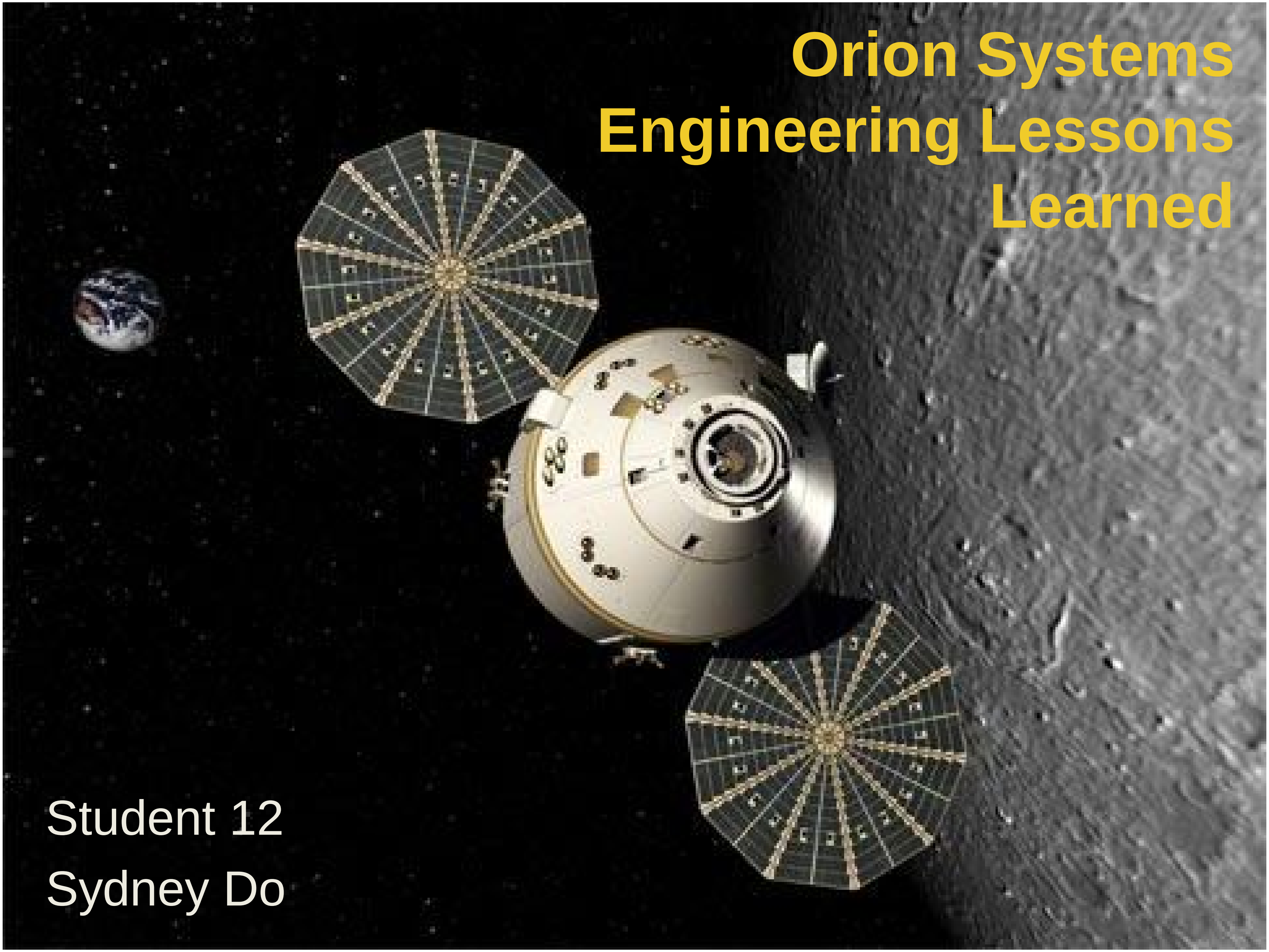
Orion Systems Engineering Lessons Learned



Student 12
Sydney Do

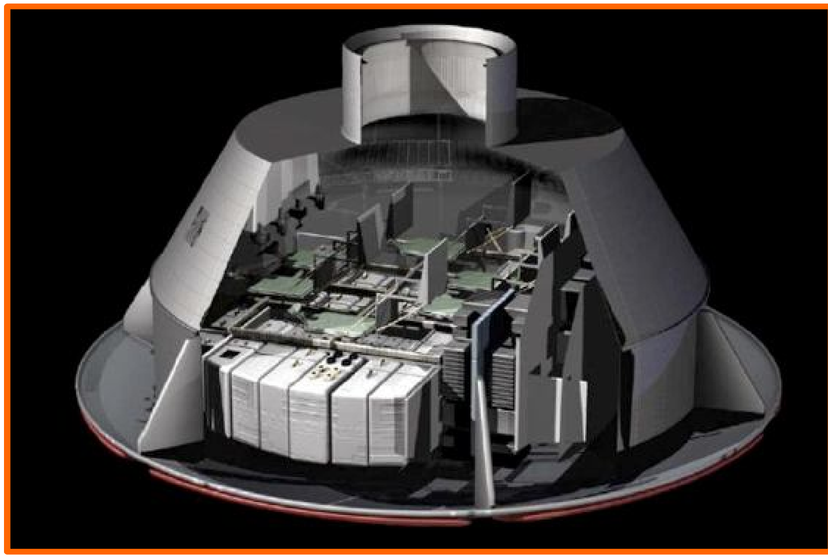
Background and History of Constellation Program

- Spacecraft design currently under development by NASA and Lockheed Martin
- Designed to be launched by the Ares I
  - Ares I also currently under development in parallel
- Both Orion and Ares I are elements of NASA's Project Constellation
  - Send human explorers back to the Moon by 2020
  - Onward to Mars and other destinations in the Solar System
    - May change as the results of the Augustine Commission are currently under review

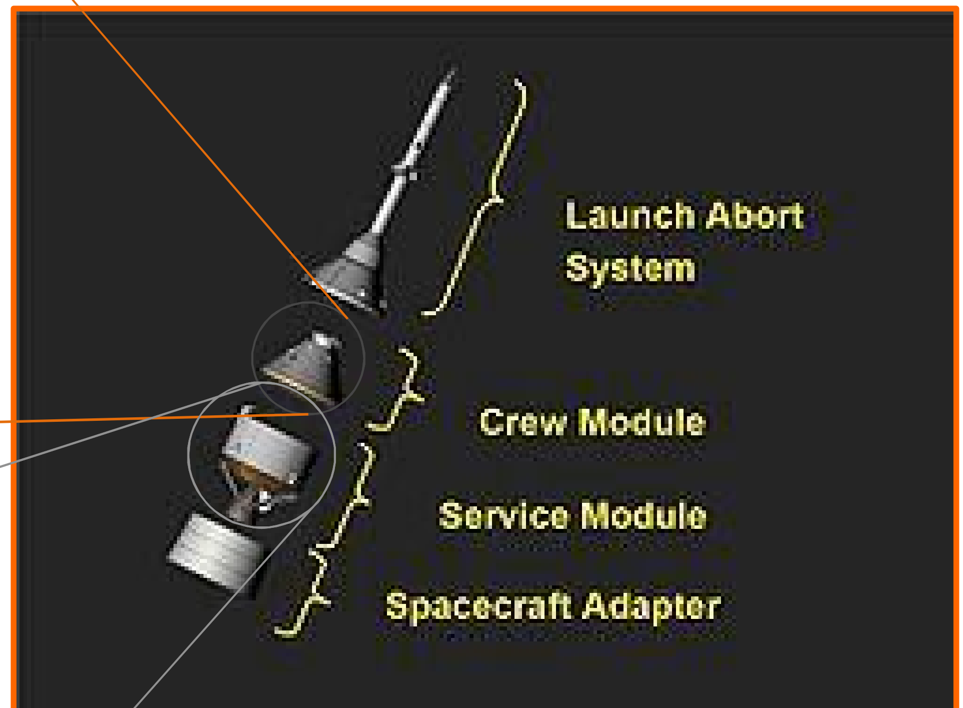
Breakdown of Orion Program

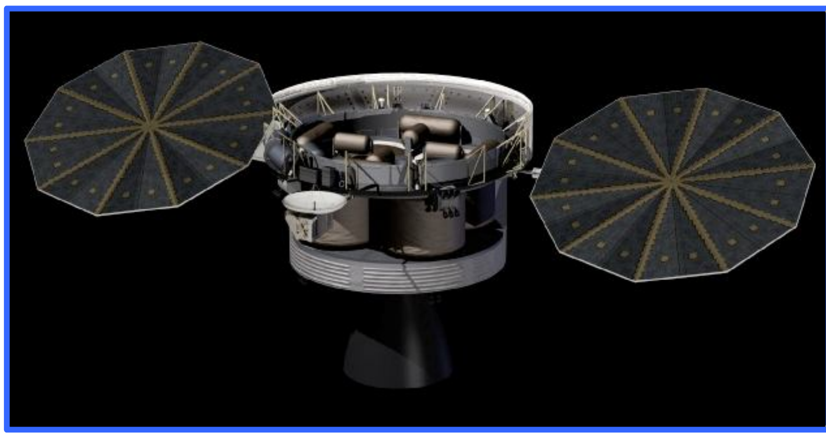
- On January 14, 2004, President Bush announced the Orion spacecraft as part of the Vision for Space Exploration
  - partly a reaction to the Space Shuttle *Columbia accident*
- Lockheed Martin contracted to develop and build Orion spacecraft
- PDR was in September 2009
- CDR planned for February 2011

Orion Physical Architecture



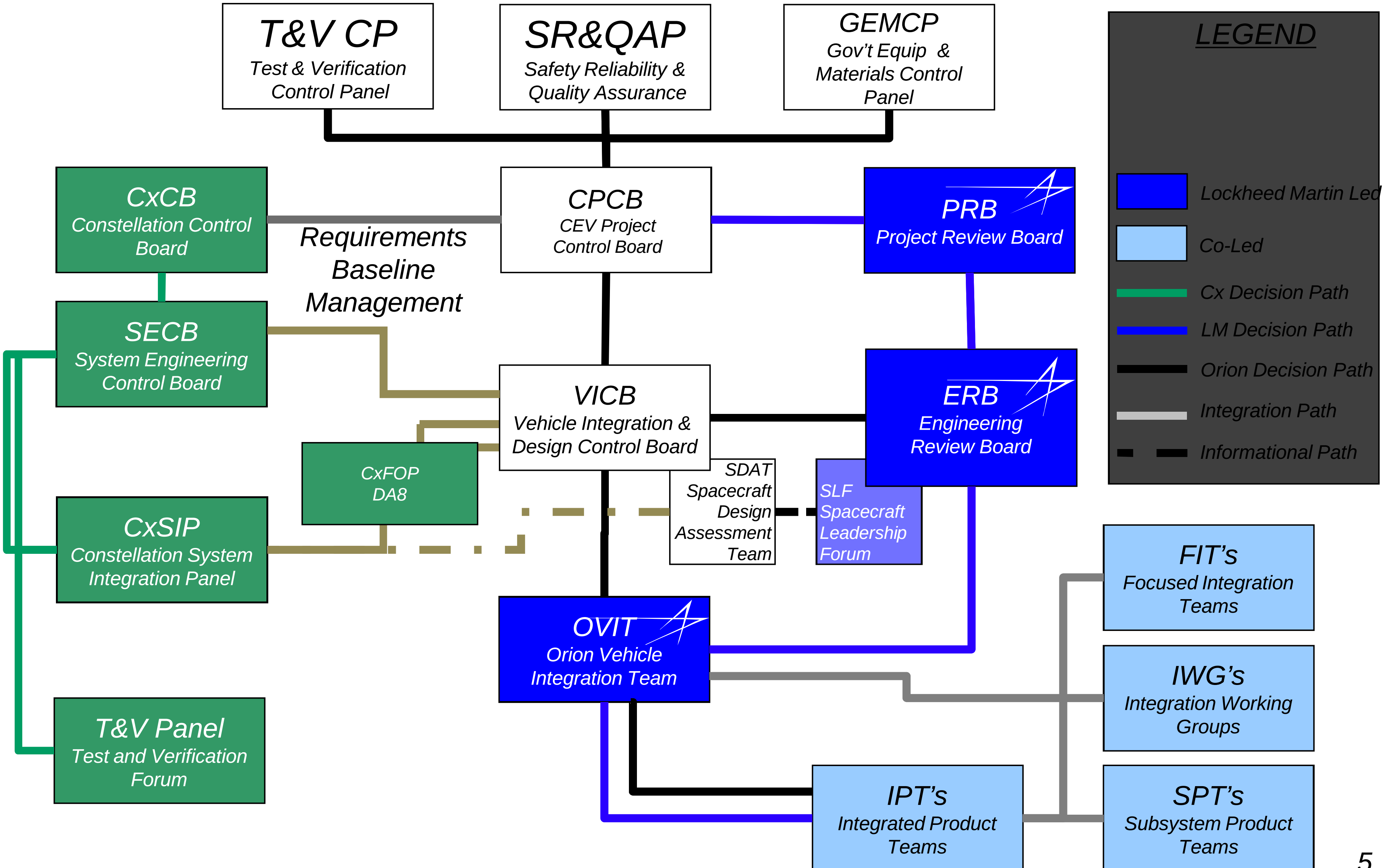
- *Four to six crew members*





- *Service module is primary power and propulsion component*
  - *discarded prior to re-entry*

Orion Program Architecture



Systems Engineering Challenges

- Every requirement costs money and every change to a requirement costs money
  - NASA is always budget limited and program has been through may changes already due to changing administrations (Augustine commission has implications on manned missions)
  - Augustine reviewing where we're going, why we're going there and how we're going to get there
- Scale and scope of Constellation Program (CxP)
  - Seven projects and over 10,000 people
  - Demands frequent and often redundant communication
- Trying to design in flexibility to adapt to future needs that are not currently known
  - Important in current political and budgetary situation → makes design much more complicated due to unknown unknowns

Lessons Learned in Orion Program

- Make requirements traceable back to stakeholder need and expectations → make sure they know what you're doing
  - "Inclusionism" defined for discussing and explaining rational to stakeholders
- Actively manage requirement discontinuities → "burn down" TBDs and TBRs by PDR
  - If no traceability, then kill it
  - Requires strong collaboration between NASA and Lockheed
  - Top level TBDs and TBRs hold up people working at subsystem level
- Clarity of functional requirements from NASA to Lockheed is essential
  - Formal clarification requests from Lockheed to NASA implemented
  - Clarification, not validation (could be misinterpreted as feedback loop but is actually ensuring full information transfer. Pushing back on requirements would be a feedback loop).

Lesson Learned in Orion Program (cont.)

- Due to complexity of system, functionality gaps can appear in unexpected places (ex. Off nominal operational modes could be overlooked)
- Horizontal integration of requirements → make sure all requirements are understood by everyone
  - Requirements centric culture makes people think they are exempt from requirements not immediately relevant to their subsystem
  - Build community of trust to reduce requirements centric culture
- Requirements changes take time to incorporate and validate
  - May need to allocate/decompose down to the component level, and roll back up to the spacecraft level before identifying issues
    - Primary impacts are relatively easy
    - Secondary and tertiary impacts take longer to identify

Operability and Design

- One ops, one voice (ground ops, flight ops, flight crew).
  - Impacts on operability
- Operations & Support should influence the design process
- Many examples where this has affected the system design at component level
  - Example: Orion stack capable of integrated lift to save 70 hours in integration critical path schedule

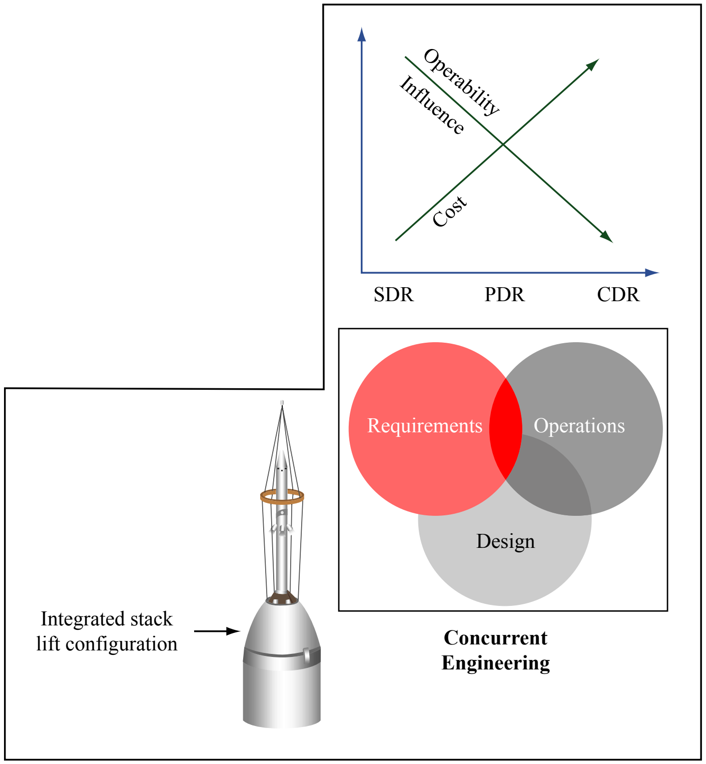
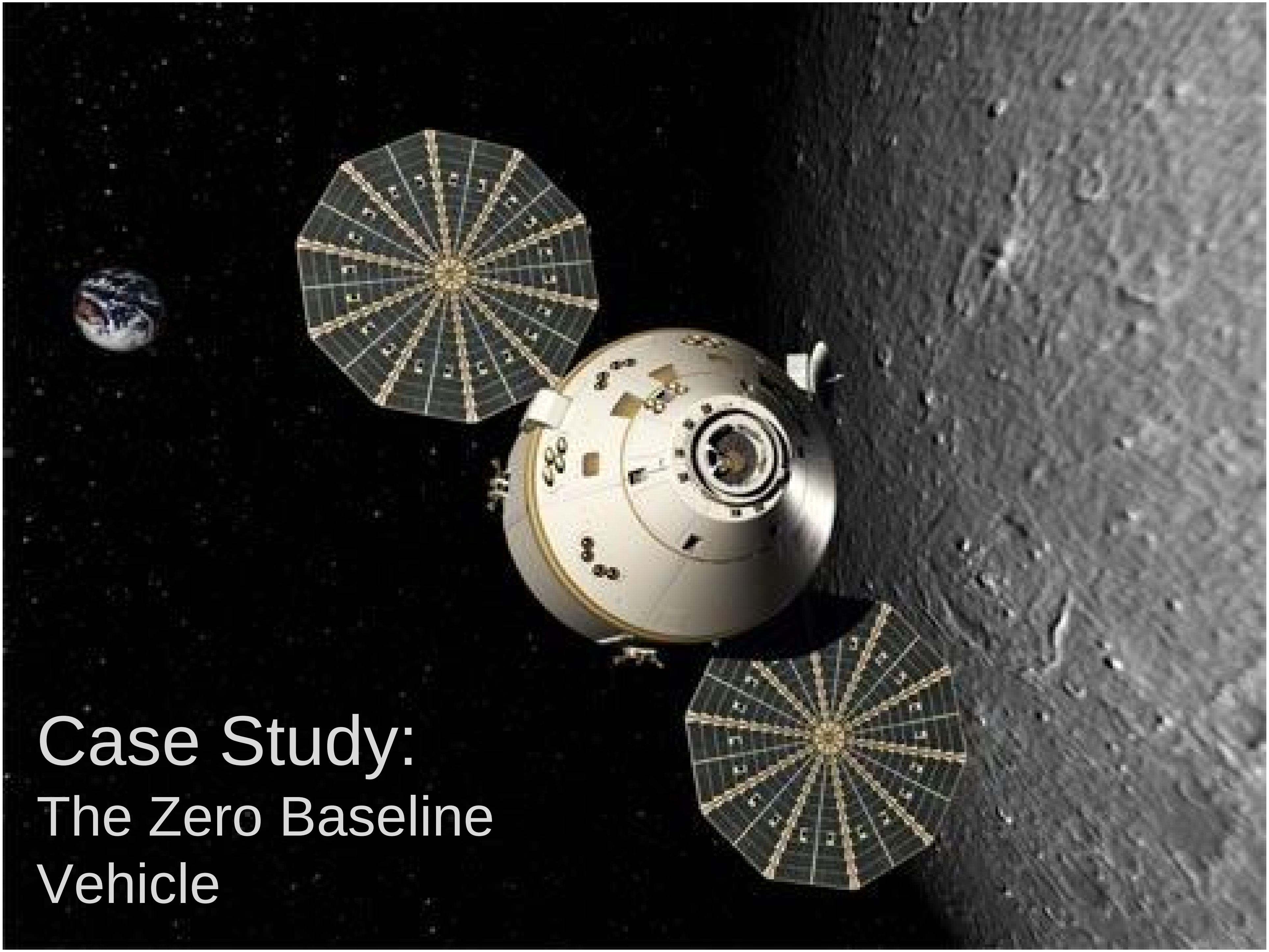


Image by MIT OpenCourseWare.


Case Study:
The Zero Baseline
Vehicle

Background & Motivation

Traditional spacecraft design has focused on multiple fault tolerance

- Redundant systems and software
- Increases system complexity, which introduces more components and their interactions (and their potential failure modes)
- Can reduce system safety and reliability
- Eg. Space Shuttle. Safety issues are **still** being uncovered today
  - Cracks in GH_2 flow control valve poppet discovered after STS-126 in November 2008.
- This principle was originally applied early on in the design of the Orion CM

This cut-away diagram of a space shuttle has been removed due to copyright restrictions.

The Mass Problem

- As the design of Orion and Ares I matured, it became apparent that the vehicle would not meet its mass budget
  - Early-Mid 2007 – Ares I found to provide insufficient thrust to transport the then Orion baseline design into orbit
    - Increased mass constraints on the Orion vehicle design
  - Early 2008 – Thrust oscillation issues identified in Ares I. All solutions to this require additional mass on Ares I, thus further constraining Orion's mass budget
- Keeping the vehicle to within an increasingly constrained mass budget has been a significant challenge
- Maintaining multiple fault tolerance is very costly in terms of mass
  - A new design methodology was required, which resulted in a lower mass system while maintaining its safety and robustness

The Zero Baseline Vehicle (ZBV)

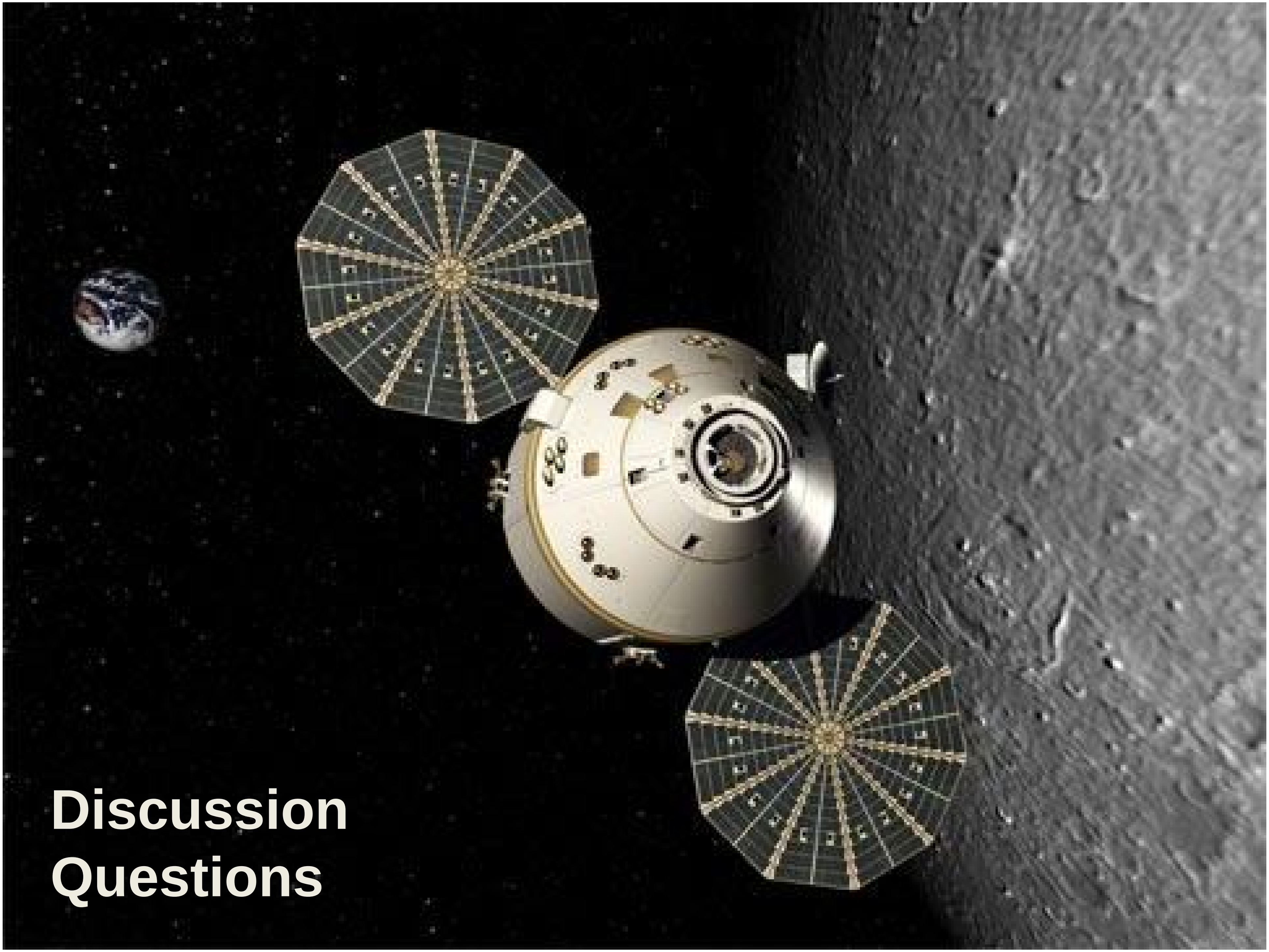
- Design a zero fault tolerant vehicle with the minimum capabilities required to meet top level mission requirements
- Identify the most safety critical areas in the design
  - Will a failure in this component cause a loss of mission or crew?
- Modify the design of these areas to address their failure modes

Observations

- Performing this process uncovered many safety critical areas which were not explicitly captured in the system requirements
  - Allows you to track the effective value per unit mass
- The end result is a “single fault tolerant system which is better than a two fault (tolerant) design”

In essence, mass is being used as a currency to purchase fault tolerance where it provides the most value

- This methodology has been adopted in the preliminary design of the Altair Lunar Lander


Discussion
Questions

Discussion Questions

- How do you efficiently facilitate strong communication between the different design teams?
  - During the early parts of design where ambiguity is at a peak and requirements are susceptible to change?
  - During the latter stages of a project where complexity becomes more difficult to manage

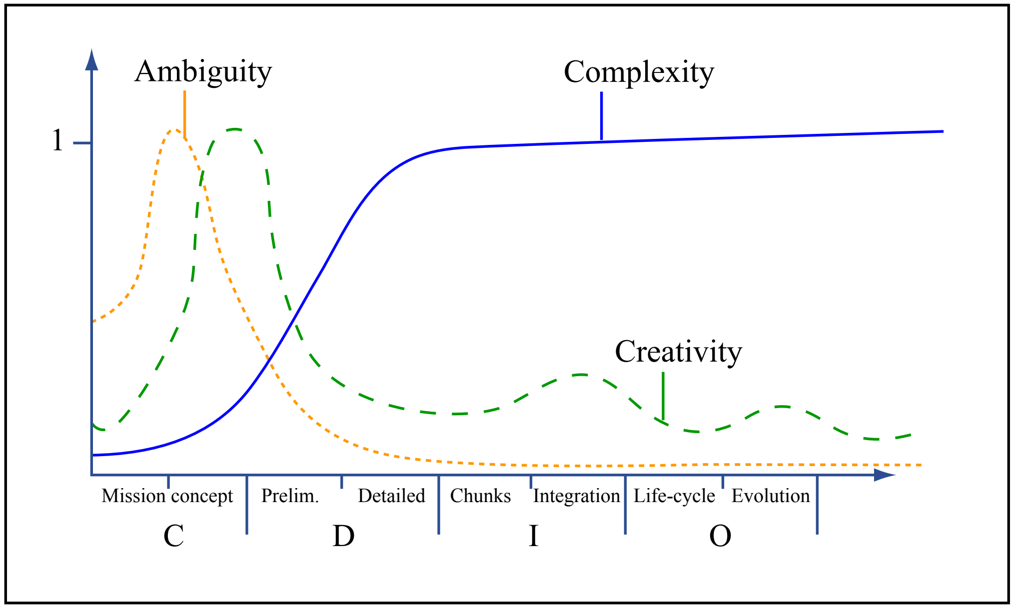


Image by MIT OpenCourseWare.

Discussion Questions (cont.)

- Do you think the ZBV technique is applicable to terrestrial programs?
  - Is it only applicable to programs where mass is the most significant constraint?
  - Can the technique be used with other design parameters in the same manner?
  - When does it become worthwhile to do this?

MIT OpenCourseWare
http://ocw.mit.edu

16.842 Fundamentals of Systems Engineering
Fall 2009

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.