AUTONOMOUS SYSTEM V&V
A SAFETY ASSURANCE APPROACH

Martin S. Chizek
Engineering Fellow / Product Safety Officer

BACKGROUND AND PROBLEM STATEMENT

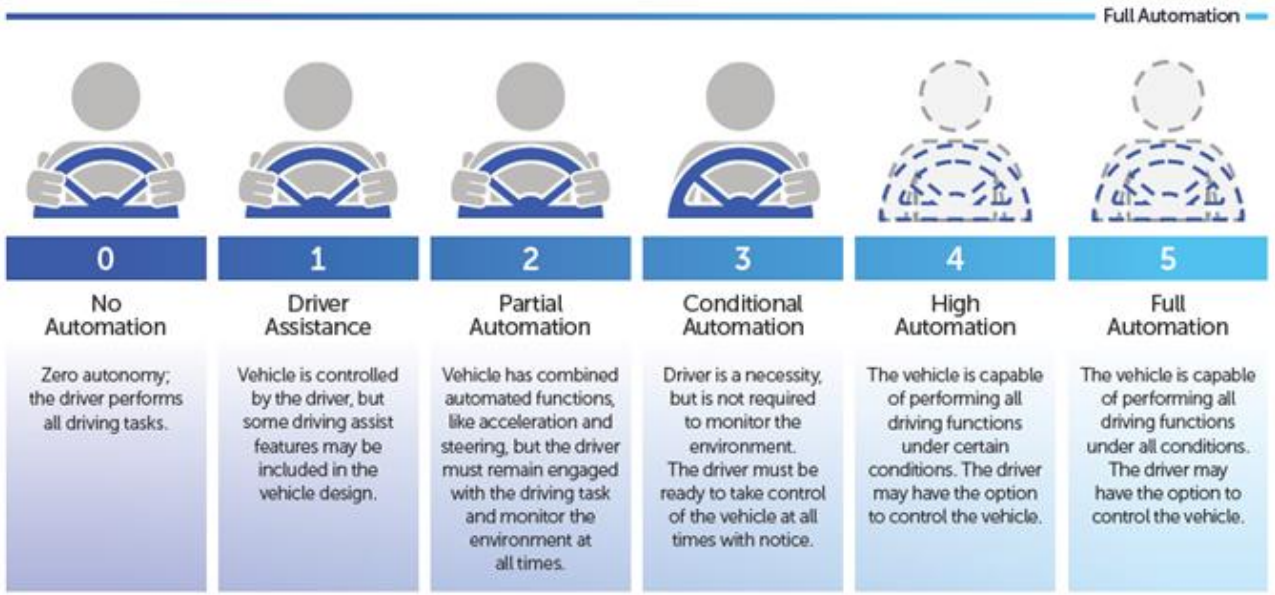
- Autonomy involves highly complex systems with millions of SLOC and is expected to be used in complex environments with high uncertainty.
- Requirements must be Verified and Validated (V&V) to instill **trust** of autonomous systems.
- No current infrastructure exists that can V&V complex autonomous systems:
  - Unpredictable environments are difficult to define in requirements.
  - There is a near-infinite state space of fielded environments.
  - Non-deterministic responses are hard to evaluate via bounded testing.
  - Learning algorithm updates potentially invalidate prior testing.

Verification - The process of providing objective evidence that the system, software, or hardware conform to requirements. "Did we build the right system?"

Validation - The process of providing evidence that the system, software, or hardware solves the right problem and satisfies intended use and user needs. "Did we build the system right?"

LEVELS OF AUTONOMY

- The most widely held definitions for levels of automation was developed by SAE International for the automotive domain. Unmanned Aerial Vehicles (UAVs) may adopt NIST Autonomy Levels.
- SAE Level 3 assumes human driver is available to take over driving responsibilities.
  - Drivers' ability to do so is limited by humans' capacity for staying alert.



SAE International

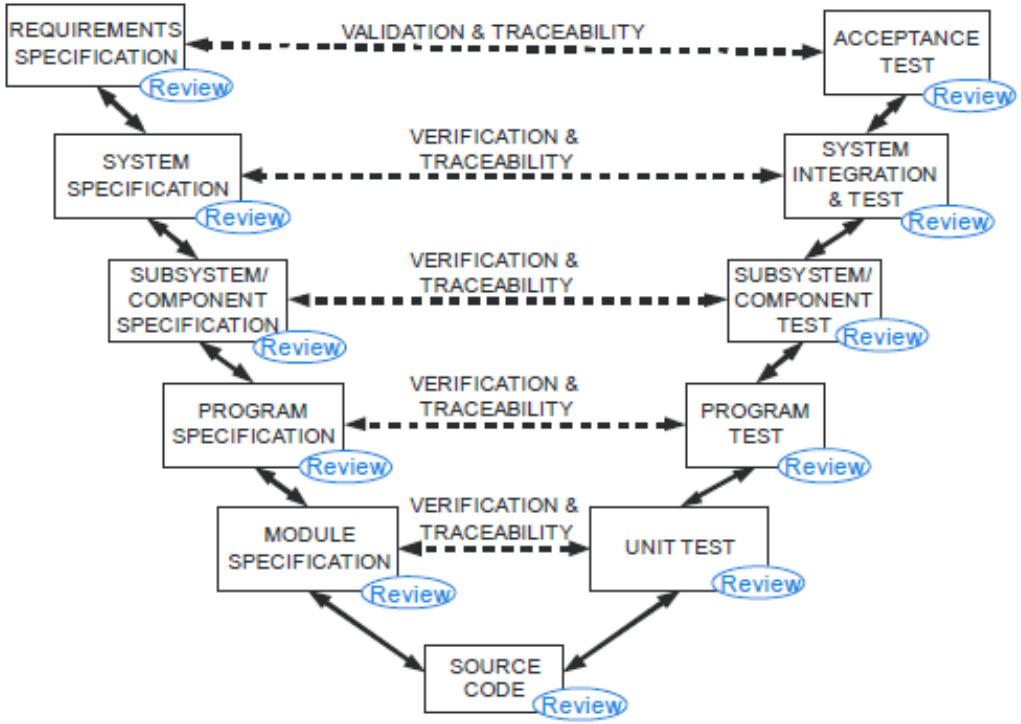
Autonomy Levels	
Level	**Description**
1	Remote Control
2	Automatic Flight Control
3	System Fault Adaptive
4	GPS Assisted Navigation
5	Path Planning and Execution
6	Real Time Path Planning
7	Dynamic Mission Planning
8	Real Time Collaborative Mission Planning
9	Swarm Group Decision Making
10	Full Autonomous

Autonomy Levels for Unmanned Systems
US National Institute of Standards and Technology (NIST)

CHALLENGES OF AUTONOMY V&V

Incomplete or Vague Autonomy Requirements

- Combinations of adverse events and driving conditions are too numerous for a classical written requirements specification. Thus, a classical "V" process is not scalable to autonomous vehicles.
- Even if requirements can be defined, autonomous systems entails highly complex software with non-deterministic and adaptive behavior, so verification that the requirements are met is difficult.
- Test and evaluation must address the inputs, processes and states, and outputs for all autonomous functions (e.g. perception, reasoning, learning, decisions, and behavior). It is difficult to define requirements that are clear and testable.
- A complete set of behavioral requirements needs to be developed before behavioral correctness can be measured to provide pass/fail criteria for testing.



CHALLENGES OF AUTONOMY V&V (CONT'D)

System Complexity

- Increased complexity in both the autonomy software and the unpredictable environment makes testing alone impossible.
- Complex adaptive system behaviors cannot be predicted, and it is impossible or impractical to analyze and test.
- One must rely on a rigorous, structured design assurance process of the item, in addition to comprehensive tests and analyses.

State Space Explosion

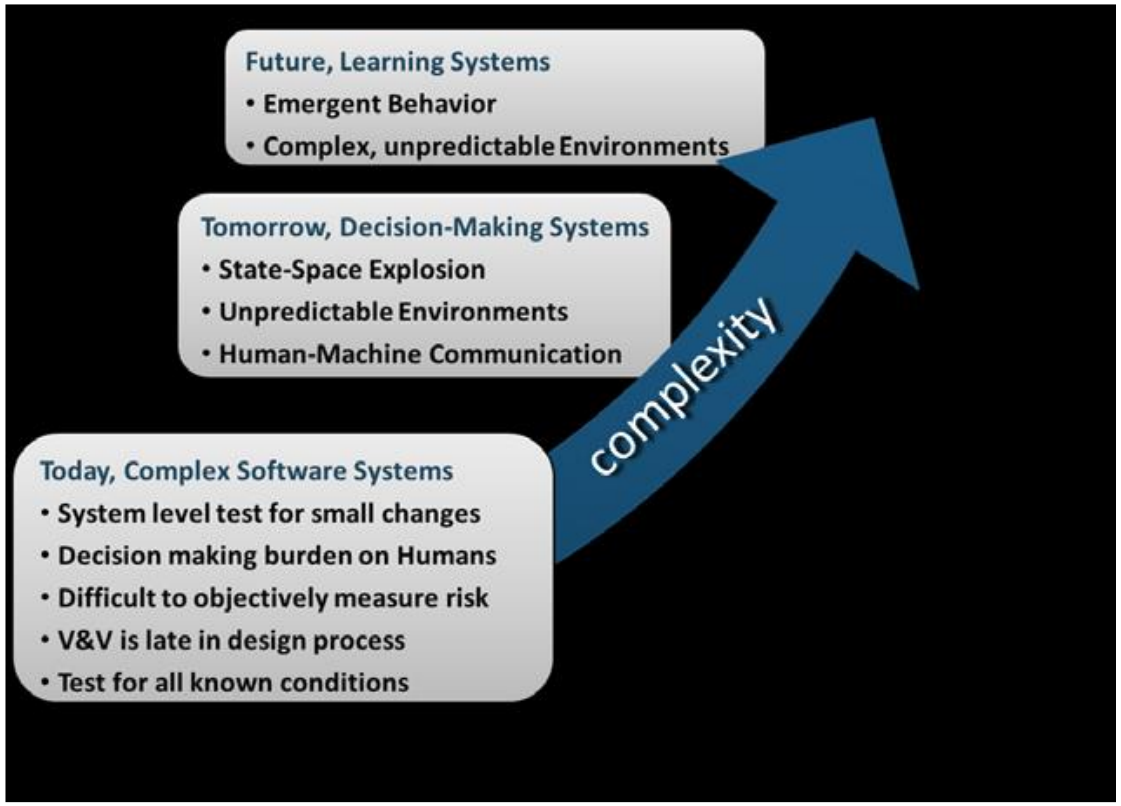
- State-space explosion describes how the possible number of conditions, factors and interactions are so exponentially large that it becomes impossible to test all combinations.
- The algorithmic decision space is non-deterministic, i.e., the output cannot be predicted due to multiple possible outcomes for each input.
  - Exercising a particular specific edge-case is difficult since it may require only if it receives a very specific sequence of inputs from the world.
  - It can be difficult to evaluate whether test results are correct or not, because there is no unique correct system behavior for a given test case.

Highly Complex Systems and Autonomous Systems Face Many of the same V&V Challenges

CHALLENGES OF AUTONOMY V&V (CONT'D)

Emergent Behavior

- Interactions between systems and system factors may generate unintended consequences.
- Because a system is anticipated to learn and change over time, autonomous systems may react differently to the same inputs.
- This non-deterministic behavior also means that one successful test does not guarantee that the system will pass the same test on the next test run.



CHALLENGES OF AUTONOMY V&V (CONT'D)

Human-Machine System

- Human Machine teaming issues were identified as the largest barrier to developing standards for certifying increasingly autonomous systems.
- Handoff, communication, and interplay between operator and autonomy are key enablers for the trust and effectiveness of an autonomous system.
- Most civil aviation applications will likely be implemented as a combination of autonomous software with real-time mission management and oversight by a human operator.
  - Certification will required V&V of the total human/machine system.

CHALLENGES OF AUTONOMY V&V (CONT'D)

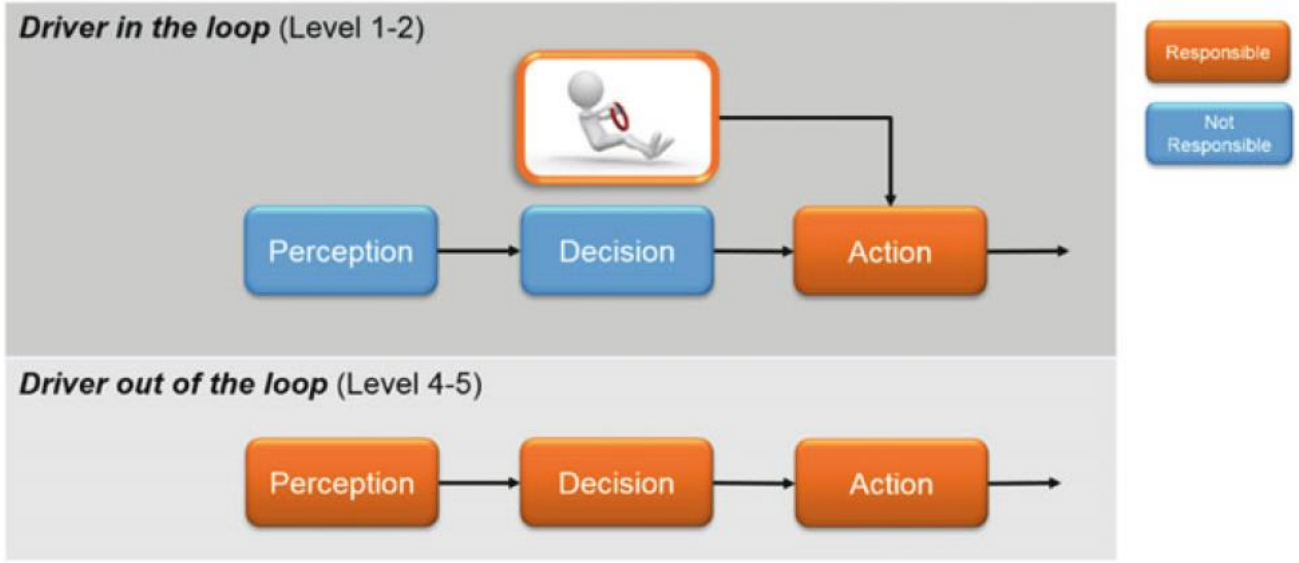
Trust and Certification

- Trust has been identified as a major challenge in the development and implementation of autonomy in Civil Aviation.
- Without such trust, autonomous systems will not be adopted except in extreme cases such as missions that cannot otherwise be performed.
- Since autonomous systems are often safety-critical (e.g., flight control) systems, they have to adhere to strict certification standards.
- Certification of aviation systems that combine autonomous software and a human operator will required V&V of the total human/machine system.

AUTONOMY V&V – DOMAIN-SPECIFIC CHALLENGES

Automotive

- SAE Level 3 systems expect human drivers to monitor for failures in autonomy, and take over driving responsibilities.
  - Drivers' ability to do so is limited by humans' capacity for staying alert.
- SAE Levels 4/5 take the driver out of the loop.
  - The system must be able to assess and safely handle every situation it encounters.
  - This requires V&V of not only the autonomous driving function, but also the capability of the vehicle to determine in advance if it cannot handle a situation.
- Basing a safety argument on accumulating road miles is impractical, since it takes millions of miles to make a credible statistical argument.



AUTONOMY V&V – DOMAIN-SPECIFIC CHALLENGES (CONT'D)

Aviation

- How do we assure that autonomous (adaptive / nondeterministic) systems that can modify their behavior are safe and reliable for Civil Aviation?
- A barrier to the introduction of autonomy into the National Airspace System (NAS) is lack of trust in autonomy by pilots, controllers, regulators, and the public.
- The FAA is cautious in certifying autonomous flight systems.
  - These systems are especially difficult to V&V, since they can be nondeterministic and possess learning capabilities that modify their behavior to address unanticipated situations.
  - Yet, highest levels of V&V analyses are required for flight critical certification.

TRUSTED AUTONOMOUS SYSTEM

- Trust has been identified as a major challenge in the development and implementation of autonomy in Civil Aviation.
- For Department of Defense (DoD), the decision to deploy autonomous systems must be based both on **trust** that they will perform effectively in their intended use, and will not result in high-regret, unintended consequences.
- Developing a trusted autonomous system requires addressing:
  - System Certification (or equivalent) by an independent third party,
  - System Integrity (SIL, DAL, LOR),
  - System Dependability (Safety, Reliability, Availability, and Cyber-security),
  - System Assurance (Design Time Assurance, Run Time Assurance).

TRUSTED AUTONOMOUS SYSTEM (CONT'D)

System Integrity

- IEC 61508 sets a Safety Integrity Level (SIL) according to the level of risk reduction.
- ISO 26262 Automotive ASIL incorporates the concept of controllability of the vehicle.

Software Assurance

- The concept of moderating the software assurance approach according to 'risk' is common across the standards.
- In DO-178C, the rigor of the process followed in developing the software varies according to the assurance level associated with that software.
- For DoD, each Safety Function must be designed, coded, tested, and verified with the appropriate Level of Rigor (LOR) to provide the assurance of safe software.

TRUSTED AUTONOMOUS SYSTEM (CONT'D)

Dependability

- Autonomous vehicles are safety-critical systems that require a high level of dependability, a term covering Reliability, Availability, Safety and Security.

Design Time Assurance (DTA)

- DTA is the process of performing V&V analysis and testing of software during design and development.
- If a set of software can be fully V&V'd at design time, then that software is considered ***trusted*** for live operation
- If a set of software cannot be fully V&V'd (due to its complexity, nondeterminism, etc.), then that software is considered ***untrusted*** for live operation.

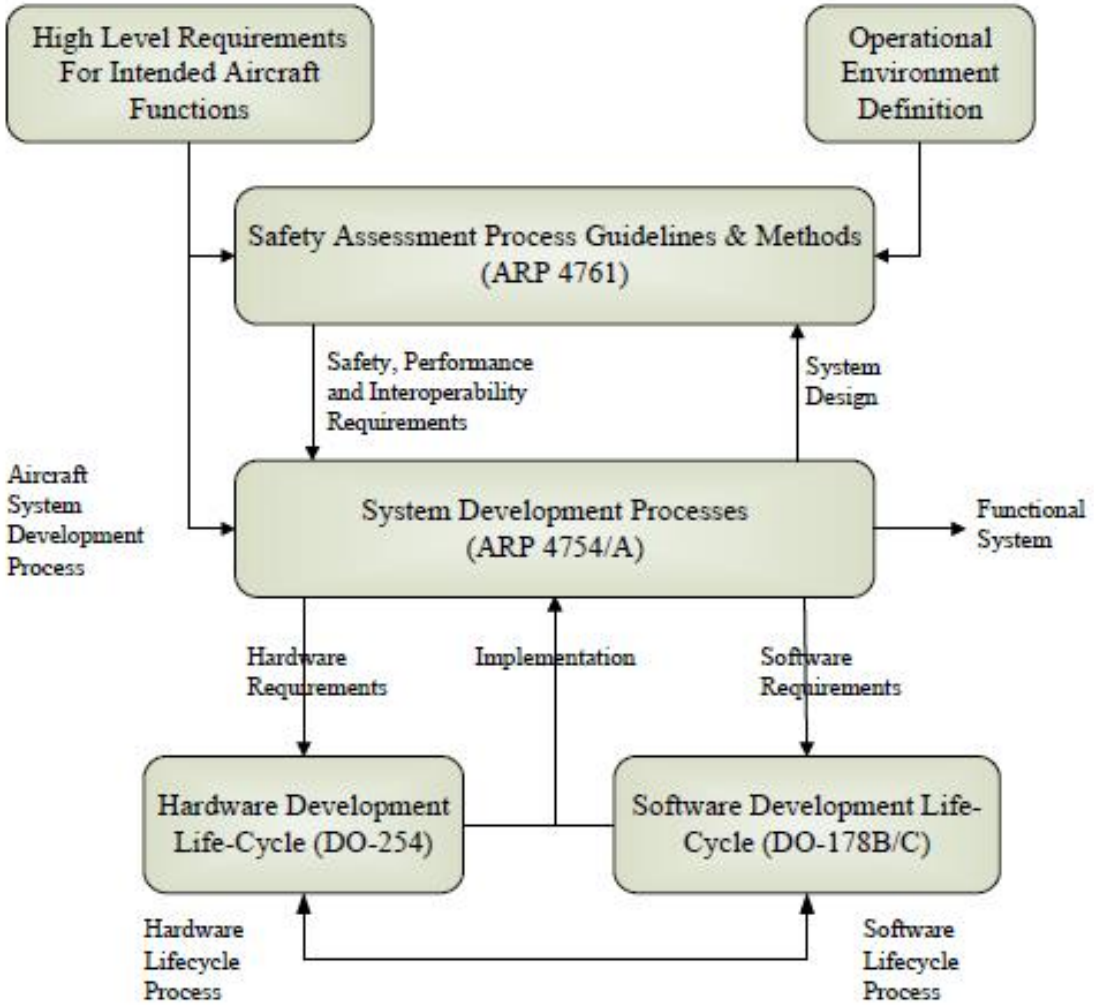
Run Time Assurance (RTA)

- RTA is the process of monitoring a system containing untrusted software during runtime to determine if it is operating correctly. If not operating correctly or anomalous or unsafe behavior is detected, then the RTA system activates recovery action to ensure continued safe or correct operations.
- The reversionary software would be able to continue safe operation of the system, but at a reduced level of functionality. In general, the reversionary software provides a "recover and return to base" capability.
- The goal of run-time verification is to detect erroneous executions that potentially could violate a safety condition before the controller produces an erroneous output and commands the vehicle with this erroneous output.

CURRENT AUTONOMY SYSTEM SAFETY PROCESS

U.S. Federal Aviation Administration (FAA)

- Today safety-critical flight systems are **trust**ed because they have met rigorous standards and certification criteria regarding **flightworthiness**, structural **integrity** and **safety-assured** flight controls under nominal conditions.
- Certification to DO-254 and DO-178C is an accepted manner of instilling **trust** in new, unproven, complex, software-intensive aviation systems.
- Unmanned aircraft (UAVs) will be subject to the same regulatory processes as manned aircraft and will therefore need to be certified.
- Complying with these standards is difficult for UAVs, since there is no fallback option to a trained human pilot, which is usually an accepted safety layer for the automation of high-level functions.

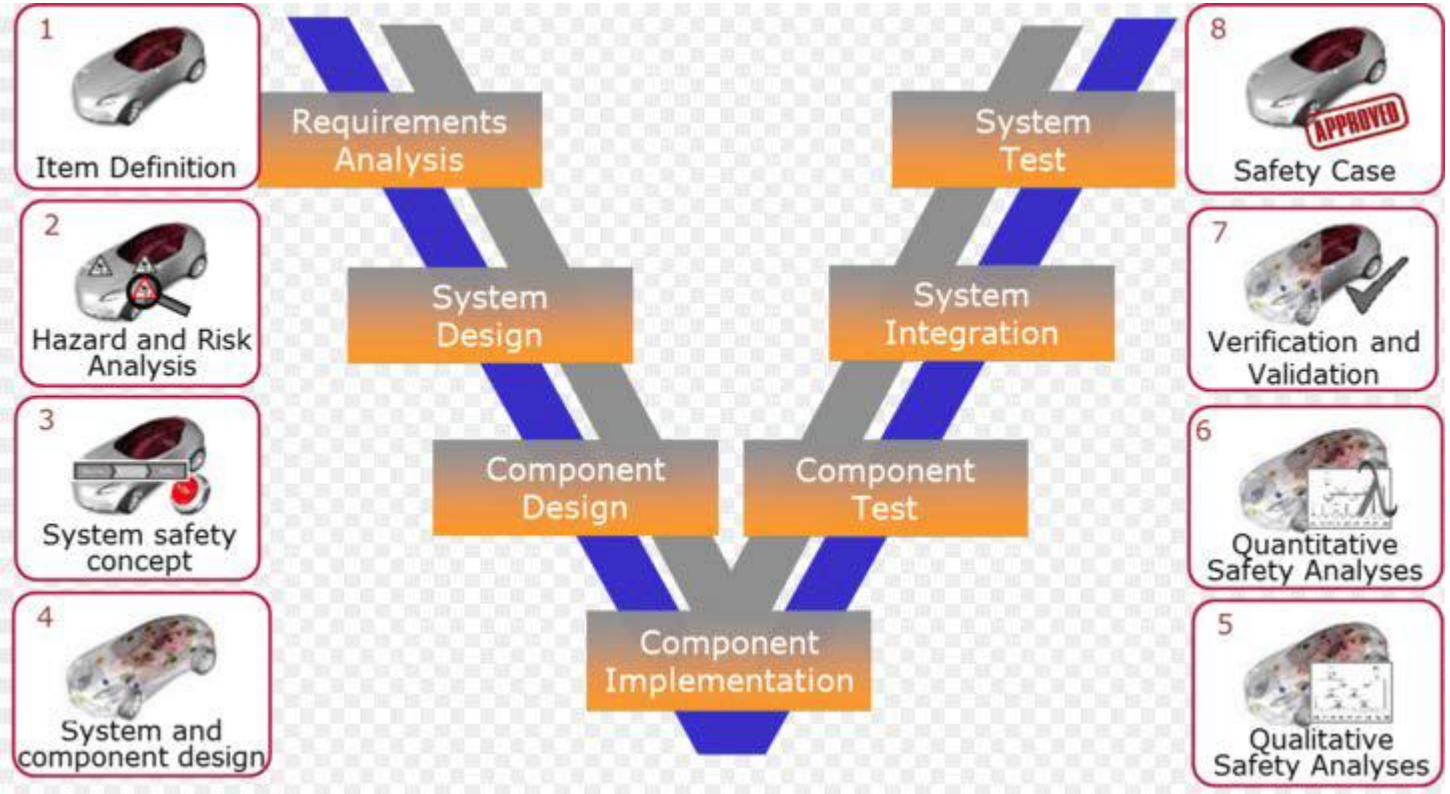


CURRENT AUTONOMY SYSTEM SAFETY PROCESS (CONT'D)

European Union (EU)

ISO 26262 *Functional Safety* ensures that the system works correctly in response to its environment.

- ISO 26262 describes the safety requirements for a corresponding Automotive Safety Integrity Level (ASIL) to avoid and mitigate the identified risks.
- The ASIL consists of severity, probability of exposure, and controllability (to which degree can harm be avoided or mitigated.



CURRENT AUTONOMY SYSTEM SAFETY PROCESS (CONT'D)

U.S. National Highway Traffic Safety Administration (NHTSA)

- NHTSA allows the introduction of new motor vehicles or motor vehicle technologies into the vehicle fleet, provided that those vehicles and technologies meet existing Federal Motor Vehicle Safety Standards (FMVSS).
- NHTSA's statute creates a self-certification system of compliance, in which vehicle and equipment manufacturers certify that their products meet applicable standards.
- NHTSA chooses vehicles and equipment from the fleet to test for compliance, and pursues enforcement actions when the Agency finds either a non-compliance or a defect posing an unreasonable risk to safety.
- All major U.S. automotive manufacturers are attempting to follow ISO 26262 as a method self-certification.

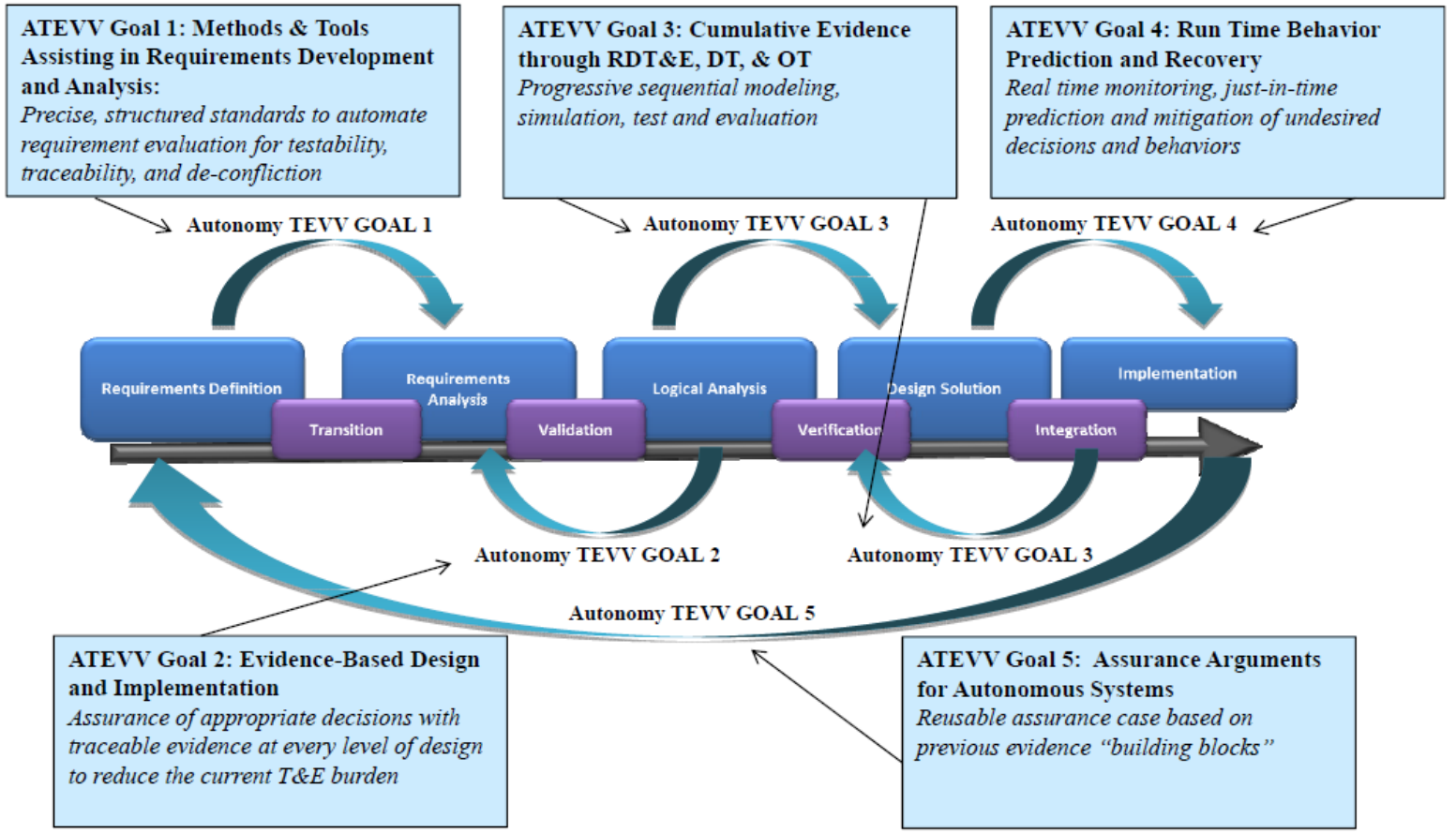
PROPOSED SOLUTION

- **Mitigation of Autonomy Requirements Complexity**
- **Autonomy Architecture Risk Mitigation**
- **Test and Verification Strategies**
- **Safety Assurance Argument for Certification**

PROPOSED SOLUTION

Mitigation of Autonomy Requirements Complexity

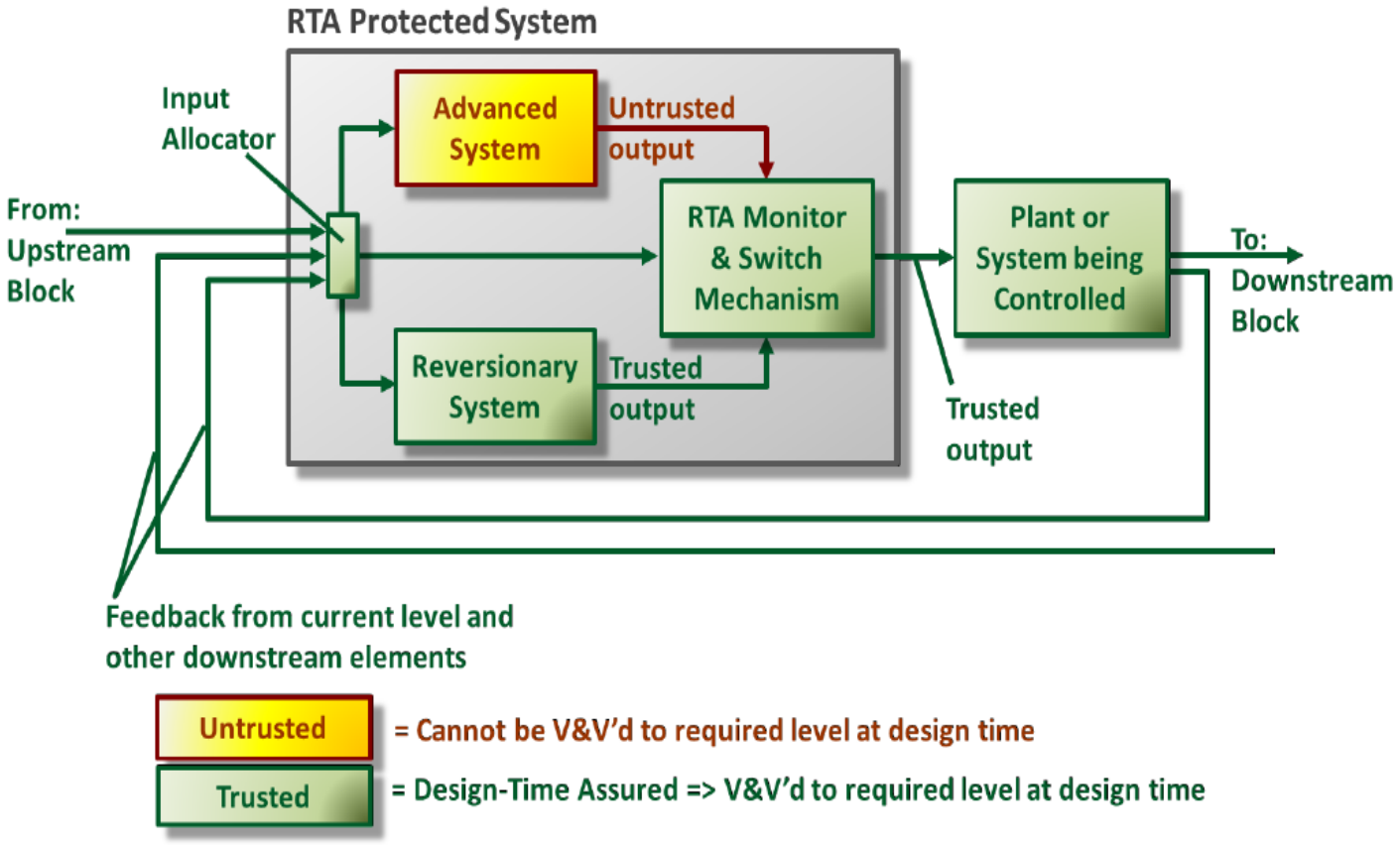
- Break down the requirement, system, and/or mission into smaller pieces, which can then be translated into rigorously quantifiable statistical designs.
- Express the functional and safety specification of the system in a form that can be checked at run-time.
- Create a set of purely safety requirements that define what “safe” means for the system.
- Flexibility in requirements, such as the ability to reason in certain settings, or negative requirements.
- Functional requirements can be allocated to a low-ASIL Actuator functional block, while safety requirements can be allocated to a high-ASIL Monitor.



PROPOSED SOLUTION (CONT'D)

Architecture Risk Mitigation

- Partitioning / Bounding autonomy functions
- Monitor/Actuator Architecture
  - Safety monitor behavior of autonomy components
  - Redundancy, Diversity and Data Integrity design features
- Run Time Assurance
  - Fail-Operational
  - Fallback with Limited Functionality



Conceptual *Run-Time Assurance* Protected System to bound behavior of untrusted advanced system

PROPOSED SOLUTION (CONT'D)

Test, Verification and Validation Strategies

- **Test Methods**
  - Use Formal Methods to prove the absence of faults in specifications and/or code.
  - Use Simulation to exercise as much of the state space as possible.
  - Consider probabilistic test results and risk assessments versus a quantifiable assessment of risk.
  - Instead of a test phase at the end of development, perform test and simulation throughout system life.
  - Instead of testing to validate a performance metric or particular behavior, test to ensure an unacceptable behavior will not arise at runtime under a range of conditions.
- **Phased Deployment**
  - Pursue autonomy in a step-wise manner; autonomy can be gradually introduced (no autonomy, partial autonomy, controlled autonomy, full autonomy).
- **Fault Injection**
  - Perform fault injection as part of a strategy of attempting to falsify claims of safety. This involves not only simulating objects for primary sensor inputs, but also inserting exceptional conditions to test the robustness of the system (e.g., inserting invalid data into maps).

PROPOSED SOLUTION (CONT'D)

Safety Argument Approach (Argument Based Assurance)

- Developers should create a safety assurance argument due to the inherent complexity of an autonomous vehicle, the lack of full requirements, and the inability to perform 100% testing.
- Use an approach based on evidence, in which one makes claims and assumptions, and then one uses an architecture, arguments, and evidence obtained using formal method tools to build assurance.
- A safety argument can be created based on the autonomy system having a detector that realizes when it is malfunctioning or has encountered a gap in its requirements. This would make the fault detector itself high-ASIL, but might permit normal autonomy functions to be low-ASIL.
- The safety argument developed for the software should consider traceability and verification of software requirements. Much of the evidence generated through following a standard such as DO-178C may be relevant to supporting claims made in the argument.

PROPOSED SOLUTION (CONT'D)

Safety Case Approach

- A software safety case comprises both product-based arguments and process-based arguments.
- Compile reusable evidence building blocks.
- A valid safety case for an autonomous vehicle consists of several inter-dependent components, namely:
  - Safety Target or Integrity Level that must be met to assure vehicle safety,
  - Evidence for the safety target obtained from study, analysis and test of the vehicle system,
  - A logical representation of the argument evidence by means of Fault Tree Analysis or Goal Structured Notation (GSN),
  - Context identifying the basis for the arguments presented.

QUESTIONS?