Independent Testing

Justin R Morris
Justin.R.Morris@nasa.gov
NASA Computer Engineer
JSTAR Team Lead

NASA's IV&V PROGRAM Dynamic Analysis

https://www.nasa.gov/centers/ivv/home/index.html

March 10, 2017

Outline

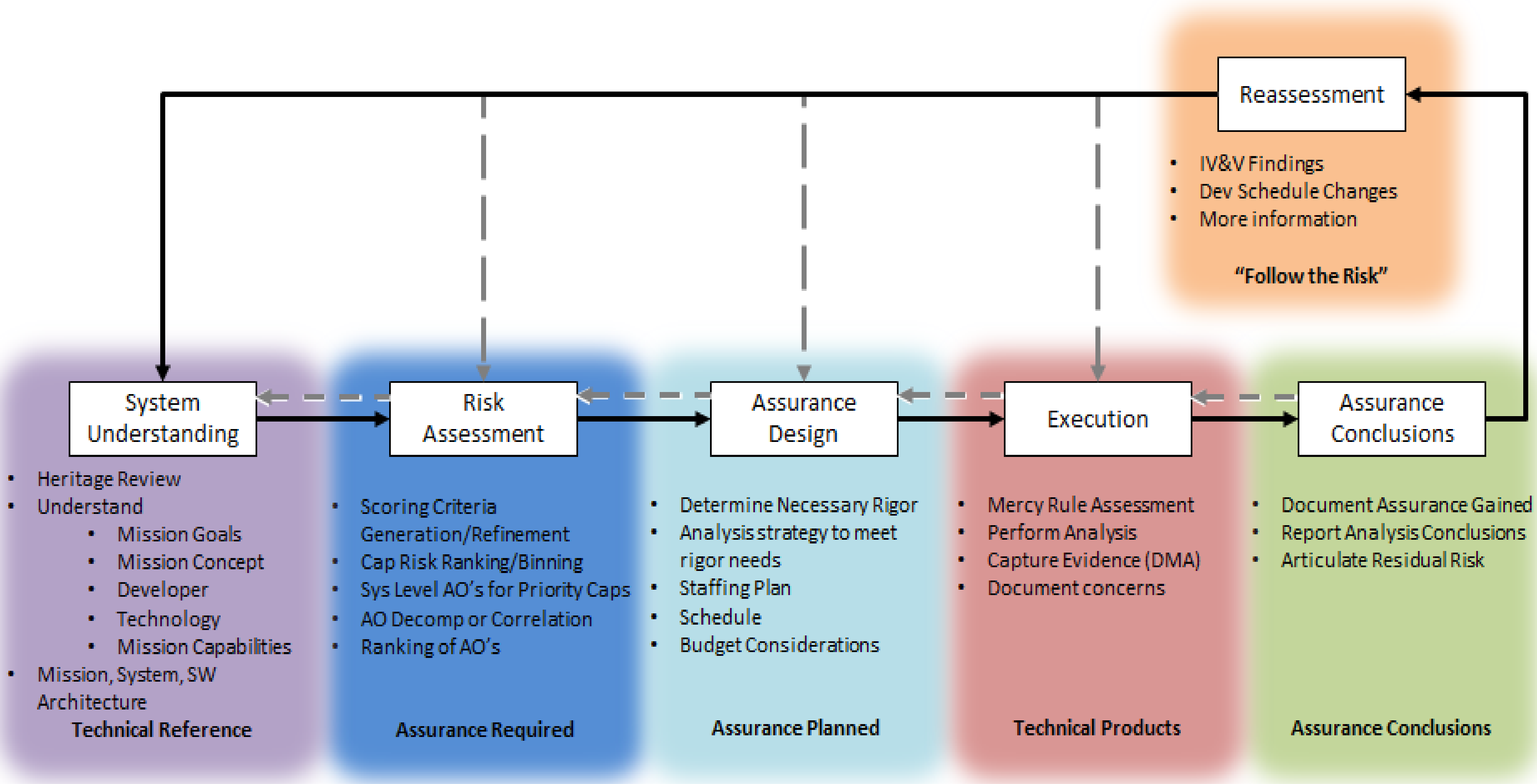
- Introduction
  - NASA IV&V Program
  - NASA IV&V Process
  - Terms and Definitions
  - NASA IV&V Program Independent Testing Objectives
  - Why Independent Test?

- NASA IV&V Independent Testing
  - Plan the Testing
  - Acquire the Test Environment
  - Prepare the Test Environment
  - Execute Tests and Analyze Results

NASA IV&V Program Overview

- Responsible for performing Independent Verification and Validation (IV&V) on high profile NASA missions
- Functionally reports to the Chief, Office of Safety and Mission Assurance (OSMA) – NASA HQ
- Agency requirements for IV&V defined in
  - NPD 7120.4D, NASA Engineering and Program/Project Management Policy
  - NPR 7150.2, NASA Software Engineering Requirements
- Staff of ~250 personnel
- ISO 9001:2008 Certified
- Only NASA Center with VPP Star Certification
- Pursuing FedRamp Certification
- Legacy of Customer Satisfaction
  - 94% acceptance rate for findings
  - 95% customer satisfaction (2015 annual customer survey)

NASA IV&V Process



Terms and Definitions

- Independent Testing – Application of dynamic analysis that is conducted by an organization apart from the system developer to verify the implementation (source/object code) is correct and complete

Independent Software Testing Objectives

- Independent Software Testing is a **method** used by our Program to reduce system and software risks
- Provides evidence-based assurance of correct and complete implementation of software behaviors in final source or binary images within an operational environment
- Provides substantiating evidence of issues discovered during other activities in the IV&V process

Pros of Independent Testing

- Testing provides strong objective evidence of defects
- Supplements Requirements Analysis, Design Analysis, and Implementation Analysis
- Supplements Test Documentation Analysis for integration coverage
- Provides high degree of assurance that assertions regarding software quality and suitability are accurate
- Practical method for verifying certain behaviors, like:
  - Deadlock and race avoidance
  - Inter-process timing and queuing (like SpaceWire backpressure)
  - Complex fault management with multiple faults
  - Extended duration operations
  - Hardware failure fault management
  - Interdependencies

Independent Test Lifecycle

1. **Plan the Testing**
   - Establishing the focus areas for testing; what assurance objectives are best accomplished via testing; resources needed to support the analysis and testing
   - Determining the requirements of the test platform to support the planned assurance objectives
2. **Acquire Test Environment**
   - Acquire the simulations, test hardware and other assets
   - Integrate all into functional test bed for injecting IV&V tests scripts
3. **Prepare for Testing**
   - Configuration of the test environment
   - Develop test plans and associated test scripts
   - Configuration management of test environment and artifacts
4. **Perform Testing**
   - Execute test, collect results and evaluate results

1. Plan the Testing

IV&V Test Planning

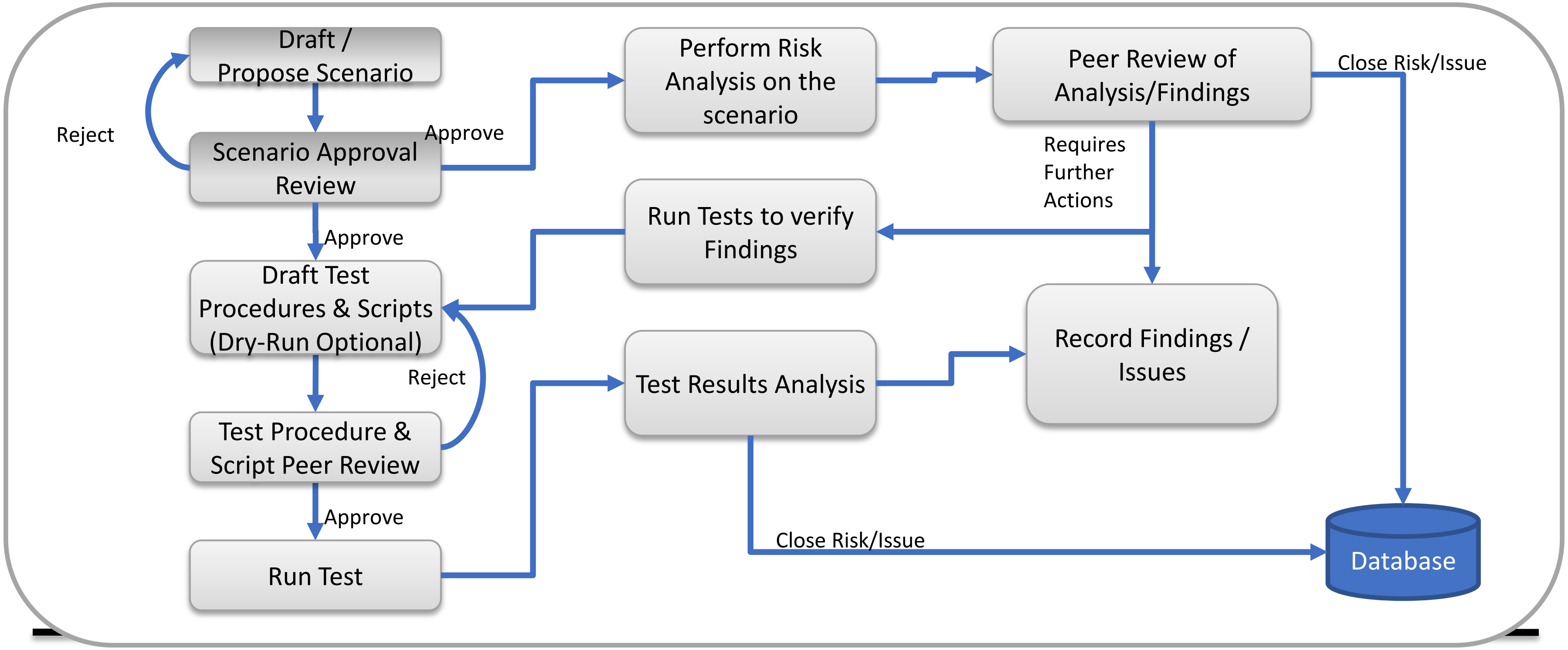
- Consider risk of system/subsystem
  - IV&V is Risk-Driven and Focused
  - IV&V analyses should "Follow the Risk"
- Select appropriate IV&V artifacts and development artifacts as test source data
- Identify a set of developer tests to validate test environment
- Plan and document test cases

Test Case Identification

- Project 1
  - IV&V Analysis team comes up with candidate test cases in lightweight form
  - Candidate test cases are screened through Independent Test Working Group and approved by lead
  - Selected Test Cases are fully developed and tested
- Project 2
  - Team identifies risk reduction scenarios (RRS) which are not being tested by the Developer
  - The RRSs are specified in the test plan
- Project 3
  - Analysts define test cases based on lifecycle analysis being performed

Risk Driven Testing
(Approach for Risk Reduction Scenarios Identification)

- Definition of Risk Reduction Scenarios:
  - Risk scenarios must have impact to the mission success or cause the loss of mission objectives. For example, a fault that happened at the wrong time may cause the loss of mission.



IV&V – "Follow the Risk" (Mission Example)

- Some Examples
  - Stored Command Sequence (SCS) Validation
  - Long-Duration FSW Testing
  - Fault Scenarios (e.g., Stuck Thruster)
  - Flight Computer fault injection (Instruction faults, etc.)
  - Primary/Backup C&DH Swapping
  - Dropped Packet Analysis
  - Visualization Capability for Attitude Control System

"Follow the Risk"

Risk

- Prior IV&V/Project Risk captures complexity and criticality of Stored Command Sequences when also considered a part of the larger integrated system of systems.

Mitigation/Action

- Mitigation included more rigorous design and peer review process, and "potential" for additional testing
- IV&V perform additional SCS testing
- Tie into developer Working Group

Results/Findings

- Test Scenarios Tested
  - 15 Issues Identified to Date

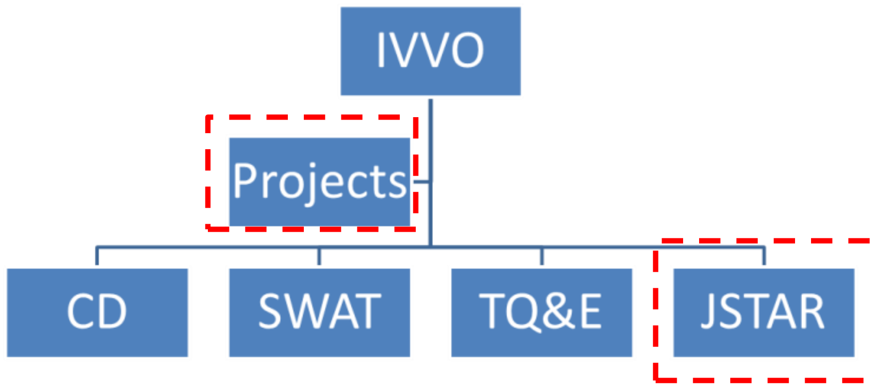
Value Added

- Impacts to Project
  - Impact to schedule and cost averted.
- Impacts to IV&V
  - Insight into FM/Observatory response increased.
  - Awareness of SCS criticality revisited

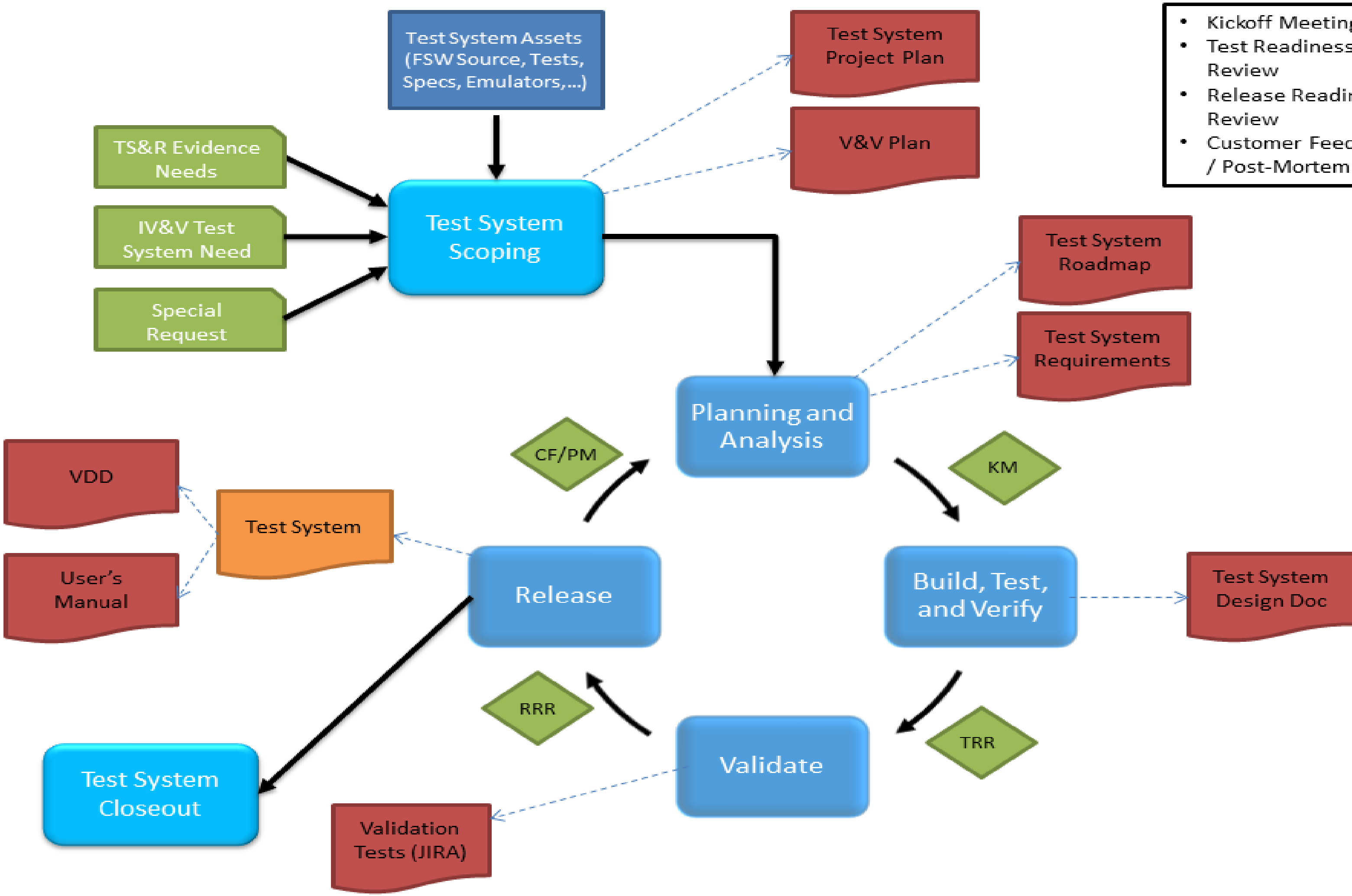
2. Acquire Test Environment

Jon McBride Software Testing and Research (JSTAR)

Develop, maintain, and operate test environments and supporting tools for the IV&V Program that enables the dynamic analysis of software behaviors for multiple NASA missions



Test Environment Development Process



- Kickoff Meeting
- Test Readiness Review
- Release Readiness Review
- Customer Feedback / Post-Mortem

Test Environment Types

I) Software only Simulation – Full-Up	II) HW/SW Hybrid Simulation – Full-Up	III) Software Only – Alt Compile Execution	IV) Subsystem – Alt Compilation
• Requires moderate development effort (cost/schedule) • Provides high degree of realism (hard to refute findings) • Dependent on build releases of software under test • Most flexible test options • TEST as Fly	• Hardest to build • Highest cost • Highest degree of realism • Limited fault injection options • Depends on HW and SW availability • TEST as Fly	• Not TEST as Fly • Requires modification of the FSW to stub out board support dependencies • May require less effort to build • May be available from development program • Not ground system dependent	• Least test environment development (stubs to cut subsystem out of system arch. may require extensive development) • Not TEST as Fly • Provides quick assessment of unit level concerns. • Algorithm verification

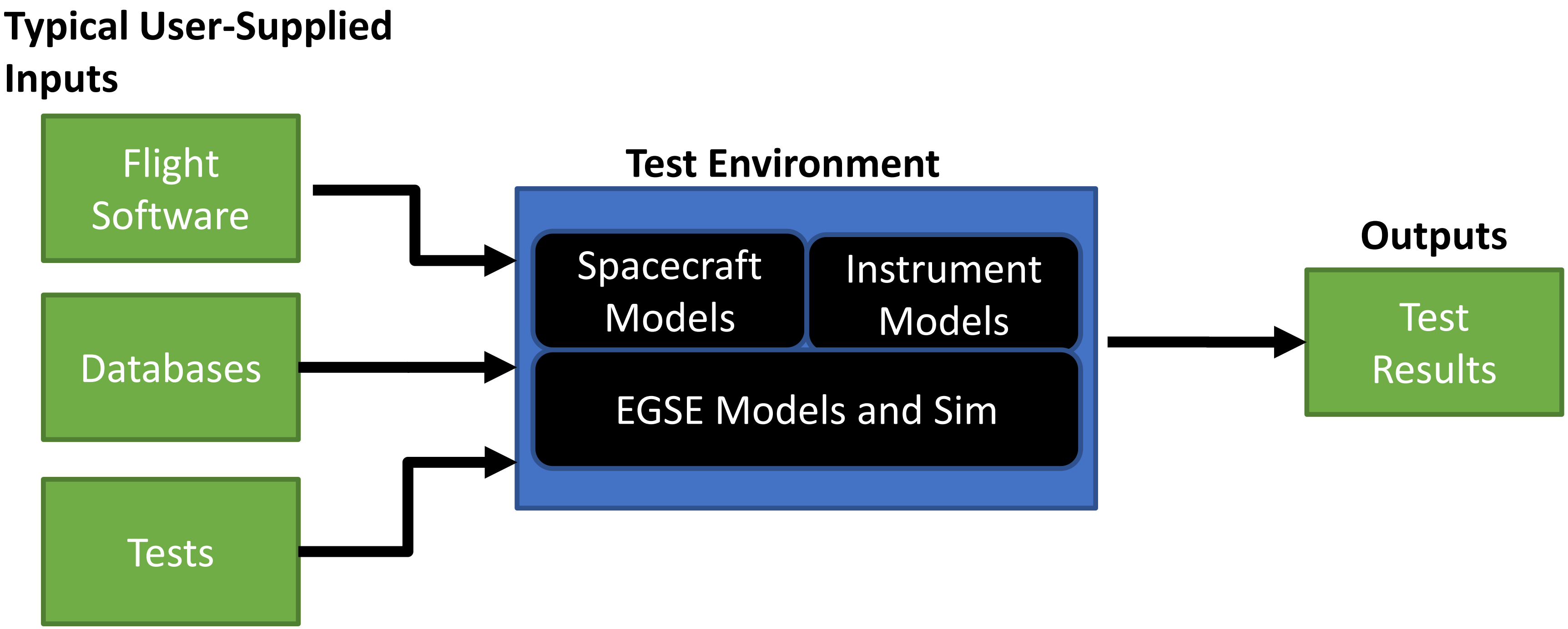
Test Environment Acquisition Methods

	Method
1	Acquire Development Project Test Environment
2	Develop Test Environment (Hybrid or Completely Independent)
3	HWIL Environment
4	Test Environment Remote Access

With respect to Test System / Environment Acquisition, Every Mission is Unique.

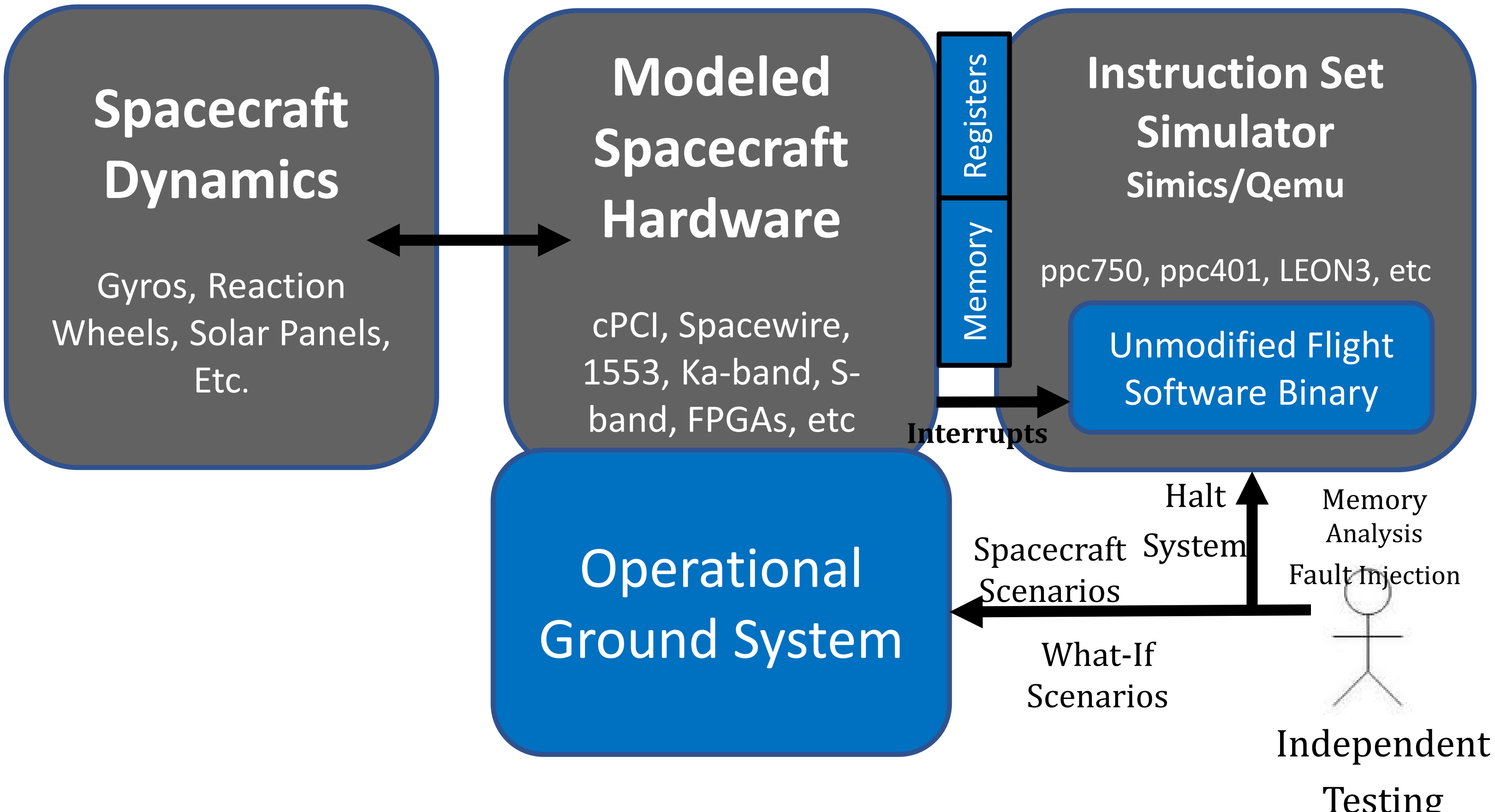
Test Environment Development Methodology

- Users must specify FSW and Database versions
- Users must specify spacecraft hardware configuration



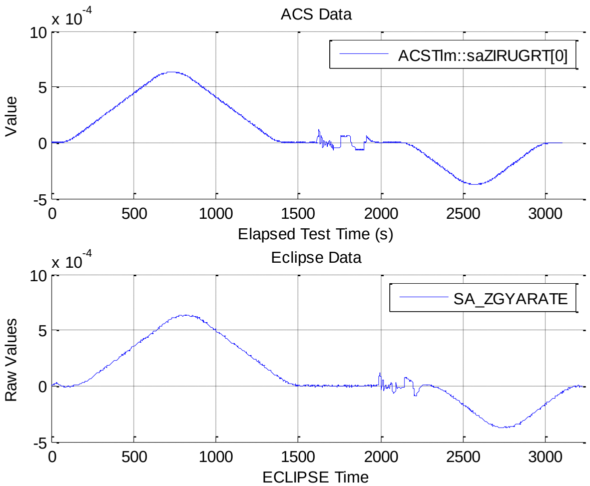
Representative Test Environment

Simulator Components

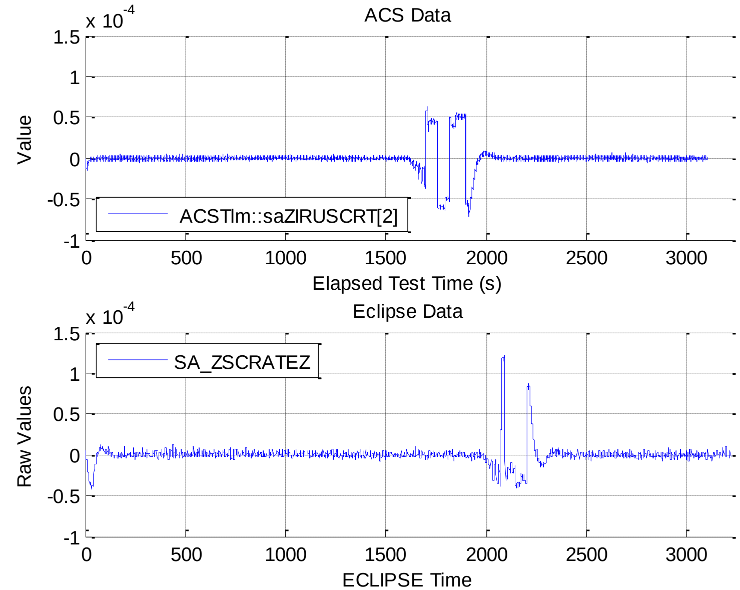


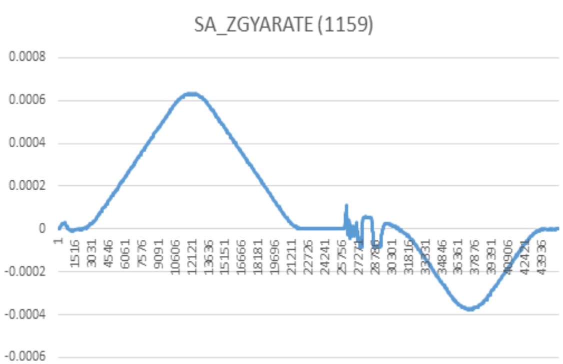
3. Prepare Test Environment

Independent Test Environment Validation

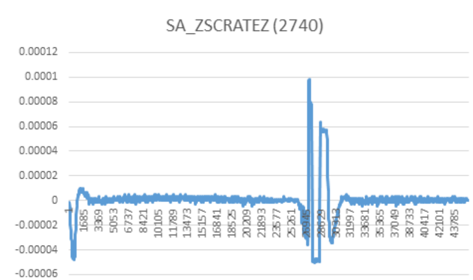


Developer Test Results





IV&V Testbed Results



4. Perform Testing and Analyze Results

Test Execution and Analysis

- Execute, Collect Results, and Analyze Results
- Adjust Test Planning as necessary for next iteration

Planning & Test Execution are iterative processes

Build and release test environment in sequential iterations

Independent Testing Results

- Increased system and software understanding
- Increased IV&V Program Capability
- Measurable IV&V assurance
- Run-time experience with most critical flight software behaviors
- Validation of fault handling software
- Verification of software behavior and identification of software issues
  - Often process forces flight software down non-happy paths
    - High Severity Software Issue Found During Model Development
    - Execution of "non-flight" code in the flight binary
    - Flight computer model forced flight software down non-happy paths of the board support package (BSP)

Thank you!

- Introduction
  - NASA IV&V Program
  - NASA IV&V Process
  - Terms and Definitions
  - NASA IV&V Program Independent Testing Objectives
  - Why Independent Test?

- NASA IV&V Independent Testing
  - Plan the Testing
  - Acquire the Test Environment
  - Prepare the Test Environment
  - Execute Tests and Analyze Results