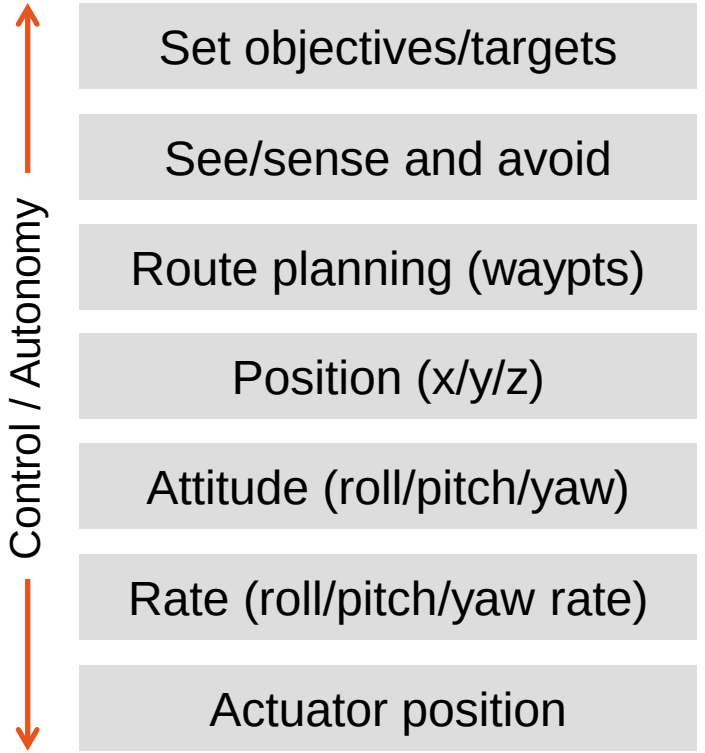
SAFETY, VERIFICATION, CERTIFICATION

ASSURED AUTONOMY WORKSHOP #1
16-17 OCTOBER 2019
DR. DARREN COFER

Collins Aerospace

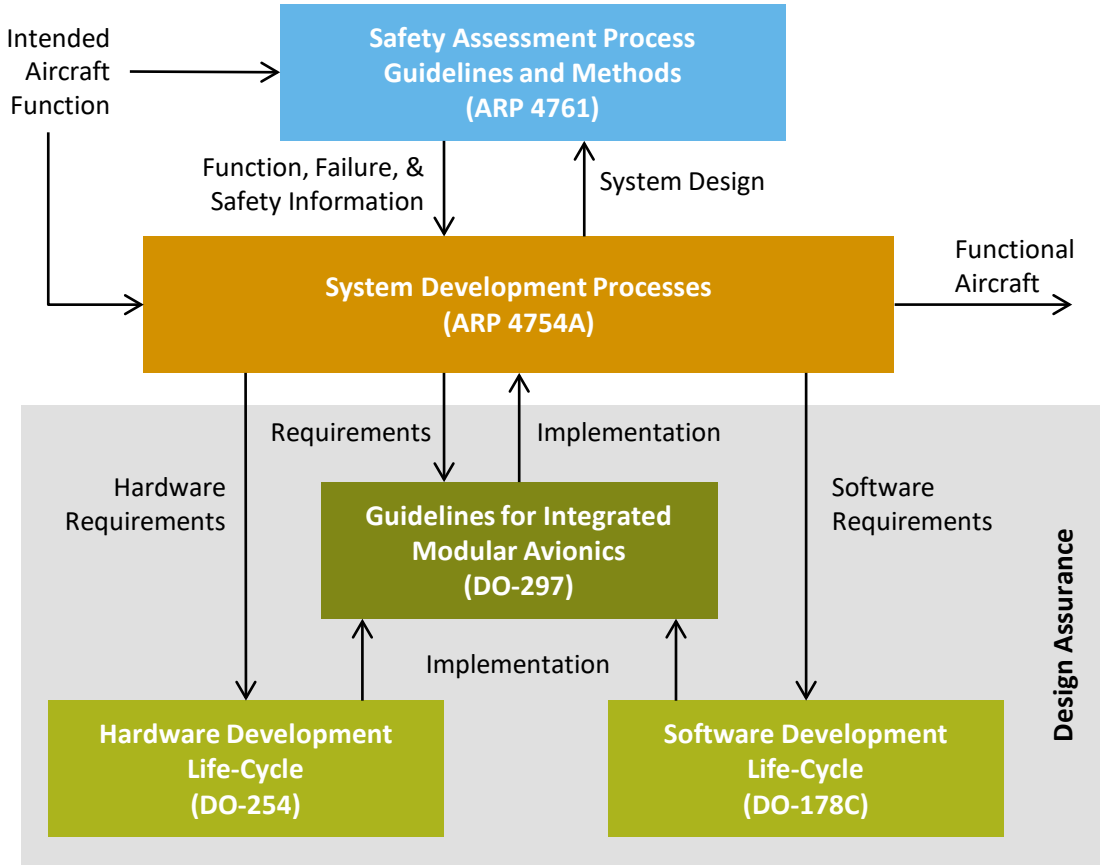
1. HOW IS "ASSURED AUTONOMY" INTERPRETED IN YOUR FIELD?

- Autonomy
  - Control at different levels – when is it autonomy?
  - Reduced crew, single-pilot operations, unmanned…
- But autonomy is not the issue
  - New methods/algorithms: AI, ML, DNN…
  - Perception applications
- Assured
  - Compliance with Airworthiness Regulations using accepted means of compliance (DO-178C)



ASSURANCE TODAY: CIVIL CERTIFICATION PROCESS

CFR Title 14 Part 25 Airworthiness Standards: Transport Category Airplanes



Means of Compliance

Part 23: small aircraft
Part 25: transport
Part 27: rotorcraft
Part 107: sUAS

2. KEY ASSURANCE CHALLENGES

Implementation

Verification

Requirements

https://xkcd.com/1838/

DO-178C VERIFICATION

REQUIREMENTS-BASED TESTING

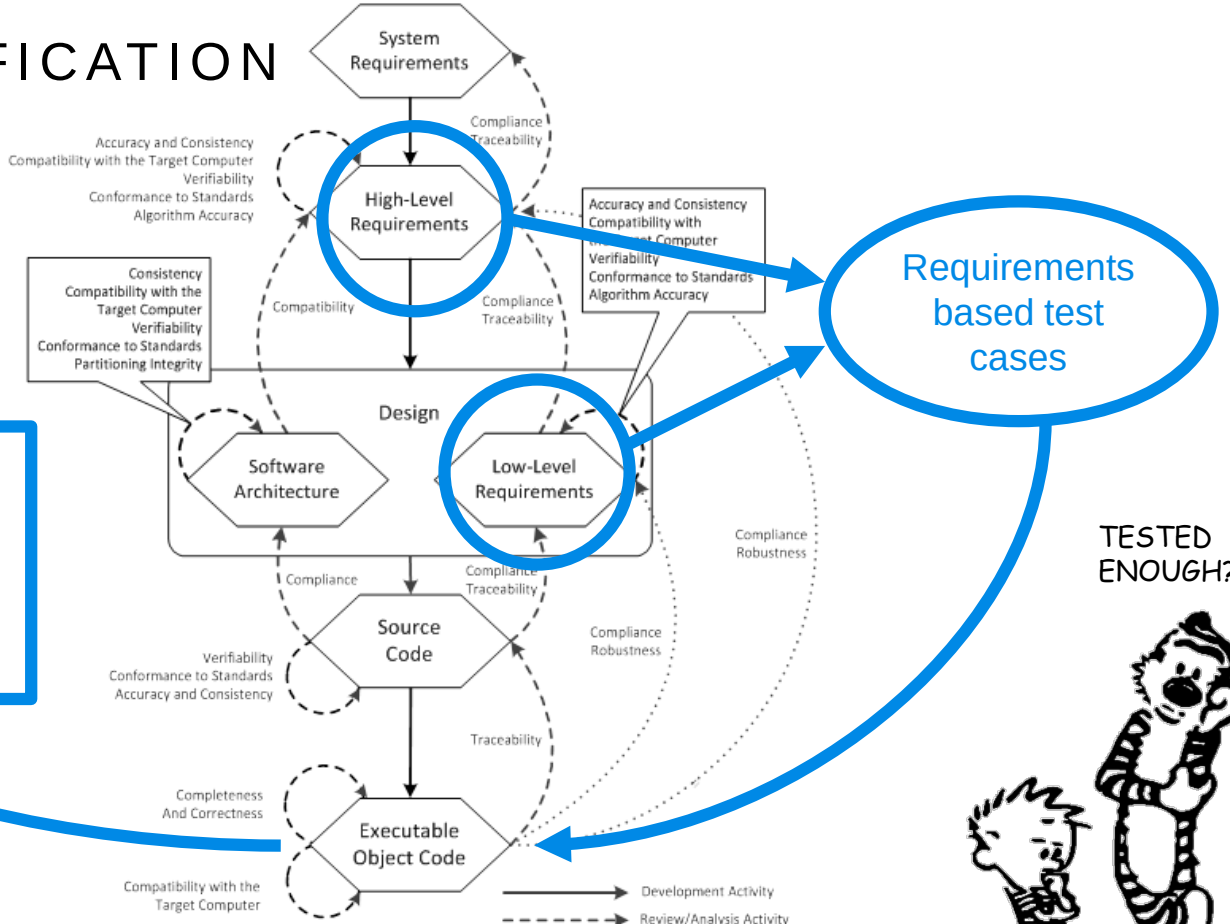


Diagram adapted from DO-333 Formal Methods Supplement to DO-178C and DO-278A

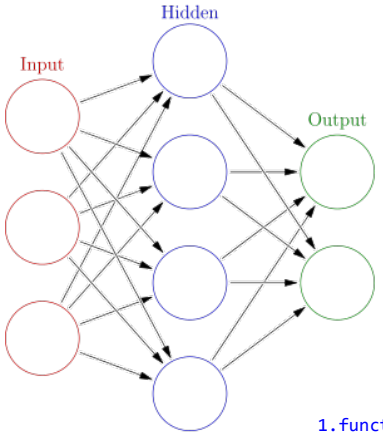
Requirements based test cases

Structural Coverage Metrics

- Unintended functionality
- Missing requirements
- Inadequacy of test cases
- Dead or deactivated code

Collins Aerospace

STRUCTURAL COVERAGE FOR NN?



- No branches or relational operators
- Complete coverage with single test case!
- How to achieve underlying coverage objectives?

```
1.function [y1] = simulateStandaloneNet(x1)
2.  % Input 1
3.  x1_step1_xoffset = 0;
4.  x1_step1_gain = 0.200475452649894;
5.  x1_step1_ymin = -1;
6. % Layer 1
7.  b1
= [6.0358701949520981;2.725693924978148;0.58426771719145909;-
5.1615078566382975];
8.  IW1_1 = [-14.001919491063946;4.90641117353245;-
15.228280764533135;-5.264207948688032];
9.   % Layer 2
10.  b2 = -0.75620725148640833;
11.  LW2_1 = [0.5484626432316061 -0.43580234386123884 -
0.085111261420612969 -1.1367922825337915];
```

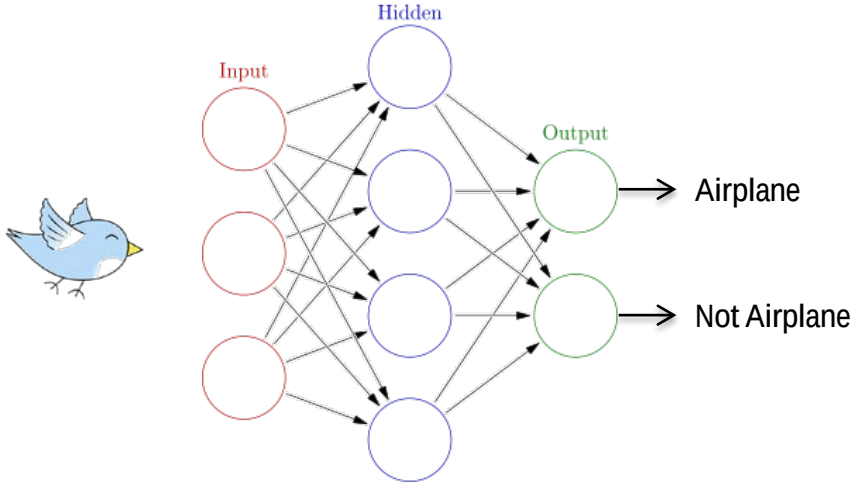
```
1.% Output 1
2.  y1_step1_ymin = -1;
3.  y1_step1_gain = 0.2;
4.  y1_step1_xoffset = 0;
5.  % ===== SIMULATION ========
6.  % Dimensions
7.  Q = size(x1,2); % samples
8.  % Input 1
9.  xp1 = mapminmax_apply(x1,x1_step1_gain,
x1_step1_xoffset,x1_step1_ymin);
10.  % Layer 1
11.  a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*xp1);
12.  % Layer 2
13.  a2 = repmat(b2,1,Q) + LW2_1*a1;
14.  % Output 1
15.  y1 = mapminmax_reverse(a2,y1_step1_gain,
y1_step1_xoffset,y1_step1_ymin);
16.end
```

```
1.% ===== MODULE FUNCTIONS ========
2.% Map Minimum and Maximum Input Processing Function
3.function y = mapminmax_apply(x, settings_gain, settings_xoffset,
settings_ymin)
4.  y = bsxfun(@minus,x,settings_xoffset);
5.  y = bsxfun(@times,y,settings_gain);
6.  y = bsxfun(@plus,y,settings_ymin);
7.End

8.% Sigmoid Symmetric Transfer Function
9.function a = tansig_apply(n)
10.  a = 2 ./ (1 + exp(-2*n)) - 1;
11.End
12.
13.% Map Minimum and Maximum Output Reverse-Processing Function
14.function x = mapminmax_reverse(y, settings_gain, settings_xoffset,
settings_ymin)
15.  x = bsxfun(@minus,y,settings_ymin);
16.  x = bsxfun(@rdivide,x,settings_gain);
17.  x = bsxfun(@plus,x,settings_xoffset);
18.end
```

BACK TO ASSURANCE CHALLENGES

- Verification
  - Testing does not provide adequate assurance.
  - New testing generation methods, new metrics? What analysis techniques can be applied?
- Requirements
  - Can we create precise, actionable requirements for learning components?
- Implementation
  - “Big pile of linear algebra,” functional languages, etc.
  - *How do we measure completeness of requirements or of verification results?*



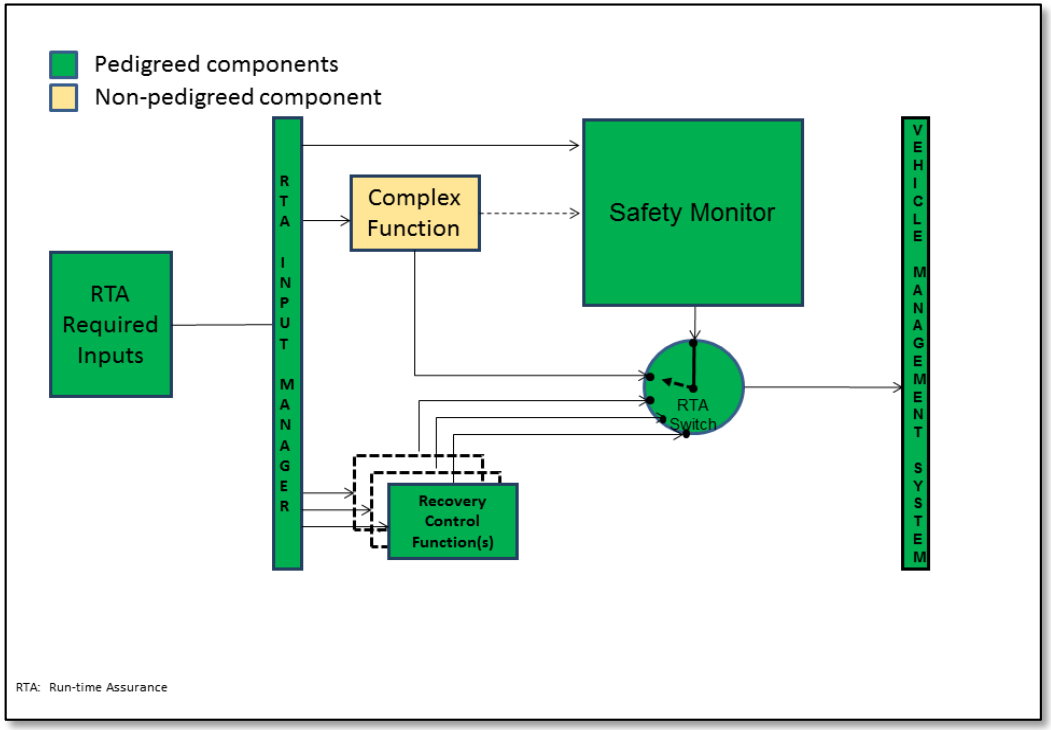
3. WHAT ADVANCES ARE NEEDED?

- NASA: Certification Considerations for Adaptive Systems
  - NASA/CR–2015-218702
- DARPA: Assured Autonomy program
- Architecture and Analysis for High-assurance Autonomy (AAHAA)
  - Collins, Stanford, Minnesota, Kestrel
  - Verified architectural mitigations (bounded autonomy/simplex with run-time monitors)
  - New testing methods
  - Advanced analysis methods (formal methods for NN)
- FAA tracking this closely

ASTM F3269-17

- Standard Practice For Methods To Safely Bound Flight Behavior Of Unmanned Aircraft Systems Containing Complex Functions
- “Complex Function” = LEC

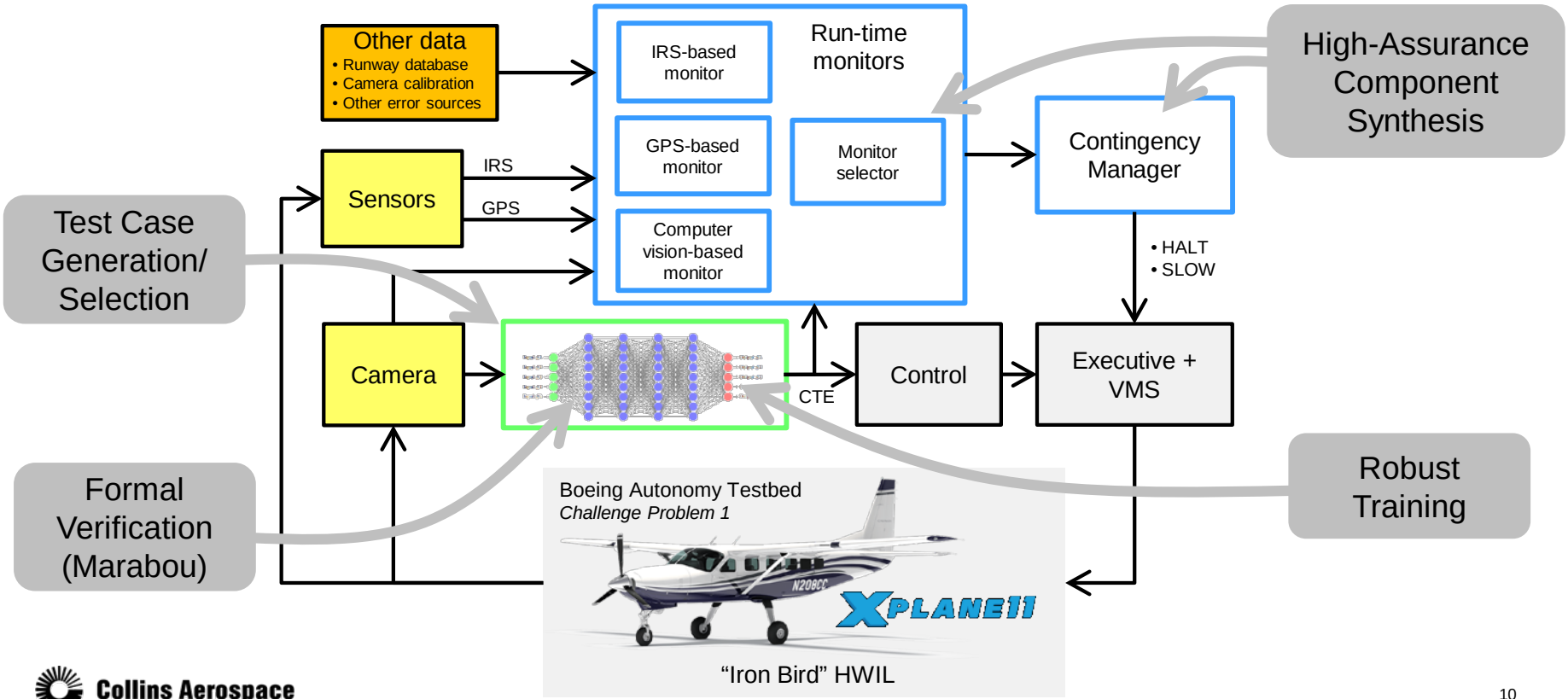
Clark, Koutsoukos, Porter, Kumar, Pappas, Sokolsky, Lee, Pike, “A Study on Run Time Assurance for Complex Cyber Physical Systems,” AFRL Report, 2013



Goal is to develop the standard to a level of capability that defines run-time monitoring (RTA) attributes to a level that the FAA or CAA will agree that monitors developed to this standard are sufficient to allow the UAS to evolve the complex function with its associated avionics equipment and sensors without requiring vehicle recertification as the CONOPS evolve after initial certification

AAHAA PHASE 1 DEMO

Architecture for Run-Time Monitoring



Collins Aerospace