

NASA's Exploration Software *Challenges and Expectations*

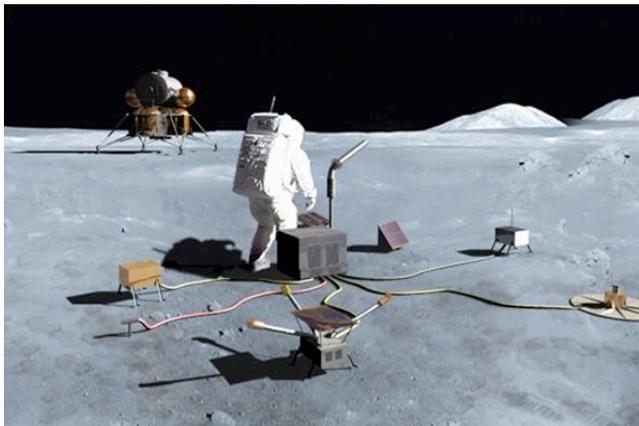
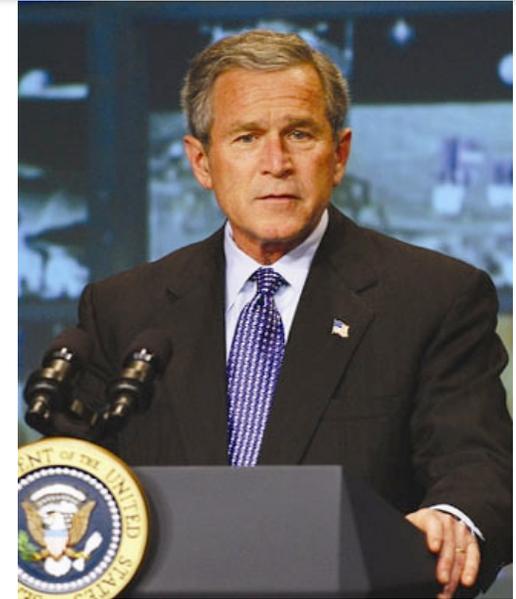
Matthew R. Barry, PhD, PE
Jet Propulsion Laboratory

Technical Integration Lead
Software and Avionics Interoperability and Reuse
Constellation Program Office
Johnson Space Center
National Aeronautics and Space Administration



A Bold Vision for Space Exploration, Authorized by Congress

- Complete the International Space Station
- Safely fly the Space Shuttle until 2010
- Develop and fly the Crew Exploration Vehicle no later than 2014 (goal of 2012)
- Return to the Moon no later than 2020
- Extend human presence across the solar system and beyond
- Implement a sustained and affordable human and robotic program
- Develop supporting innovative technologies, knowledge, and infrastructures
- Promote international and commercial participation in exploration



March 29, 2006

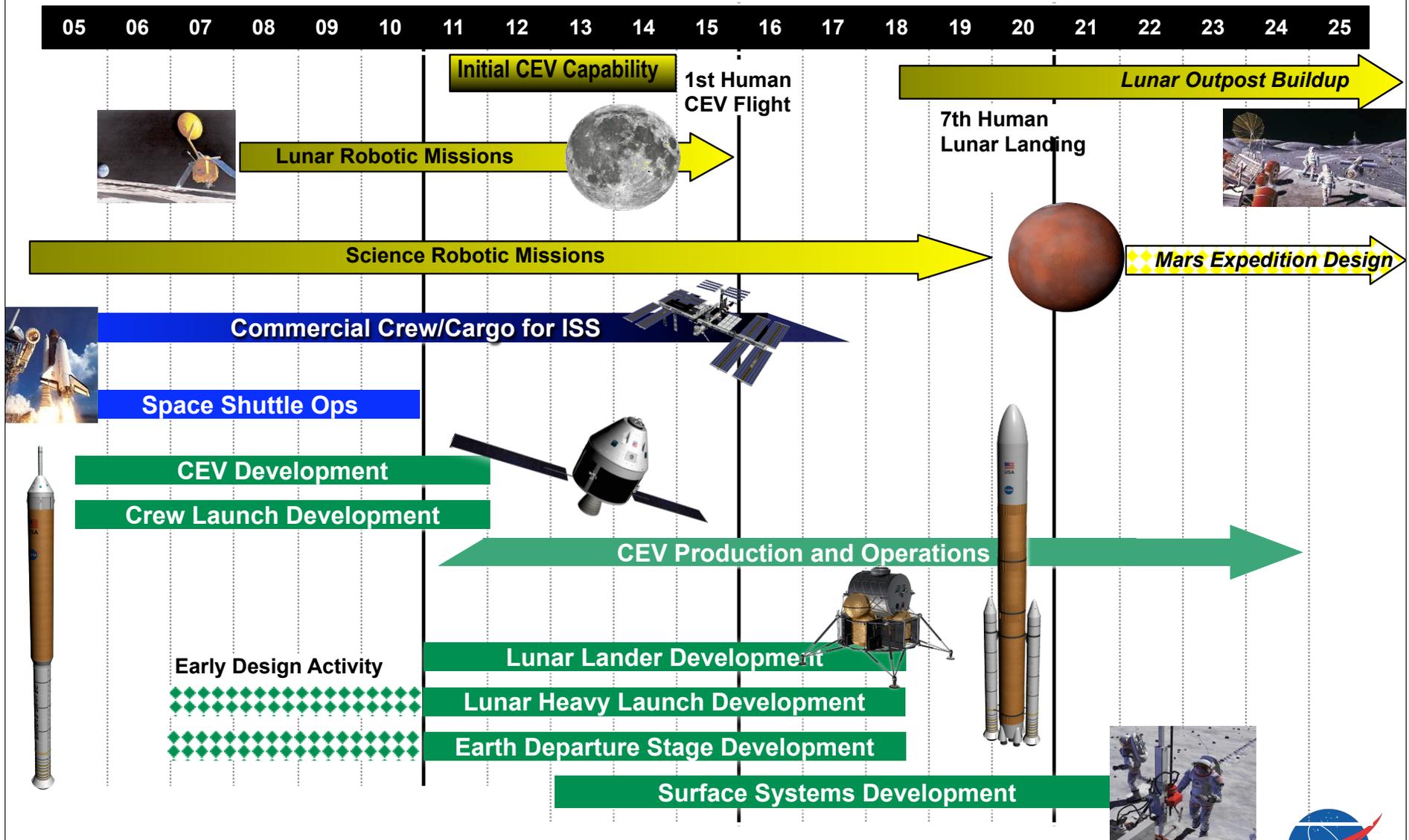
NASA Authorization Act of 2005

The Administrator shall establish a program to develop a sustained human presence on the Moon, including a robust precursor program to promote exploration, science, commerce and U.S. preeminence in space, and as a stepping stone to future exploration of Mars and other destinations.

NASA's Exploration Software Challenges and Expectations



NASA's Exploration Roadmap



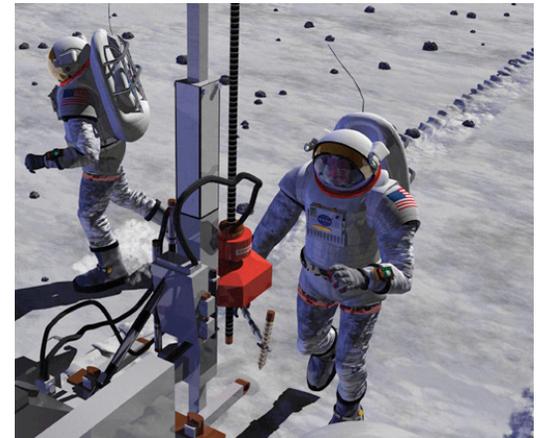
March 29, 2006

NASA's Exploration Software Challenges and Expectations



The Moon: First Step to Mars and Beyond....

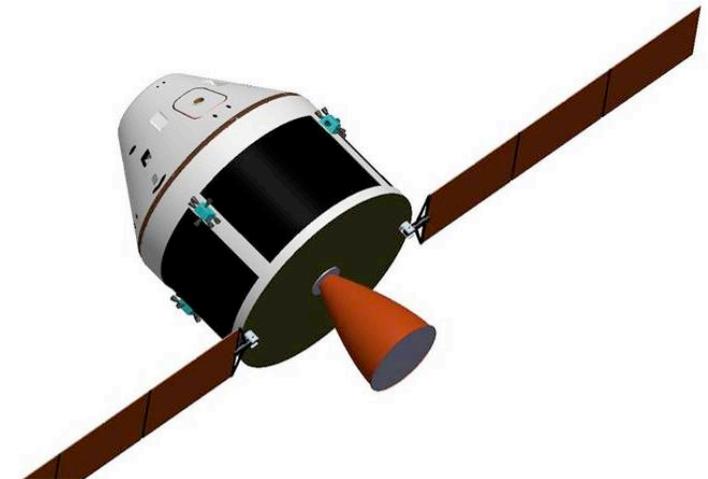
- Regaining and extending operational experience in a hostile planetary environment
- Developing capabilities for opening the space frontier
- Preparing for human exploration of Mars
- Science operations and discovery



Next Step in Fulfilling Our Destiny As Explorers

Crew Exploration Vehicle

- **Command Module**
 - Mold Line: Apollo-Derived Capsule
 - Crew: 6 for ISS & Mars, 4 for Moon
 - Size: 16.4 ft (5 Meter) Diameter
 - Docking Mechanism: APAS or LIDS
- **Service Module**
 - Propulsion: Industry Propose Best Solution
 - Some Capability for Delivering Unpressurized Cargo
- **Ongoing Analysis**
 - Impact of Reducing Volume
 - Trading Functionality between Command and Service Module
 - Eventual Migration to Non-Toxic Propellants

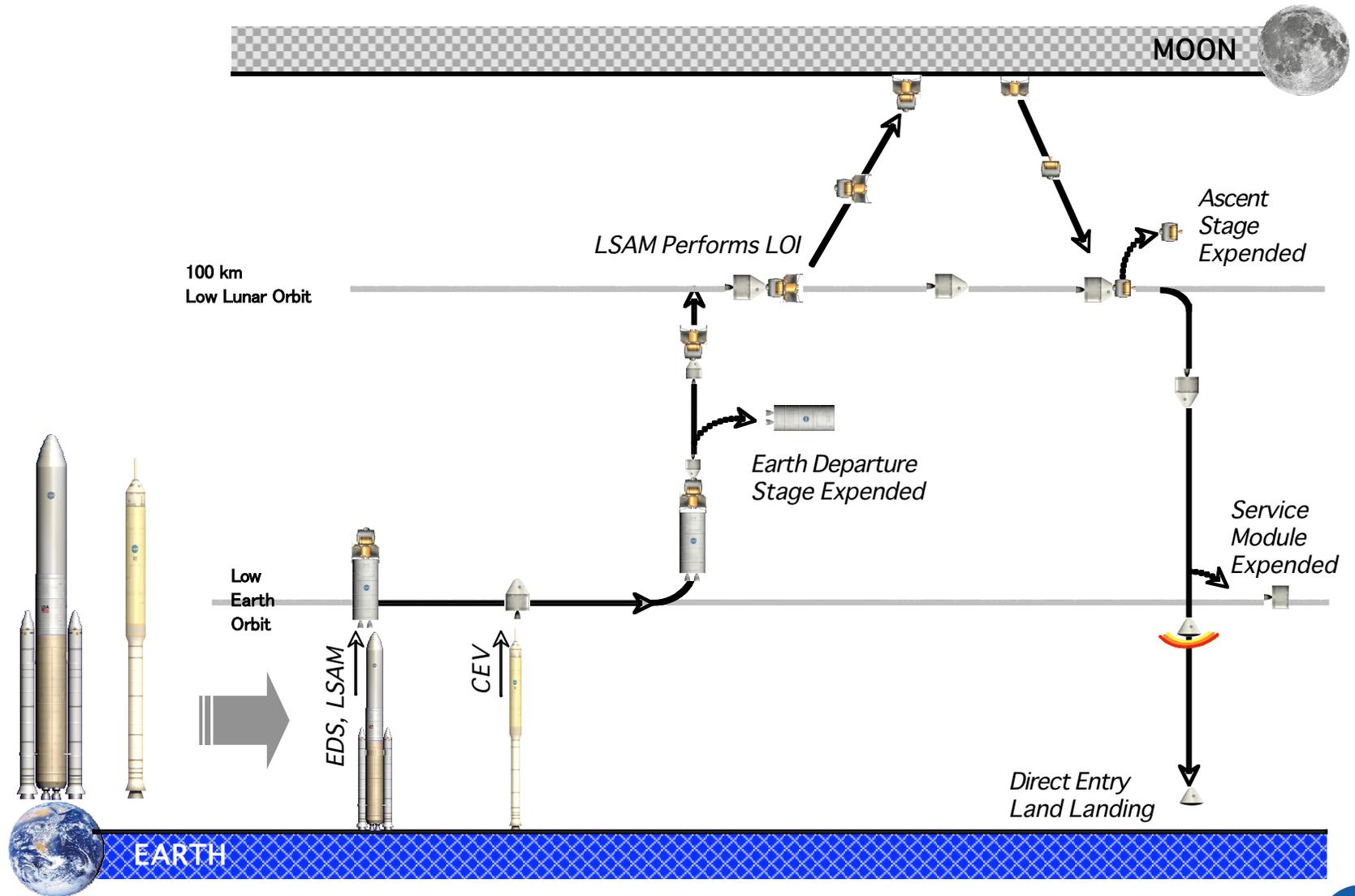


Crew Launch Vehicle Cargo Launch Vehicle

- Crew Launch Vehicle
 - Single 5 segment RSRB/M 1st stage
 - Upper stage powered by a single engine derived from the Saturn J-2
- Cargo Launch Vehicle
 - Twin 5 segment RSRB/M 1st stage
 - Core stage derived from the External Tank
 - Powered by 5 low cost SSMEs
 - CLV-derived avionics
- Earth Departure Stage
 - Upper stage derived from the External Tank
 - Powered by a single J-2 derived engine - 2 burn capability
 - CLV-derived main propulsion systems and avionics



“1.5 Launch” EOR-LOR



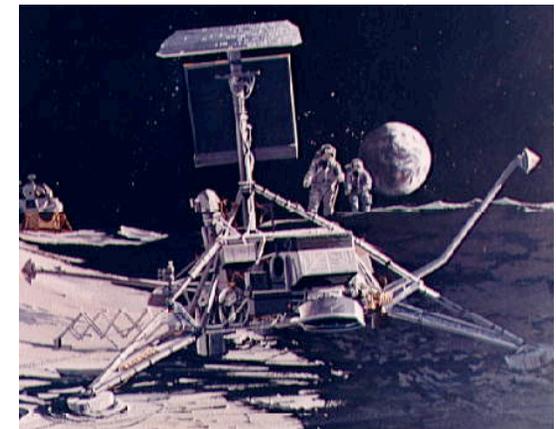
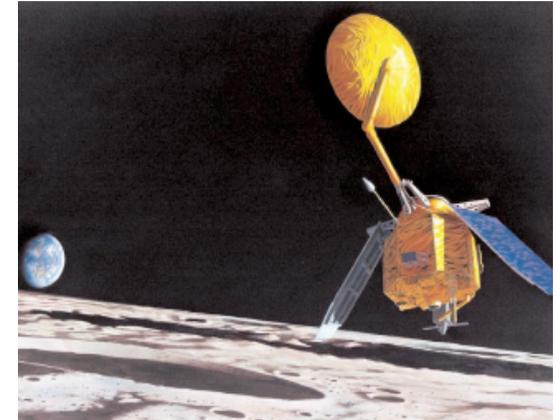
March 29, 2006

NASA's Exploration Software Challenges and Expectations



Robotic Lunar Exploration Program

- Provide early information for human missions to the Moon
- Evolvable to later human systems
- Most unknowns are associated with the poles - likely destinations for lunar outpost
- Key requirements involve establishment of
 - Terrain and surface properties
 - Knowledge of polar regions
 - Support infrastructure
- Lunar Reconnaissance Orbiter planned for 2008
- Major scientific and exploration benefit by 2009



Centennial Challenges

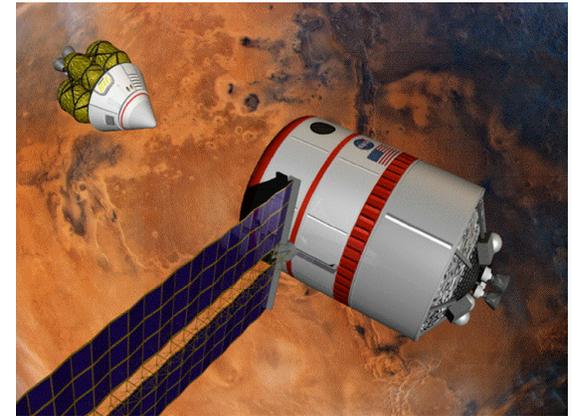
- Program of contests with cash purses to stimulate innovation and competition in technical areas of interest to space exploration and ongoing NASA priorities
- Four Categories of Challenges / Purses
 - Flagship / Tens of Millions
 - Keystone / \$.5 - 1 Million
 - Alliance / up to \$250,000
 - Quest / Promote science, technology, engineering, and math (all ages)

Commercial Crew/Cargo Project

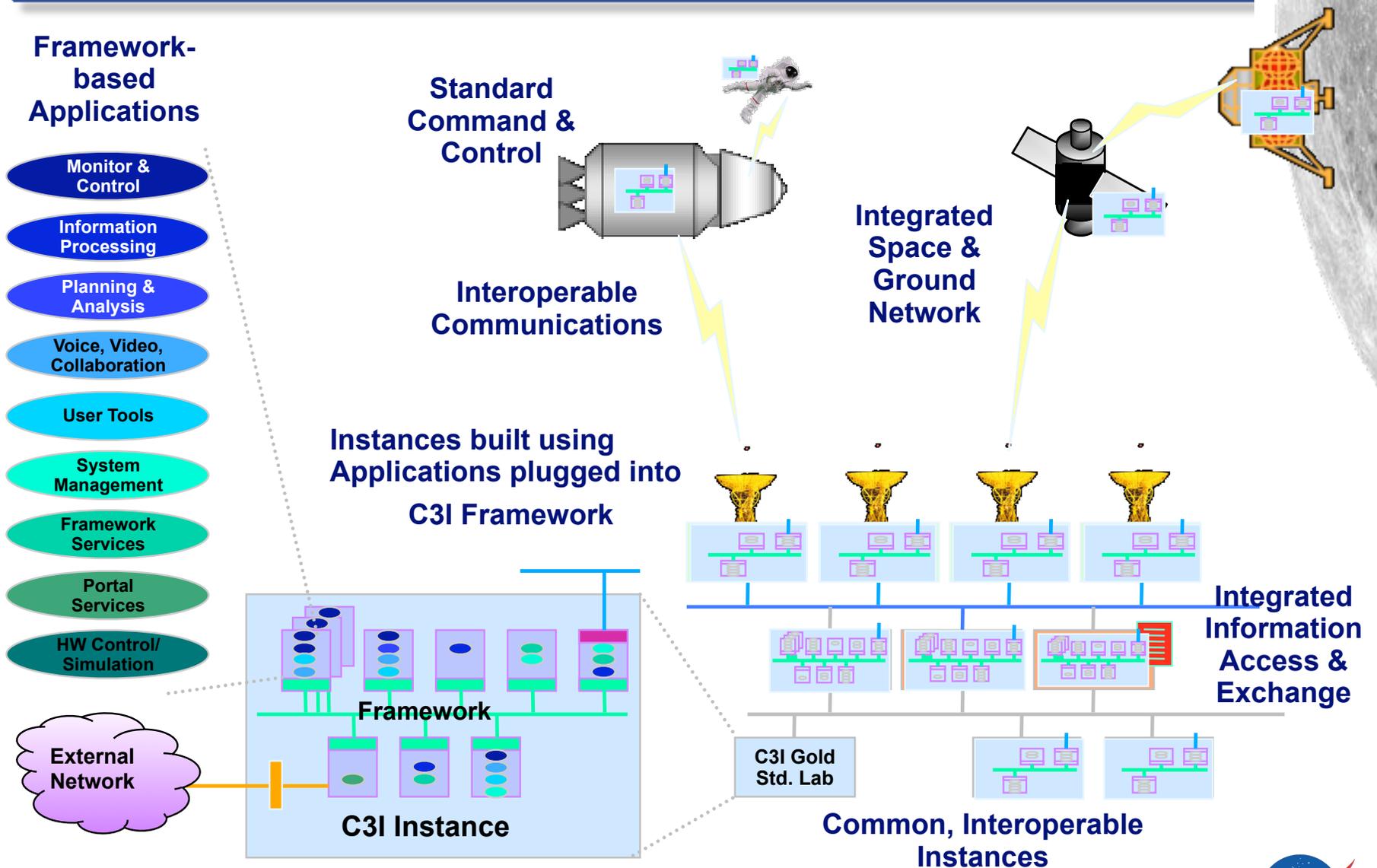
- Challenge to U.S. industry to establish capabilities and services to open new space markets
- May eventually support the transportation needs of the ISS
- Agreements expected to be awarded this summer

- **Acquisitions Well Underway**
 - Expected for summer 2006:
 - CEV: Downselect from two developers to single developer
 - CLV: Award of development contracts for 1st stage and upper stage engine
 - Commercial: Begin work under Space Act Agreements

- **Design Analysis Continuing**
 - Next cycle will report in April
 - Planning to baseline requirements in September
 - Includes analysis of lunar lander and propulsion systems



What is the C3I Architecture?

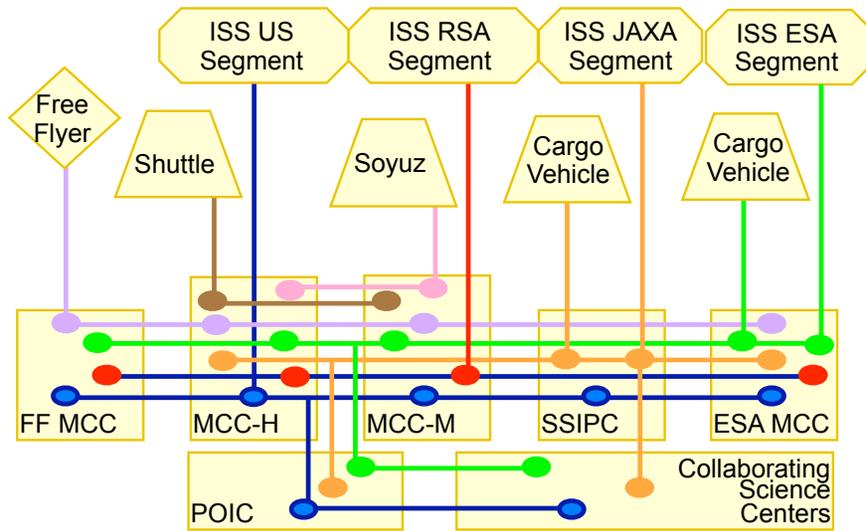


March 29, 2006

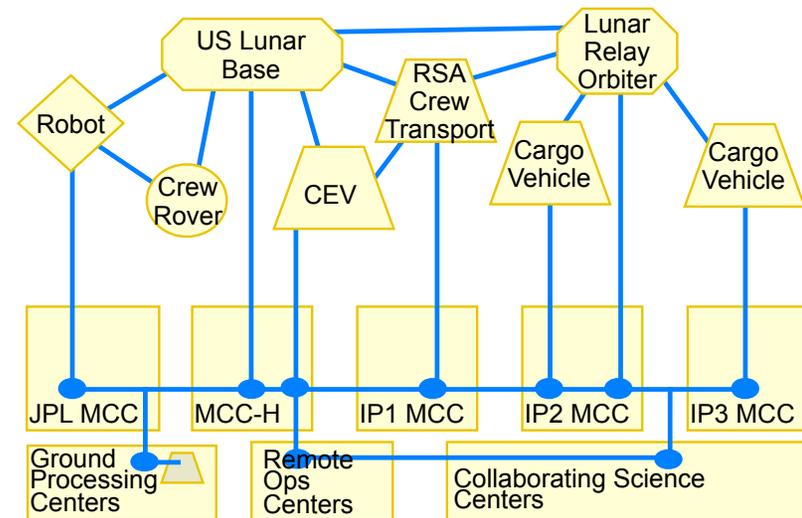
NASA's Exploration Software Challenges and Expectations



C3I Architecture Direction



Current ISS telemetry flow
partial – some are not shown



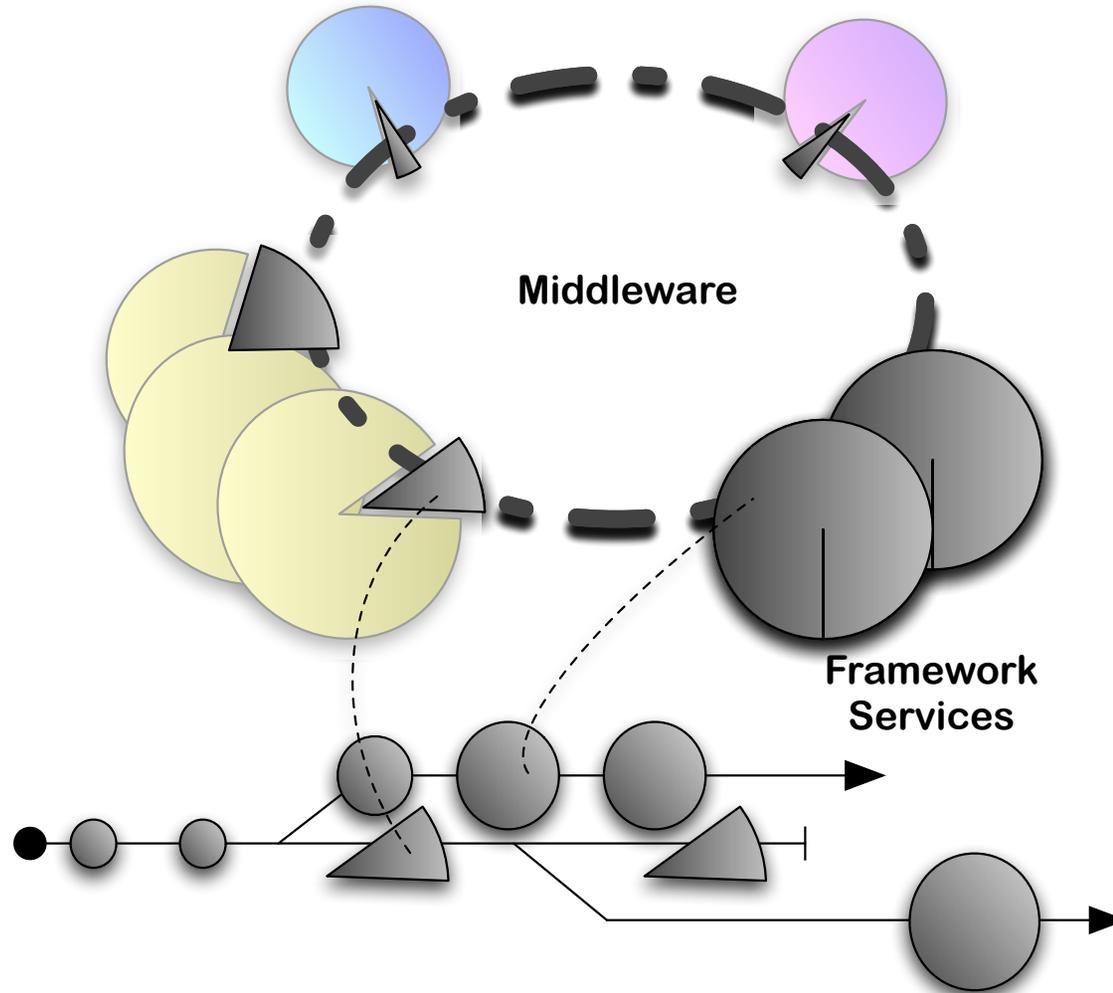
Proposed Constellation
C3I architecture

C3I Architecture Differences

Current NASA Systems	Projected C3I Approach	Major Differences
Point-to-point, manually configured/switched communications links.	Network-based communications including routing, relay, store and forward, and ad-hoc link establishment.	With number and geometry of Constellation elements, network provides less manual configuration, more flexibility.
Fixed configuration, hardware based communications equipment.	Standardized Software Defined Radios based on military JTRS and NASA STRS programs.	Enables interoperability and sustainability. Uses less RF equipment for more RF links.
Dedicated channels for data, voice, and video.	Digital network based voice and video (shared voice, video, data network).	More efficient use of bandwidth. Voice and video consumes bandwidth only when in use.
Site-specific data processing and distribution.	Common application framework integrating publish/subscribe data distribution (message bus) and information management.	Reduced cost of common system, enables flexible redundancy and evolution via loose coupling of applications.
Tightly integrated systems	Layered architecture	Ability to upgrade or change components with minimum impact
Command and telemetry databases, disconnected, no standard interchange mechanism.	Information management system using connectivity, common framework for enterprise level information access.	Information model and information management architecture provides operations payback in workflow automation.

Framework Potential for Software Product Line

Core Assets + Prescribed Evolution } **Product Line**
Management Process + Product Adaptation }



Safety-Critical Software

"System accidents are caused by interactive complexity and tight coupling. Software allows us to build systems with a level of complexity and coupling that is beyond our ability to control; in fact, we are building systems where the interactions among the components (often controlled by software) cannot all be planned, understood, anticipated, or guarded against. This change is not solely the result of using digital components, but it is made possible because of the flexibility of software. Note that the use of redundancy only makes the problem worse -- the added complexity introduced by redundancy has resulted in accidents that otherwise might not have occurred."

***Professor Nancy Leveson
MIT Aeronautics and Astronautics***

Source: Leveson, "The Role of Software in Spacecraft Accidents," in *AIAA Journal of Spacecraft and Rockets*, Vol. 41, No. 4, July 2004.

March 29, 2006

NASA's Exploration Software Challenges and Expectations



Risks Unique to Software

◆ Requirements related

- Ambiguous, contradictory, incomplete, or untestable requirements
- Invalid assumptions about context

◆ Design / system architecture

- Undefined or ill-defined interfaces
- Unqualified external dependencies

◆ Coding

- Inappropriate software reuse
 - limited access to original owners
 - lack of metrics
 - unresolved defects
 - change of context has unknown effect
- Choice of implementation language
- Choice of development/test tools
- Choice of developers/skills

◆ Effect of residual defects

- Risk of introducing new defects in maintenance

◆ Inherent complexity

◆ Mitigation contributors

- **Rigorous requirements process, with verifiable artifacts**
- **High-level design modeling and verification**
- **Verifiable software component libraries**
- **Proper metrics collection**
- **Avoiding complex languages**
- **Adoption of simple, robust coding rules**
- **Use of state-of-the art development, simulation, testing, debugging, and code analysis tools**
- **Disciplined multi-threaded design**
- **Strict coding standards**
- **Regression testing**
- **Automated test case generation**
- **Property based testing**
- **Model-driven verification**

System-of-Systems Software Issues

1. Acquisition Management and Staffing
2. Requirements/Architecture Feasibility
3. Achievable Software Schedules
4. Supplier Integration
5. Adaptation to Rapid Change
6. Software Quality Factor Achievability
7. Product Integration and Electronic Upgrade
8. Software COTS and Reuse Feasibility
9. External Interoperability
10. Technology Readiness

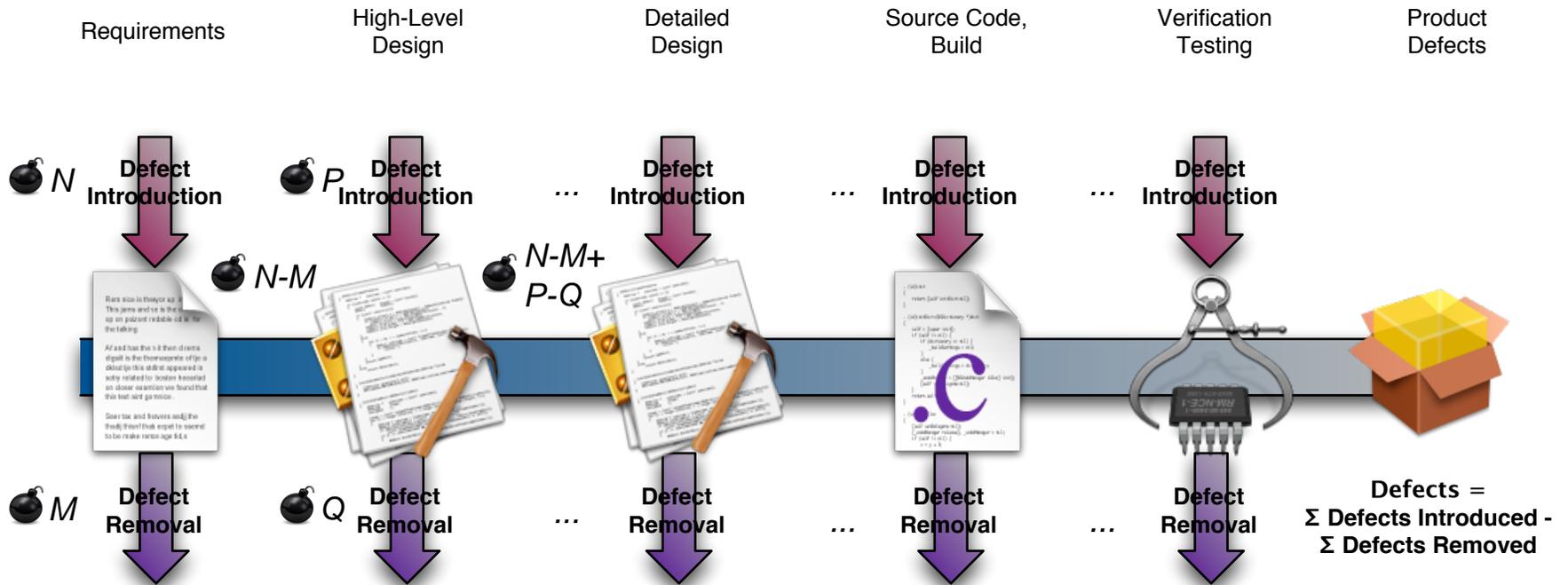
List source: Barry Boehm et al., "*Spiral Development of Software-Intensive Systems of Systems*," ICSE 2005

March 29, 2006

NASA's Exploration Software Challenges and Expectations



Software Defect Introduction and Removal



Each step in the software development process has an associated defect introduction rate and an associated defect removal rate. After filtering through successive steps, the defects remaining in the product is the sum of defects introduced minus defects removed. Improving the quality of the product requires attacking each step in the process, decreasing each step's defect introduction rate and increasing each step's defect removal rate.

Adapted from: Gerard Holzmann briefing, NSS 2005.

Software Engineering Emphasis

- Focus on drivers to safety and quality
- Use software architecture as development coordination tool
- Ensure proper understanding of architecture and evolution
- Make decisions based on risk imposition or elimination
- Impose frequent requirements feasibility demonstrations
- Require safety-critical software hazard analysis
- Emphasize model-based engineering
- Implement defect prevention technologies
- Develop measures of assuredness for product quality
- Focus on tool chains, training, staff, and sustainability

- **Interoperability**
 - High-level architecture description
 - Significant contribution of C3I team
 - Legacy and COTS integration issues
- **Reuse**
 - *Planned* reuse through software product lines
 - Component-level integration and certification
 - Safety and product quality demonstration techniques
- **Advisors**
 - Technical Integration Lead “owns” the requirements
 - Agency-wide Systems Integration Group handles the issues
 - External expert advisory groups envisioned

NASA's Exploration Software *Challenges and Expectations*

Matthew R. Barry, PhD, PE
Jet Propulsion Laboratory

Technical Integration Lead
Software and Avionics Interoperability and Reuse
Constellation Program Office
Johnson Space Center
National Aeronautics and Space Administration

