Toward Practical Formal Analysis of Flight Control Systems in a Model-Based Development Environment

Highlights of the CerTA FCS Program

Advanced Development Programs

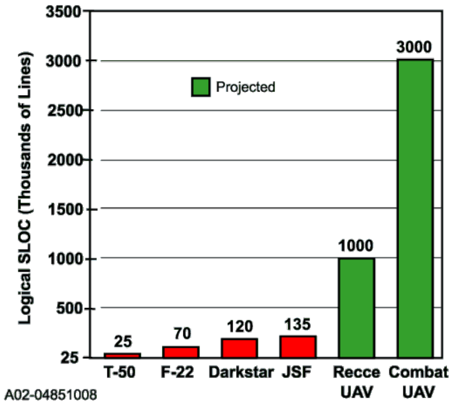
LOCKHEED MARTIN

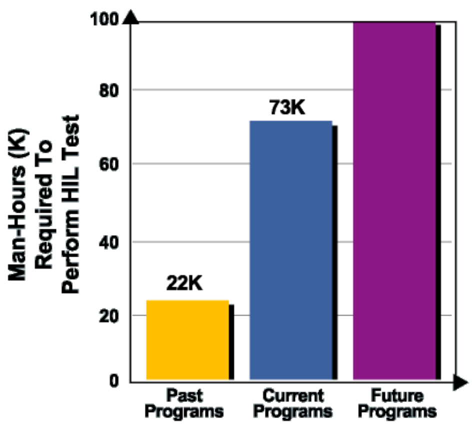
Walter Storm

20.May.2009

Background (VVIACS)

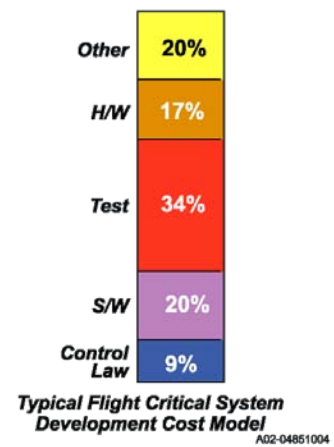
- System Complexity is Exponentially Increasing





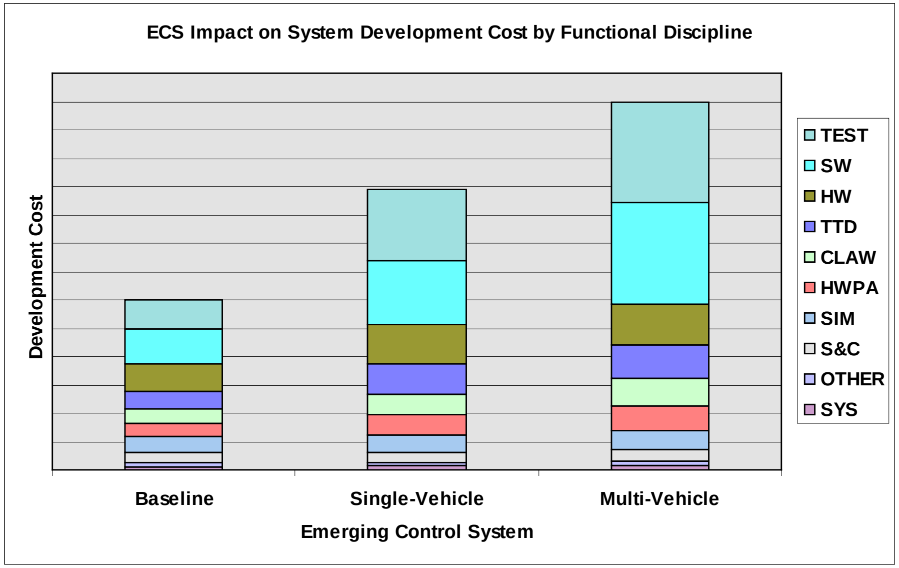
- Future Military Program Testing Hours Are Forecast to Triple

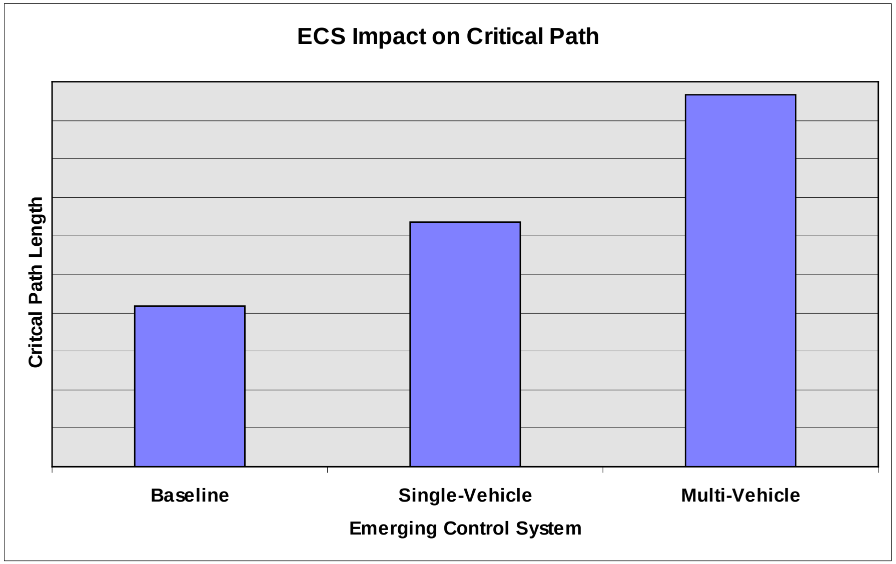
- Testing Consumes Over 1/3 of the System Development Cost



Typical Flight Critical System Development Cost Model

Background – V&V Cost and Schedule Have the Most Impact on Development





- Single-Vehicle ECS Increases Development Costs ~ 50%, V&V Costs ~ 100%, and Critical Path Length ~ 50%
- Multiple-Vehicle ECS Increases Development Costs ~ 100%, V&V Costs ~ 150%, and Critical Path Length ~ 125%
- Software: Single-Vehicle 100% Increase and Multiple-Vehicle 200% Increase in V&V Costs
- Test: Single-Vehicle 150% Increase and Multiple-Vehicle 250% Increase in V&V Costs

Total Cost of System Testing

- **The total cost of Integrated System Testing for Program X includes many resources often taken for granted.**

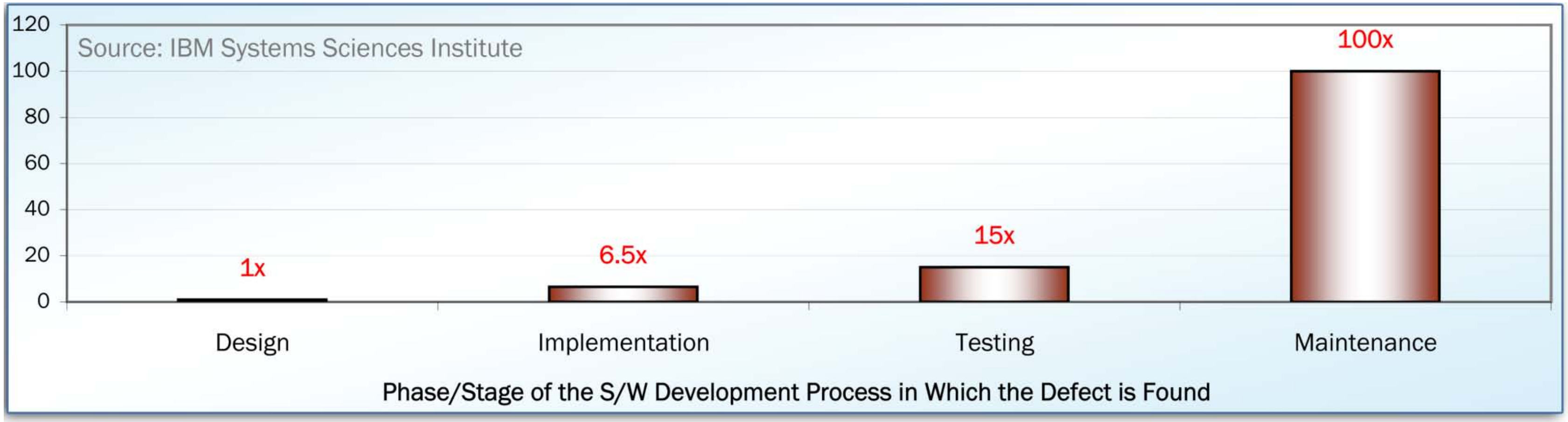
Resource	Assumption	Hourly Rate
Simulation Hardware	$1.2M/7yr, 2000hr/yr	$ 85.70
Flight Hardware	$800K/7yr, 2000hr/yr	$ 57.14
Test Station	$1.2M/7yr, 2000hr/yr	$ 85.71
Simulation Support	$150/hr * 2	$ 300.00
Test Engineer	$200/hr	$ 200.00
Facility	$15M/30yr, 2000hr/yr	$ 250.00
Power Requirements	2000kW @ 0.14/kw-H	$ 280.00
Total		**$1,258.55**

- **In all, a full-up test program can cost over $20 per minute.**

Advanced Development Programs

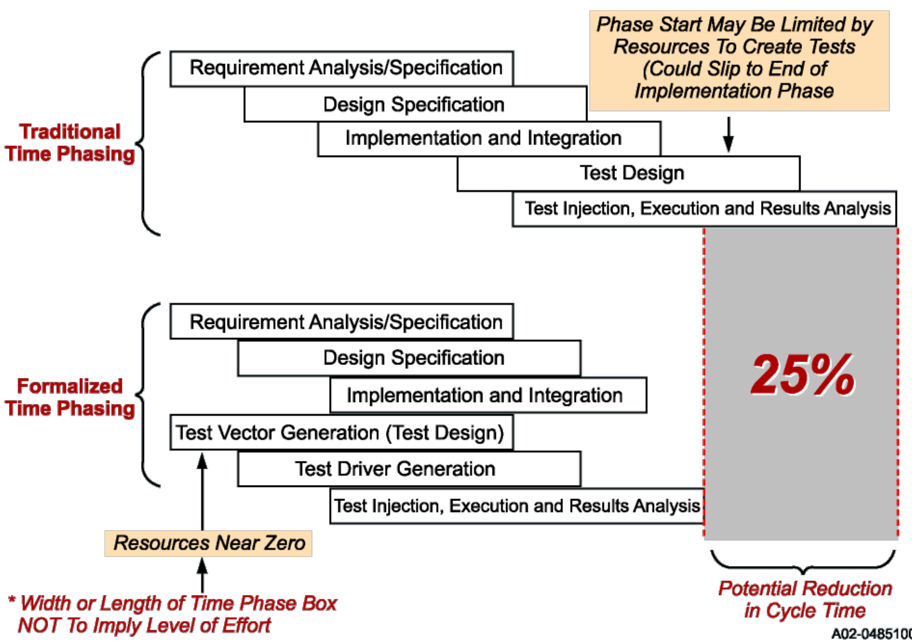
Defect Cost vs. Development Phase

- **It is imperative that software defects are identified early in the development cycle.**
  - **Defects found during test cost 15x more to fix than those found during design.**



The Goal for CerTA FCS

- **Detect errors during system design.**
- **Maximize system test resource utilization.**
- **Demonstrate a reduction in system development cycle time as proposed by VVIACS.**



Advanced Development Programs

CerTA FCS Technology Focus

- We chose to focus on VVIACS technologies that:
  - Best align with the CerTA FCS goal
  - Offer the best balance between:
    - Overall Cost/Benefit Ratio
    - Near-/Mid-/Far- Term Application

VVIACS Technologies Ranked according to Overall Cost/Benefit Ratio →

1. Automated Verification Management
2. Formal Requirements Specifications
3. Requirements and Traceability Analysis
4. Formal Methods
5. Probabilistic / Statistical Test
6. Requirements and Design Abstraction
7. V&V Run-Time Design
8. Testing Metrics
9. Rigorous Analysis for Test Reduction
10. Computer-Aided System Engineering

Primary CerTA FCS Focus — *Secondary CerTA FCS Focus* — *TASS SBIR*

CFR001-02

Advanced Development Programs

CerTA FCS Objectives

- **Develop and integrate a demonstration environment with an advanced flight control system for a UAV that provides a platform for assessing new certification techniques (Tasks 1&2).**
- **Show how Automated Verification Management can be applied to existing infrastructure to optimize certification tasks (Task 3).**
- **Apply Model Checking to a representative flight control system to prove critical properties of complex redundancy management (Task 4).**

CerTA FCS Objectives Cont'd

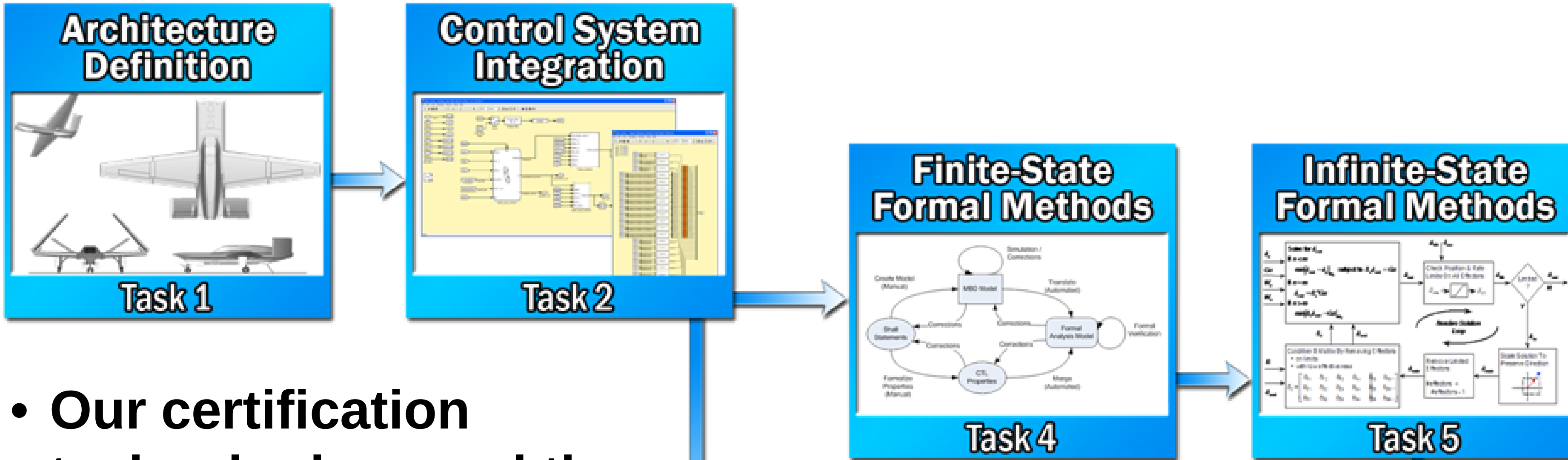
- **Extend Theorem Proving technology to address infinite-state systems within the safety-critical flight control domain (Task 5).**

- **Assess the improvements to the certification process made by these advancements (Task 6).**

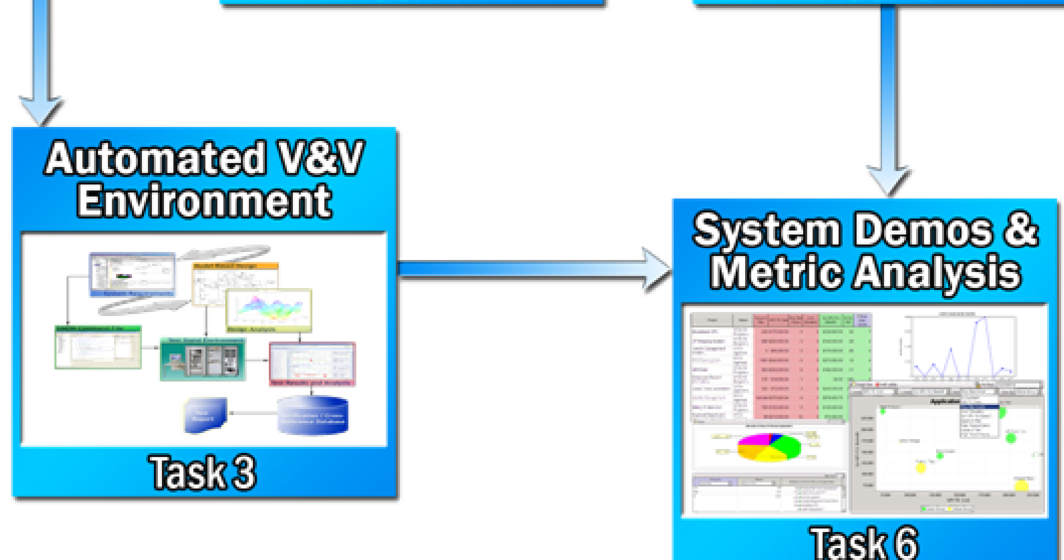
- **Provide a technology roadmap in terms of future advancements required for further demonstration and assessment (Tasks 3, 4, &5).**

Advanced Development Programs

Program Approach



- **Our certification technologies used the same baseline system for development.**



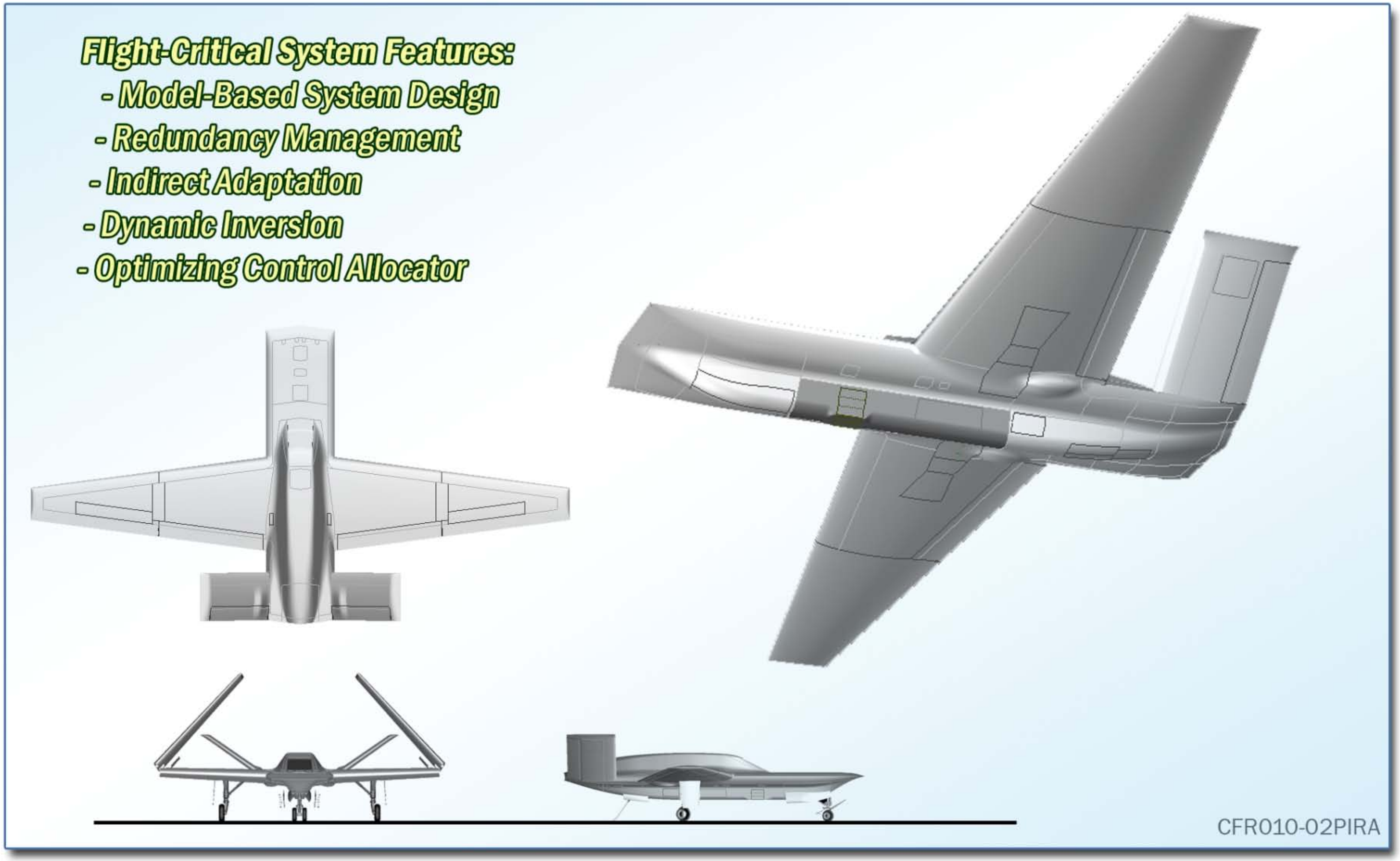
Advanced Development Programs

Research Findings – Model Checking

- **Finite-State Model Checking offers immediate benefit to the certification process**
  - **Finds subtle errors that are difficult to test (e.g., intermittent failures)**
  - **Finds errors during system design (1x cost to fix)**
  - **Reduces system testing, as tests shift focus from V&V of design to V&V of integrated system operation.**

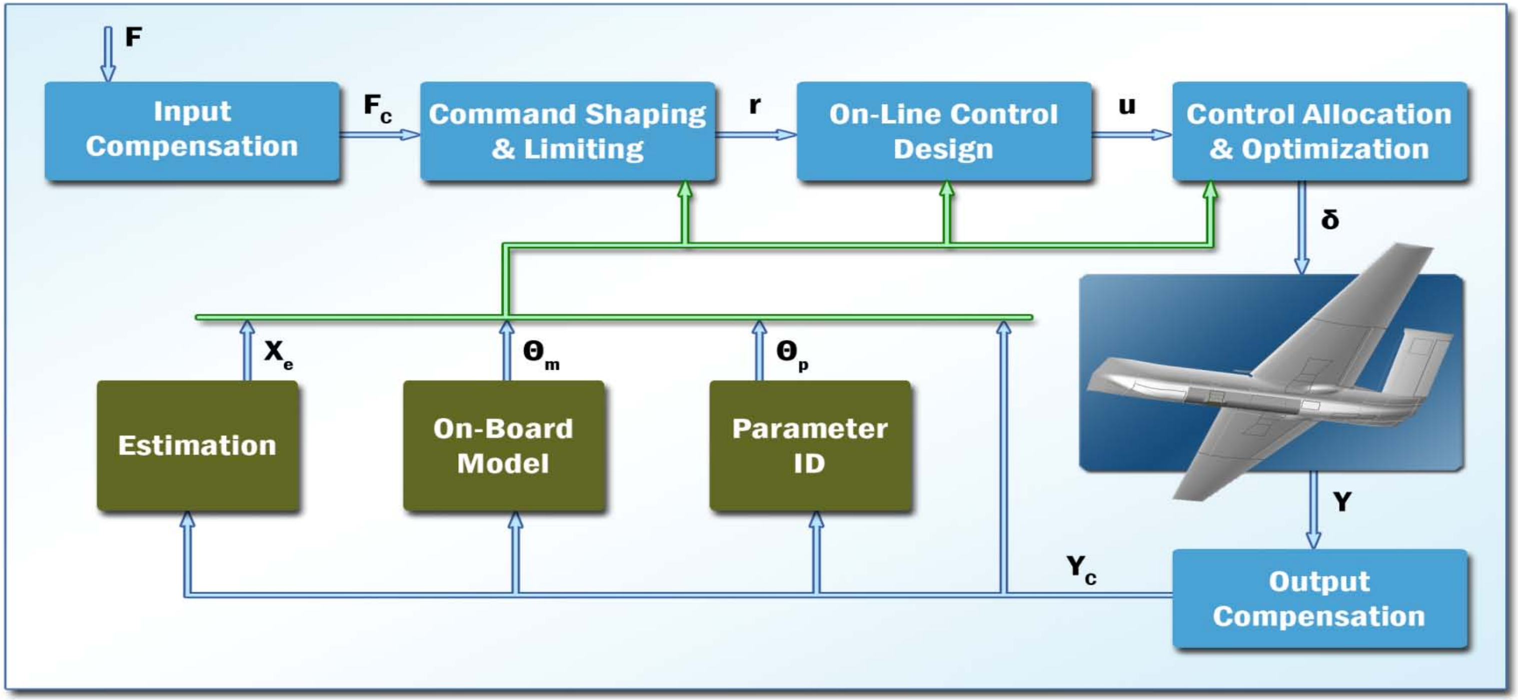
Baseline Representative UAV

- **Our CerTA FCS technologies were demonstrated on the Sea-Based Endurance UAV.**

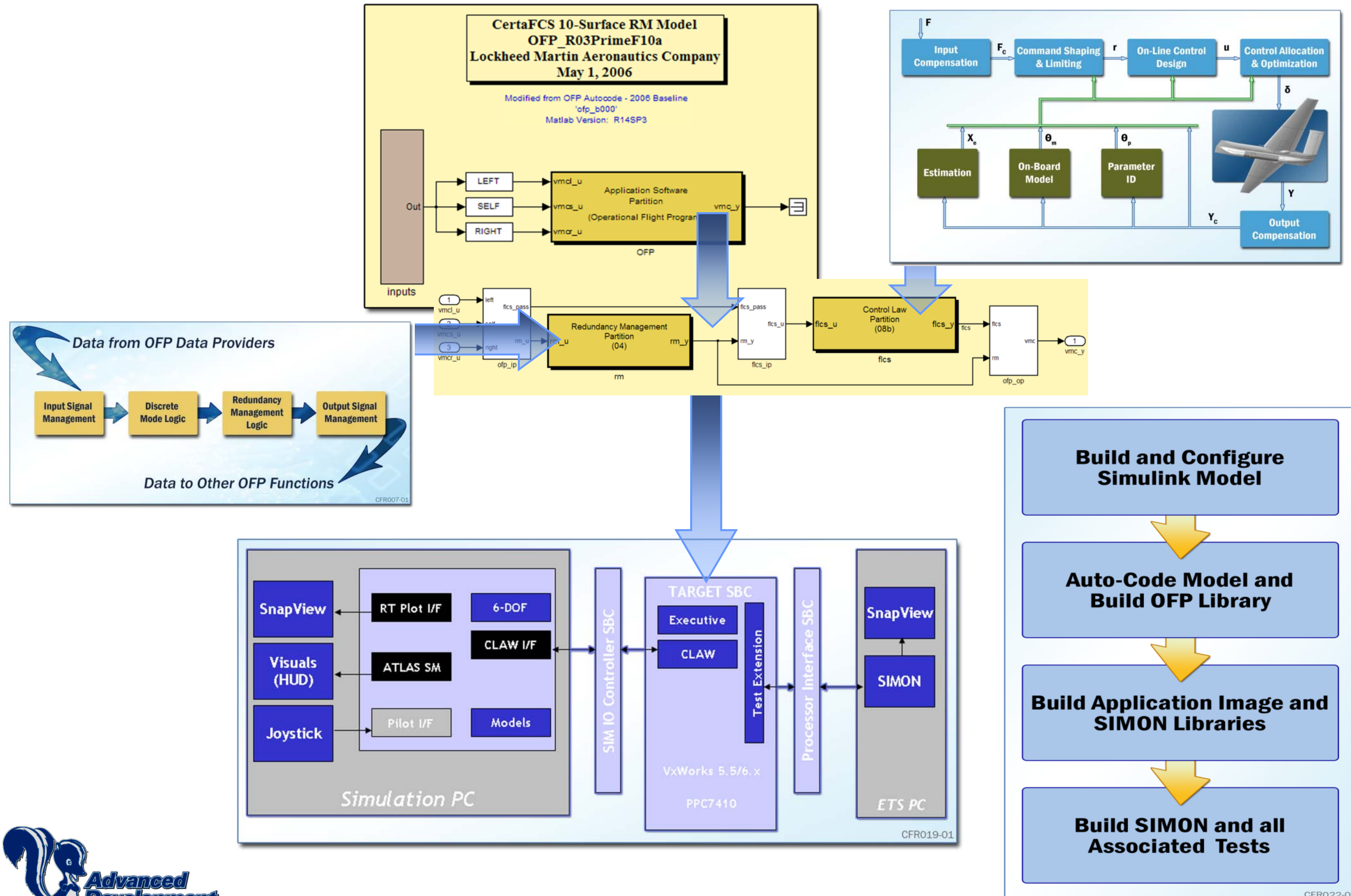


Inner-Loop Control System

- Dynamic Inversion CLAW
  - Supports ‘design-to-flying-qualities’ philosophy
  - Eliminates ‘tuning’ of large gain schedules
- Indirect Adaptation via Parameter ID
  - Additional robustness in presence of system failures or unforeseen conditions



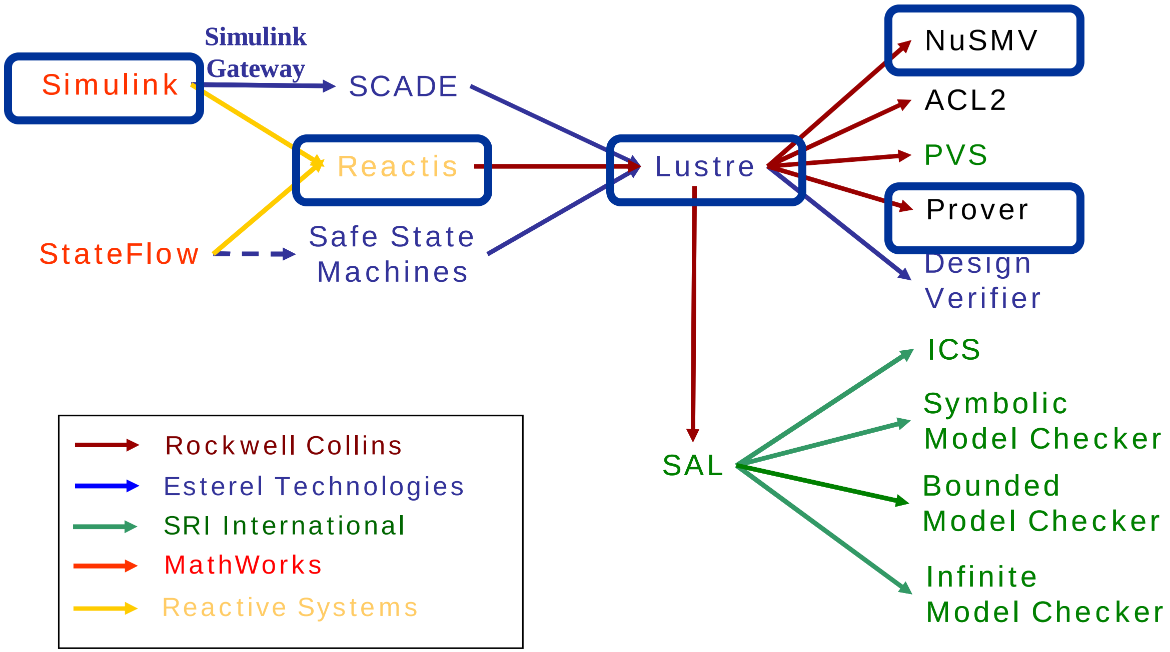
Baseline Triplex System Implementation



Advanced Development Programs

Analysis process
Automated infrastructure

- **Simulink + StateFlow models are automatically translated NuSMV verification model**
  - **includes CTL properties**
  - **type substitutions**

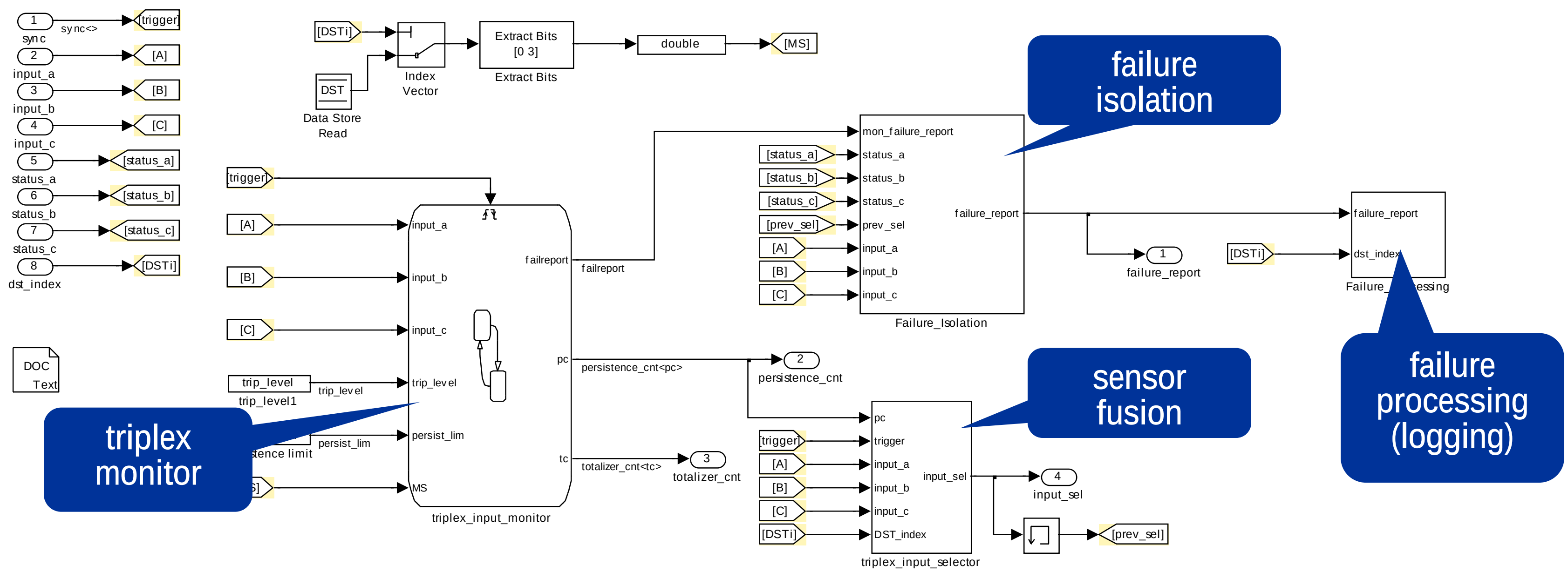


Advanced Development Programs

Finite State Analysis
Redundancy Manager demonstration

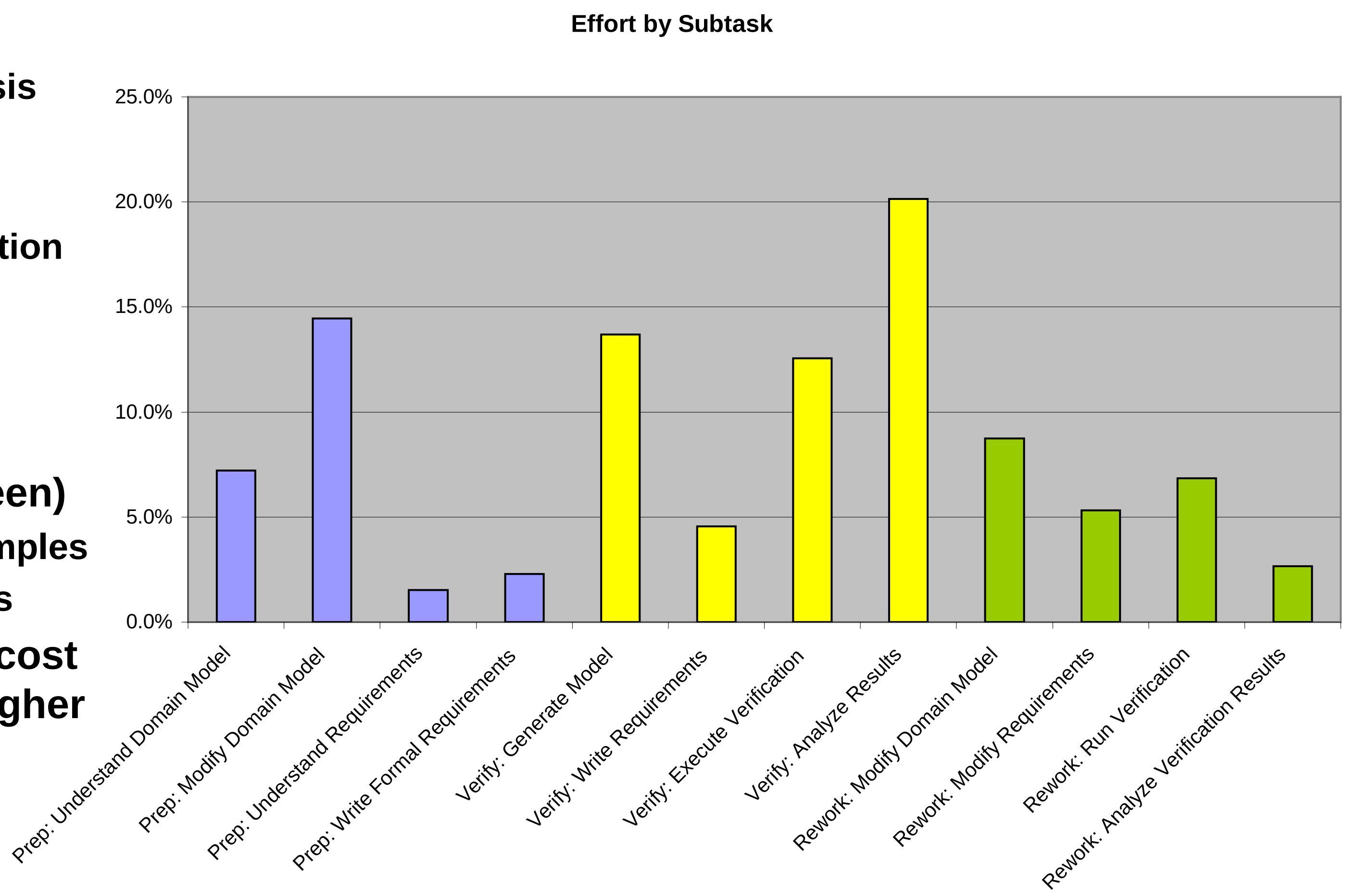
- **Sensor fusion, failure detection, and reset management for sets of triply redundant sensors**
- **FHT factored into separate *failure_ processing* model to reduce state**
- **Fixed-point operations for bit manipulation replaced by simpler blocks**

Subsystem	Subsystems / Blocks	Charts / Transitions / TT Cells	Reachable State Space	Properties	Confirmed Errors
Triplex voter no FHT	10 / 96	3 / 35 / 198	$6.0 * 10^{13}$	48	5
failure processing	7 / 42	0 / 0 / 0	$2.1 * 10^{4}$	6	3
reset manager	6 / 31	2 / 26 / 0	$1.32 * 10^{11}$	8	4
Totals	23 / 169	5 / 61 / 198	N/A	62	12



Redundancy Manager
Recurring analysis effort

- **25% cost for model preparation (blue)**
  - **Models were not designed for analysis**
- **50% cost for initial verification (yellow)**
  - **Property Formalization**
  - **Analysis**
  - **Counterexample Understanding and Explanation**
- **25% rework cost (green)**
  - **Fixing Counterexamples**
  - **Re-running analysis**
- **Usually, model prep cost is lower, rework is higher**



Testing vs. Model Checking

- **Successful demonstration**
  - **Collected metrics for verification of OFP redundancy management system**
  - **Extension of analysis framework**
  - **Design verification → shift from test-based to formal analysis**

Task 4 Study:
OFP Redundancy Manager

	Effort % of Total	Errors found
Testing	60%	0
FM	40%	12

Lockheed Aero – Testing

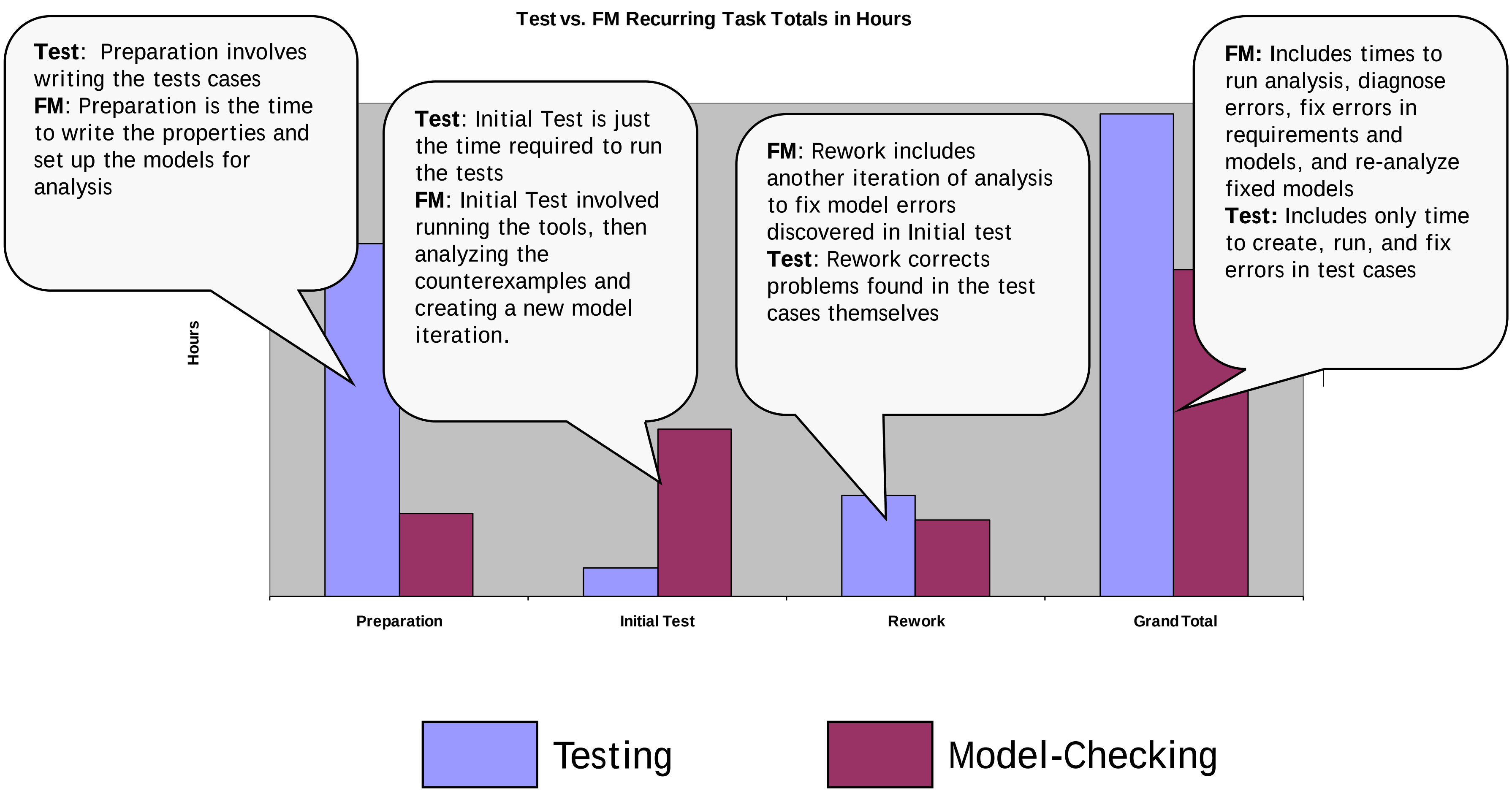
- Based on SIMON Test Rig
- Enhanced During CerTA FCS
  - Graphical Viewer of Test Cases
  - Support for XML/XSLT Test Cases
  - Added C++ Oracle Framework
- Developed Tests from Reqts
- Executed Tests Cases on SIMON

RCI – Model Checking

- Based on Gryphon Model-Checker
- Enhanced During CerTA FCS
  - Support for Simulink blocks
  - Support for Stateflow
  - Support for Prover model-checker
- Developed Properties from Reqts
- Proved Properties using Gryphon

Advanced Development Programs

Testing vs. Model Checking



Task 4 summary
Finite State Formal Methods

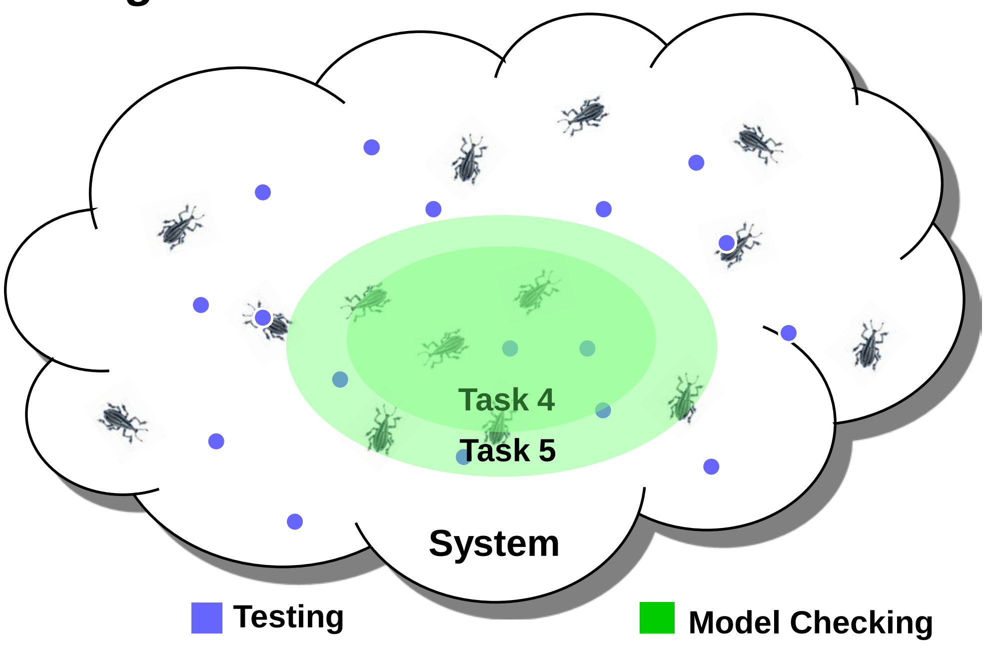
- **Successful application to formal verification to significant finite state software design**
- **Finite state model checking can provide:**
  - **Cost savings through automated analysis**
  - **Risk reduction through comprehensive & early error detection**
  - **Value-added process →**

    Smoother integration through explicit specification of component interfaces and environmental assumptions & constraints
- **Complementary to traditional V&V processes**
- **Task 4 Report**
  - **Guidance for insertion and use of automated translation and analysis environment into MBD process**

Advanced Development Programs

Formal Methods vs. Testing

- **Model checking and testing are complementary**
  - ***Errors are always made during development***
  - ***Testing can be used everywhere***
    …but does not provide complete coverage
  - ***Model-checking is very good at finding errors***
    …but doesn't work everywhere
  - ***Use model-checking where it works now***
    …technology is improving rapidly and will be even better in the future



Advanced Development Programs

Demonstration Results

- **Successful demonstration**
  - **Collected metrics for verification of OFP redundancy management system**
  - **Extension of analysis framework**
  - **Design verification → shift from test-based to formal analysis**

OFP Redundancy Manager

	Effort % of Total	Errors found
Testing	60%	0
FM	40%	12

Lockheed Martin – Testing

- Based on SIMON within AVVE
- Enhanced During CerTA FCS
  - Graphical Viewer of Test Cases
  - Support for XML/XSLT Test Cases
  - Added C++ Oracle Framework
- Developed Tests from Reqts
- Executed Tests Cases in AVVE

RCI – Model Checking

- Based on Gryphon Model-Checker
- Enhanced During CerTA FCS
  - Support for Simulink blocks
  - Support for Stateflow
  - Support for Prover model-checker
- Developed Properties from Reqts
- Proved Properties using Gryphon

Advanced Development Programs

Analysis Approach

- Determine Significance of Results
  - Why were no errors found by test?
  - What types of errors did Formal Methods find?
  - What is the impact on the bottom line (cost/schedule)?

- Tie-Back to VVIACS
  - Where are these technologies applicable?
  - What V&V parameters do these technologies affect?

Why No Errors were Found by Testing

- **Primarily, the demonstration was not a comprehensive test program.**
  - **Although some of these errors *could* be found through test, the cost would be much higher. *(not only to find, but also to fix)***
- **Furthermore, the types of errors were those that are infeasible to test at the system level.**
  - **i.e.:**
    - **Intermittent Failures**
    - **Near-Simultaneous Failures**
    - **Combinatory Failure Sequences**

What Types of Errors did Formal Methods Find?

- Of the 12 errors found using formal methods, for this example:
  - 4 would be classified Severity 3 (Severities 1&2 affect safety of flight.)
  - 2 would be classified Severity 4
  - 2 resulted in requirement changes
  - 1 was redundant
  - 3 were not applicable (requirement not implemented in demo system)

- Given a comprehensive test program:
  - 1 of the Severity 3 errors would likely be found
  - Both of the Severity 4 errors would likely be found

Advanced Development Programs

Impact on Cost/Schedule

- The use of model checking results in a robust system design that reduces integration and static testing effort.
  - Integration and static tests can now be written at a higher level with fewer combinations of cases, thus allowing fewer tests to offer the same level of confidence as the original test plan previously would.

Where do we go from here?

- **Obvious Gaps**
  - **Completeness/Consistency of Requirements**
    - **Although Formal Methods provides an iron-clad analysis of specified system properties, the question remains: *"Do the properties adequately characterize the system?"***
  - **Sound and Thorough Risk Analysis**
    - **Complex inhabited systems assume a certain level of acceptable risk based on a pilot's training and awareness.**
    - **What is an equally acceptable threshold for software?**
  - **Technology and Process Integration**
    - **There is no single technique on which certification of advanced flight-critical systems can rely.**