther Conrad, Alwyn Goodloe, Laura Titolo (NIA), Ivan Perez (NIA), Johann Schumann (KBR), Nastaran Shafei (KBR), Peter Mehlitz (KBR)

Explores

- the generation of run-time monitors for ICAROUS from high-level requirements.
- the use of R2U2 for run-time monitoring of a Boeing use case.
- the use of MESA for an ADS-B use case.

(working with **National Institutes of Aerospace, Boeing**)

Step Three: Make System-Level Arguments



Need new patterns for system-level safety arguments that

- Take MLEC and runtime monitor reliability of evidence
- Understand the effect on safety of moving to the failover component as necessary
- Compose the modes of operation to make high-level safety arguments
- May need to integrate new argument patterns with traditional safety arguments and design patterns to understand safety for the whole system

More research is necessary to understand how to compose the evidence to make claims about system-level safety, and about how to have reliable contracts at the subsystem levels.



NASA Work: Make System-Level Arguments (1 of 2)



Certification Development: Michael Holloway, Mallory Graydon, Paul Miner, Alwyn Goodloe

Explores the use and evaluation of certification methodologies (Overarching Properties OPs, Automated Rapid Certification of Software ARCOS, etc.) to gain certification approval for use on civilian and/or military aircraft. The ARCOS approach intends to tackle the deficits of human evaluators for software as well as determine a principle means to decompose evaluations. This investigation also includes developing guidelines, particularly implementation verification, for certifying AI-enabled systems like neural networks, etc. (working with **Raytheon Technologies, Pratt & Whitney, DARPA, SAE G-34, Collins Aerospace, ONERA**)

Requirements for autonomy: Dimitra Giannakopoulou, Tom Pressburger, Matt Knudson, Anastasia Mavridou (SGT)

Develops tool (FRET) for eliciting and formalizing requirements of increasingly autonomous systems to support formal analysis by other tools. FRET comes with built-in analyses (e.g., requirement consistency) and validation of the formalization through visual diagrams. FRET is integrating with Co-Pilot to help support the use of formal monitors. (working with **GE, Boeing, Raytheon, and NASA(VIPER)**)

Operational Safety Cases (AdvoCATE): Ewen Denney (SGT), Ganesh Pai (SGT)

Develops a tool and methodology to build safety cases for complex aviation systems and facilitate inspection by regulatory agencies. The work is now evolving safety cases from aviation certification role to in-time dynamic safety dashboard for evaluating risks (including new ones) in operations. (working with **Boeing, GE, Lockheed Martin, and other businesses outside Aviation**)

Model Based Safety Analysis: Kurt Woodham, Michael Holloway

Explores the Property Model Methodology developed by Airbus (and other potential alternatives like the Systems-Theoretic Model) as a way to capture and validate system specification models and their subsequent decomposition. This task should provide benefit to NASA Agency MBSE initiatives and should reveal a process that is relevant to system-safety assurance that includes automation. (working with **Aerospace Vehicle Systems Institute AVSI, Boeing, Airbus, Collins Aerospace, Thales, GE, Honeywell and the University of Virginia**)

NASA Work: Make System-Level Arguments (2 of 2)



Assuring Increasingly Autonomous Systems with Non-Traditional Human-Machine Roles: Natasha Neogi

Explores and analyzes where the machine fits into the team in safety-critical operations and how to assure safety in human-machine teams with increasingly autonomous (IA) capabilities using IA examples relevant to Urban Air Mobility (working with **Pennsylvania State University, Iowa State University, Collins Aerospace, Florida Institute of Technology, Soar Technologies**)

System Theoretic Process Analysis: Natasha Neogi, Mallory Graydon

Explores the use of novel hazard analysis techniques, such as STPA (Systems-Theoretic Process Analysis), for complex and increasingly autonomous systems. STPA is a powerful hazard analysis technique based on STAMP (Systems-Theoretic Accident Model and Processes) which is an accident causality model based on systems theory and systems thinking that integrates engineering analysis causal factors such as software, human decision-making and human factors, new technology, social and organizational design, and safety culture into its framework. (working with **Massachusetts Institute of Technology**)

Requirement Tracing Validation with Drishti: Nija Shi, Kevin Seywald (KBR), Carlos Paradis (KBR)

Uses NLP to address deficiencies to meet required safety standards in the development process of complex aerospace systems. For example, it can validate requirement traceability, find missing connections between requirements and tests/code, and help validate consistency between project documentation and standards.

Resilience of complex systems: Hannah Walsh, Daniel Hulse, Sequoia Andrade (KBR)

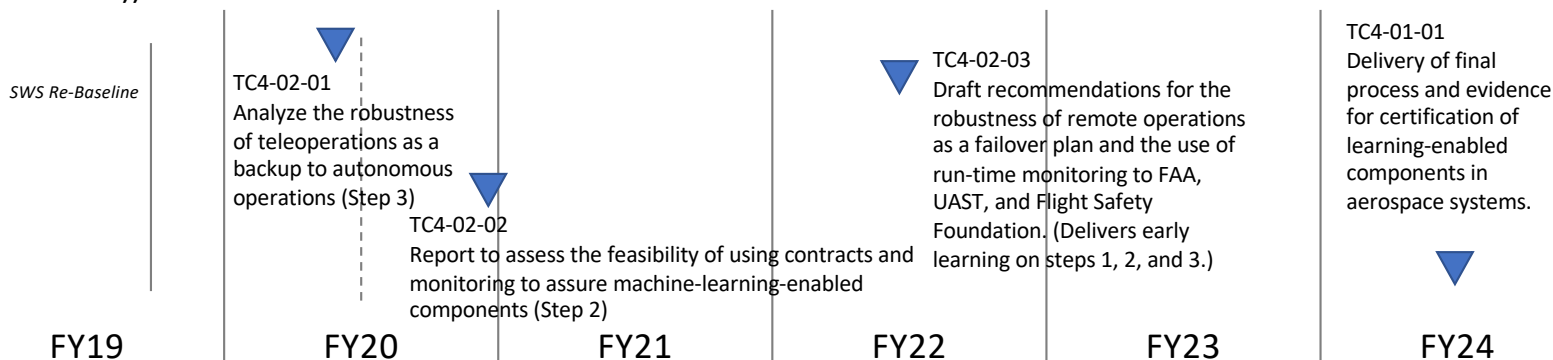
Evaluates resiliency metrics and performs risk analysis for complex aerospace systems. Creates a design failure taxonomy using data mining of NASA's Lessons Learned Information System (LLIS).

Progress Indicator Chart (FY20 Update)



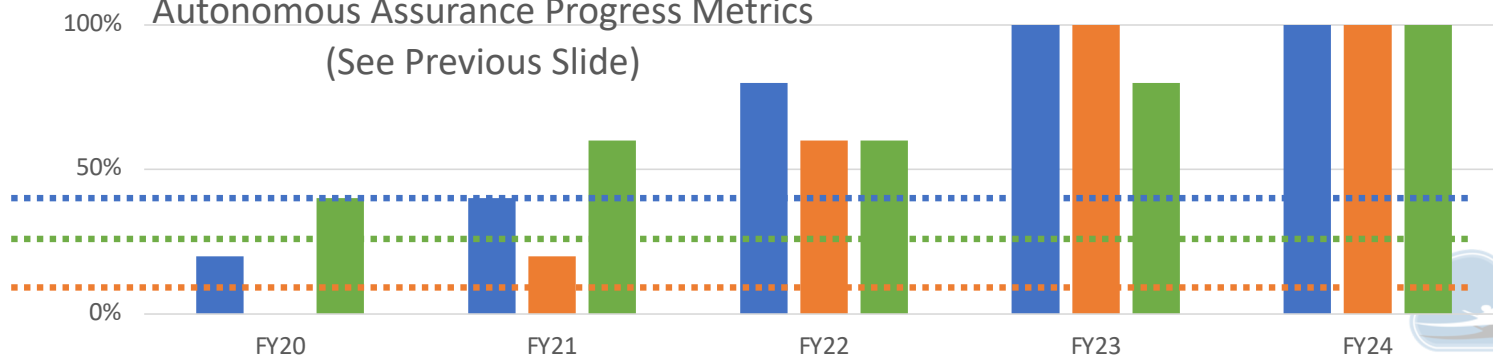
Thrust 5 Critical Commitment: Demo of an end-to-end safety-assured and validated IASMS framework that enables *monitor, assess* and *mitigate* functions in a rapidly changing and uncertain environment (TC-4: Develop a prototype certification process for autonomous aerospace systems that will be NASA's recommendation to the FAA, UAST, the Flight Safety Foundation and other standards committees (Completes FY24, closeout FY25))

Key Milestones & Schedule



Autonomous Assurance Progress Metrics (See Previous Slide)

- Step One: ML-Enabled Component (MLEC) Selection
- Step Two: Rigorous Assurance of Run-Time Monitors
- Step Three: Assurance of Systems with MLEC and Failover Plans



Dotted lines indicate progress as of 9/30/2020



Progress Indicators



STEPS					
Autonomous System Certification Process Development Criteria					
1. ML-Enabled Component (MLEC) Selection	1	2	3	4	5
	Completing the following as 20% each: <u>1. Complete Survey of Potential MLEC</u> 2. <i>Evaluate Limits of Formal Analysis of Select ML Algorithms</i> . 3. Evaluate Assurance Method of Select AI Components 4. Evaluation of Generalization and Explainability of ML Algorithms 5. Demonstration of Assured ML Enabled Component				
2. Rigorous Assurance of Run-Time Monitors	1	2	3	4	5
	Completing the following as 20% each: 1. Demonstration of a Run-Time Monitor In-Flight (R2U2 – COPILOT) 2. <i>Formal Assessment of Run-Time Monitoring as an Alternative Assurance Technique</i> 3. Distributed Run-Time Assurance Monitors within Reference Framework (ICAROUS) 4. Untrusted Components Assured by a Run-Time Monitoring Framework 5. ML Assured by a Run-Time Monitoring Framework				
3. Assurance of Systems with MLEC and Failover Plans	1	2	3	4	5
	Completing the following as 20% each: 1. Deliver Guidance on Overarching Properties (OP) to OP Working Group and FAA. 2. <u>Assessment of Robustness of Teleoperations as a Backup for Autonomy</u> 3. Demonstrate Safety Case Dashboard for Risk Based Decision Support 4. Complete Industry Use Case Leveraging OP process. 5. <i>Support Industry Partner for Certification of MLEC</i>				

Underlined elements are complete.

Italicized elements have significant work towards completion.

