

# **Test and Evaluation of Autonomous Systems NDIA T&E Conference**

San Diego, Calif.

7 Mar 2017

Copyright © 2017 Raytheon Company. All rights reserved.

*Customer Success Is Our Mission* is a registered trademark of Raytheon Company.

The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

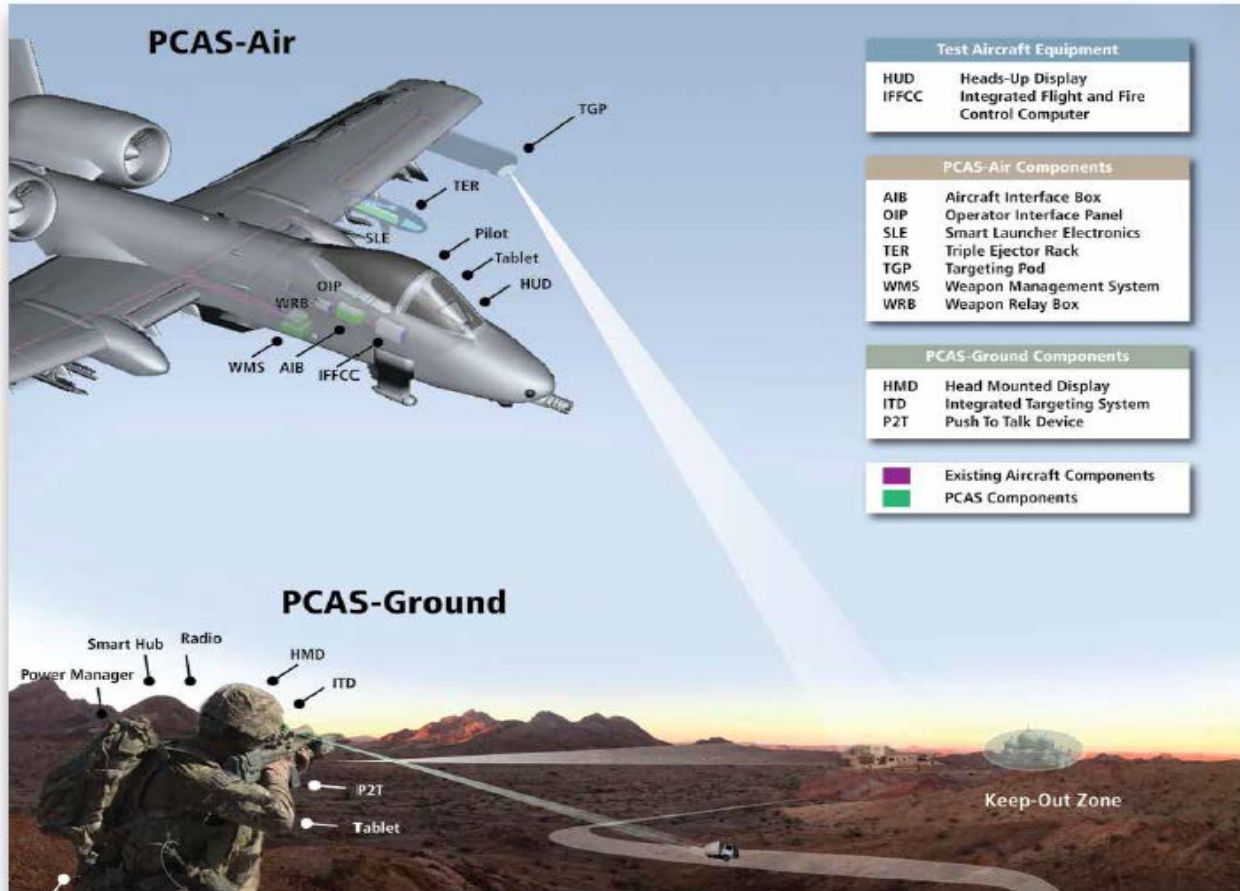
# The Autonomous Test and Evaluation Challenge

---

- As we add more and more autonomy into our systems, how do we test that autonomy?
- Some algorithms may choose a different answer – both of which may be correct – depending on the random start seed
- How do we do enough testing to build trust in a system?
- How do we represent the autonomy into the user interface and then test it accordingly?
- This presentation will look at one example – the DARPA Persistent Close Air Support (PCAS) program

# PCAS Baseline System Architecture

## PCAS Air and Ground Segments



**Baseline system enables all the PCAS Measures of Performance to be Demonstrated**

# Video

---

## DARPA Public Release Summary Video

Video Courtesy of DARPA:

<http://www.darpa.mil/news-events/2015-09-17>

# PCAS Autonomy Challenge

- Advances targeting for rapid and more accurate engagements.
- Autonomous planning tools conduct A/C auto-routing, sensor tasking and weapon launch/trajectory shaping for desired terminal effects.
- Common visualization of planned A/C routing, sensor tasking and target engagement for better JTAC/pilot oversight and coordination.
- Open Standards CoT messaging provides for multiple users.



Photo courtesy of DARPA.mil



# User Interfaces to Autonomy for PCAS



*Weapon Effects*



*Laser Safety Fan*



*Collateral Damage Estimates*



*High-Resolution Video*

# Lessons Learned in Autonomy Testing

---

- For user interfaces, get frequent user feedback so that they understand and trust the autonomy
- Test as much in simulation as you can before going to flight test
- Concentrate on the most likely use cases, but pay attention to the edge cases which can “break” the system
- Develop scripts to exercise the autonomy ... Have the scripts emulate user inputs
  - Very time consuming to have users conduct each test by hand
  - Bring in users to test the system after the scripts have worked out most of the bugs

**Autonomous System Test Is Possible – But Requires Different Thinking on Testing**