GPS Control Segment Modernization – Acquisition of a Large Software Intensive System

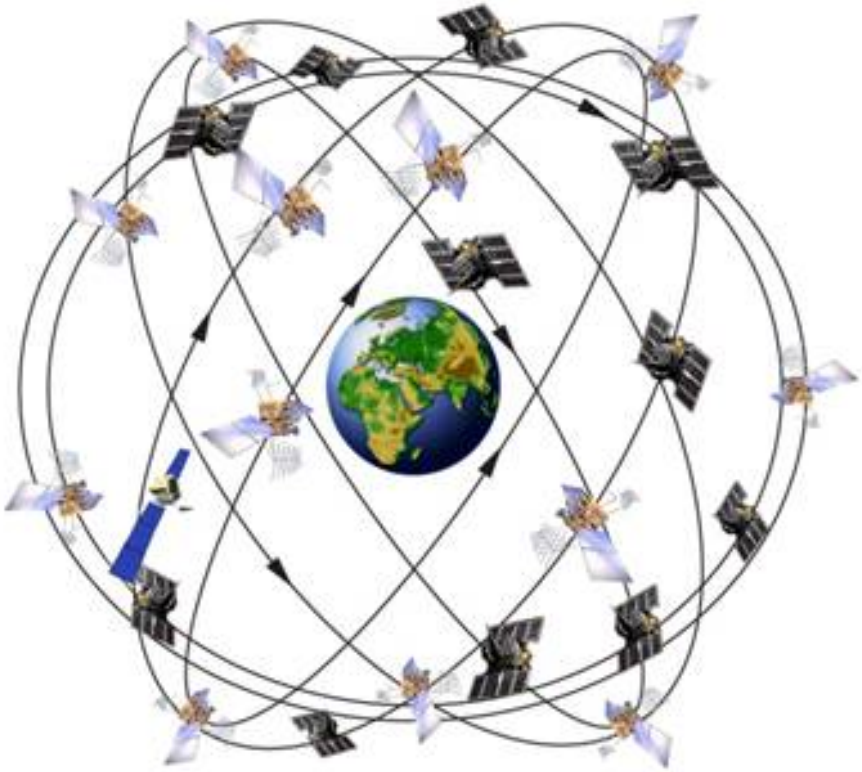
Alexander J. Polack
Principal Director
Control Systems Sub-Division
Navigation Division
The Aerospace Corporation

IEEE/STC
March 2014

Agenda

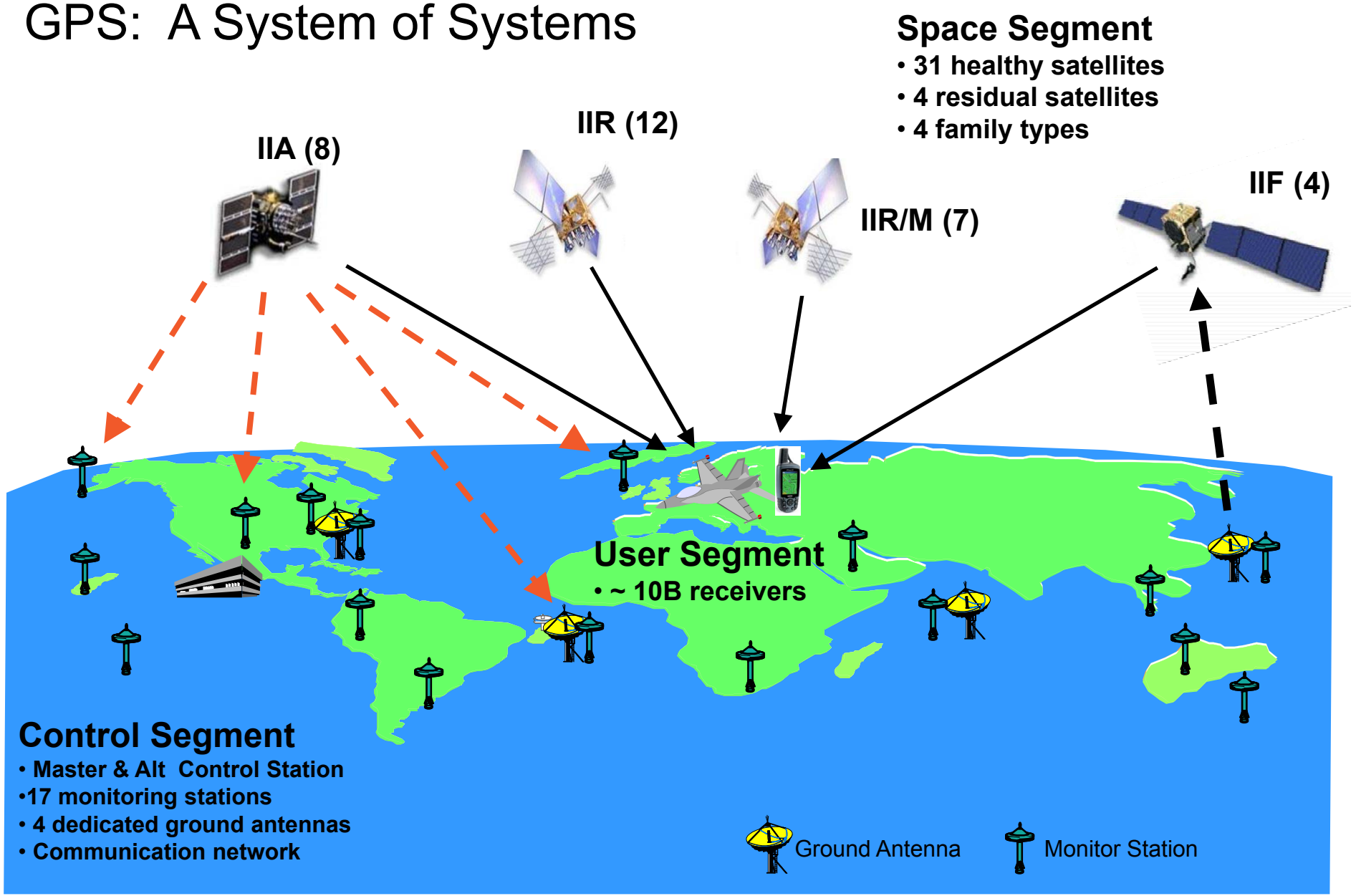
- GPS Overview
- Control Segment Role
- History of Control Segment
- Overview of OCX
- OCX Acquisition Approach
- PDR Approach
- CDR Approach
- Lessons Learned

What is GPS & How It Works



- Satellite-based radio navigation system
- There are always a minimum of 24 GPS satellites in orbit
- Every spot on Earth can see at least 4 satellites
- A GPS receiver uses position of 4 satellites to solve for its own 3-d position and time

GPS: A System of Systems



AEROSPACE

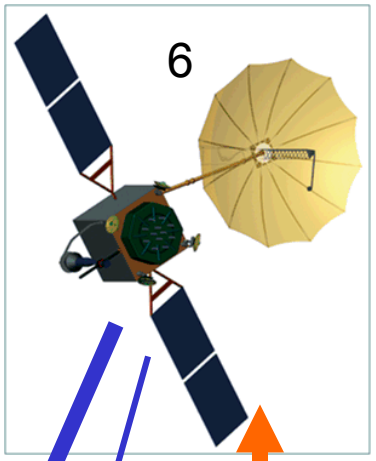
GPS Control Segment: The Heart and Brains of GPS

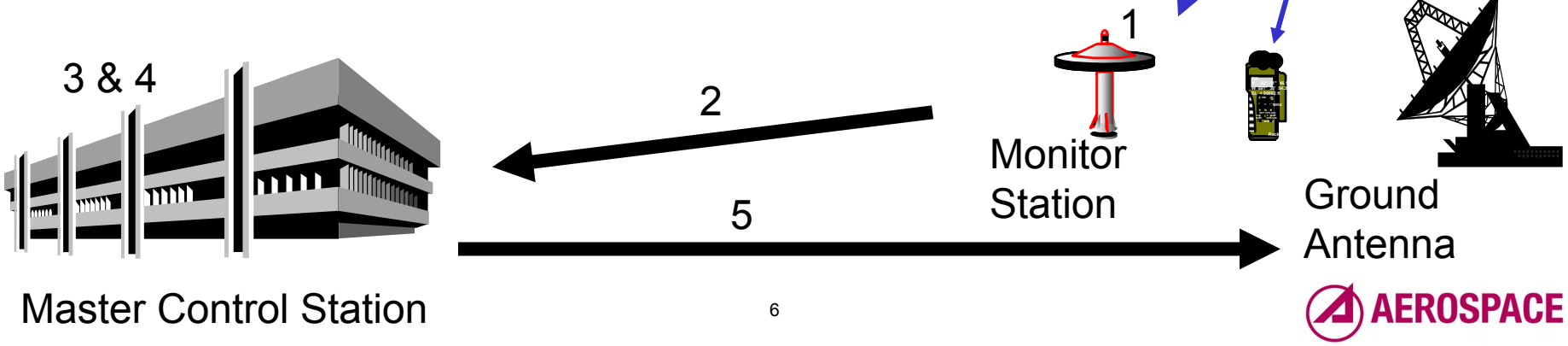
Roles and Responsibilities

- Launch and Checkout of new satellites
- Support The Mission
  - *Orbit Determination – Ensures that each satellite broadcasts the most accurate estimate of its current location in space*
  - *Clock Synchronization – Ensures that the "GPS Time" broadcast by each GPS satellite is synchronized with all the others*
  - *Satellites maintenance*
    - Ensures that no GPS satellite broadcasts dangerous or misleading data
    - Fix problems before they cause trouble
    - Take bad satellites off the air
    - Dispose of old satellites

GPS Control Segment Mission Rhythms

1. Monitor Stations (MSs) at fixed, known locations take measurements from satellites in view (L-Band)
2. Data is piped from MS's to the Master Control Station (MCS) for processing
3. MCS uses this data in Kalman Filter at 15 minute intervals to estimate precise clock and satellite orbital parameters
4. Navigation uploads are built based on updated data
5. OCX uploads each satellite with navigation data at least once per day
6. Satellites build NAV message
7. Satellites broadcast NAV message to users and MSs





AEROSPACE

GPS Ground Segment Evolution

GPS Ground Segment Evolution	Timeframe	GPS Space Segment Evolution
Demonstration System • Manually intensive operations • Closed system	1970s - 1980s	Block I
First Generation Ground System • Mainframe-based architecture • 10's of external users	1980s - 2007	Block II IIA/IIR/IIRM
Second Generation Ground System – OCS • Client-server architecture • Dozens of external users	2007 - 2016	Block II IIA/IIR/IIRM/IIF
Next Generation Operational Control System – OCX • Flexible, modular, scalable architecture • Net-Centric interfaces • 100's of external users and growing	2016 -	Block II-III IIR/IIRM/IIF/III

OCX Tenets

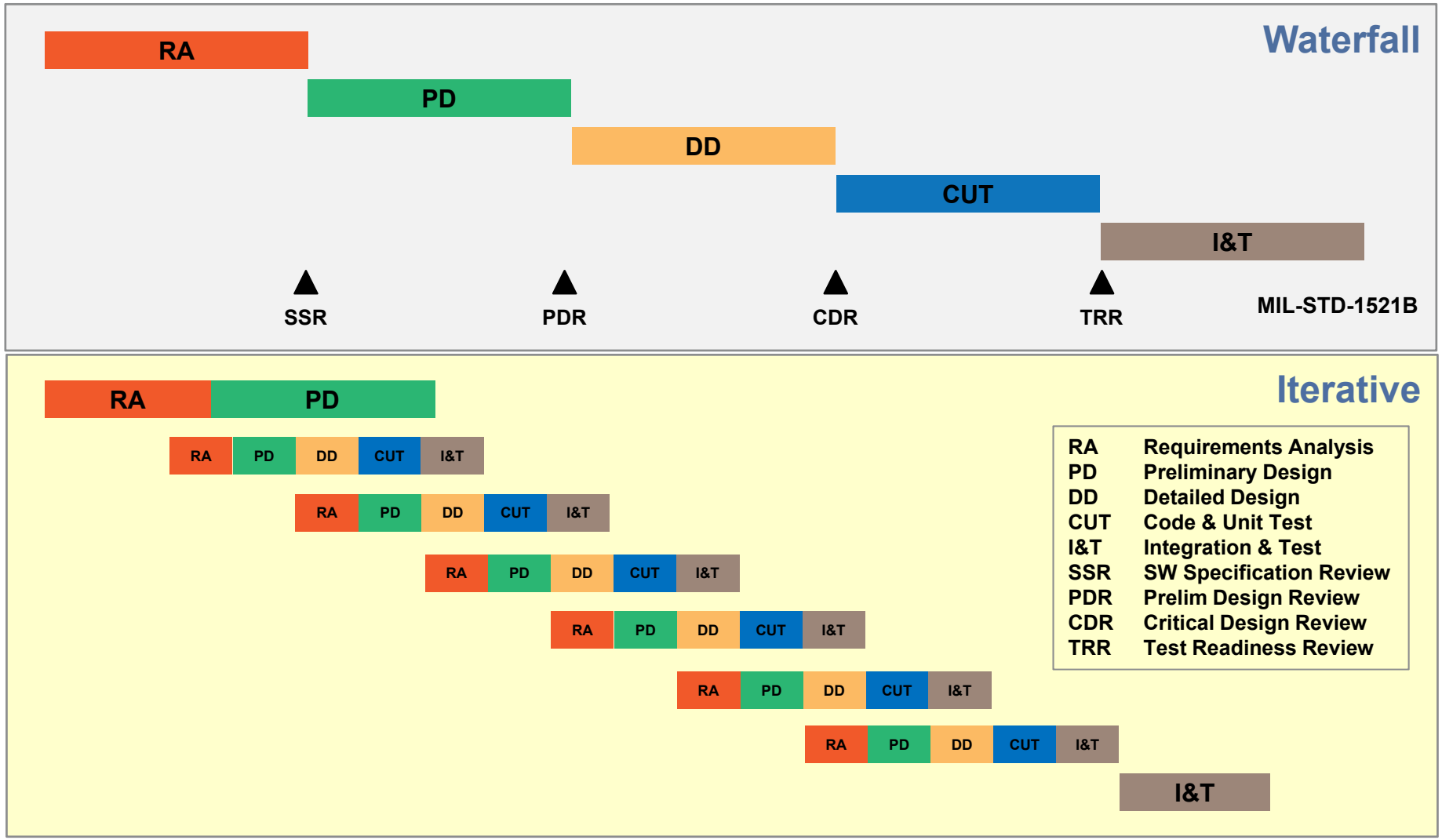
- Provide robust GPS Control Segment foundation
  - *Ability to continue mission support after changes to the mission itself*
  - *Independent of just flying satellites*
- Flexible and expandable architecture
- Robust Information Assurance an integral part of the architecture
  - *Facilitate response to current and potential future Cyber threats*
- Support future GPS Enterprise architecture needs
  - *Current & future changes to GPS mission*
  - *Address more and different payloads in different orbits, hosted payloads, terrestrial objects*

OCX Architecture is critical for success

OCX – General Program Observations

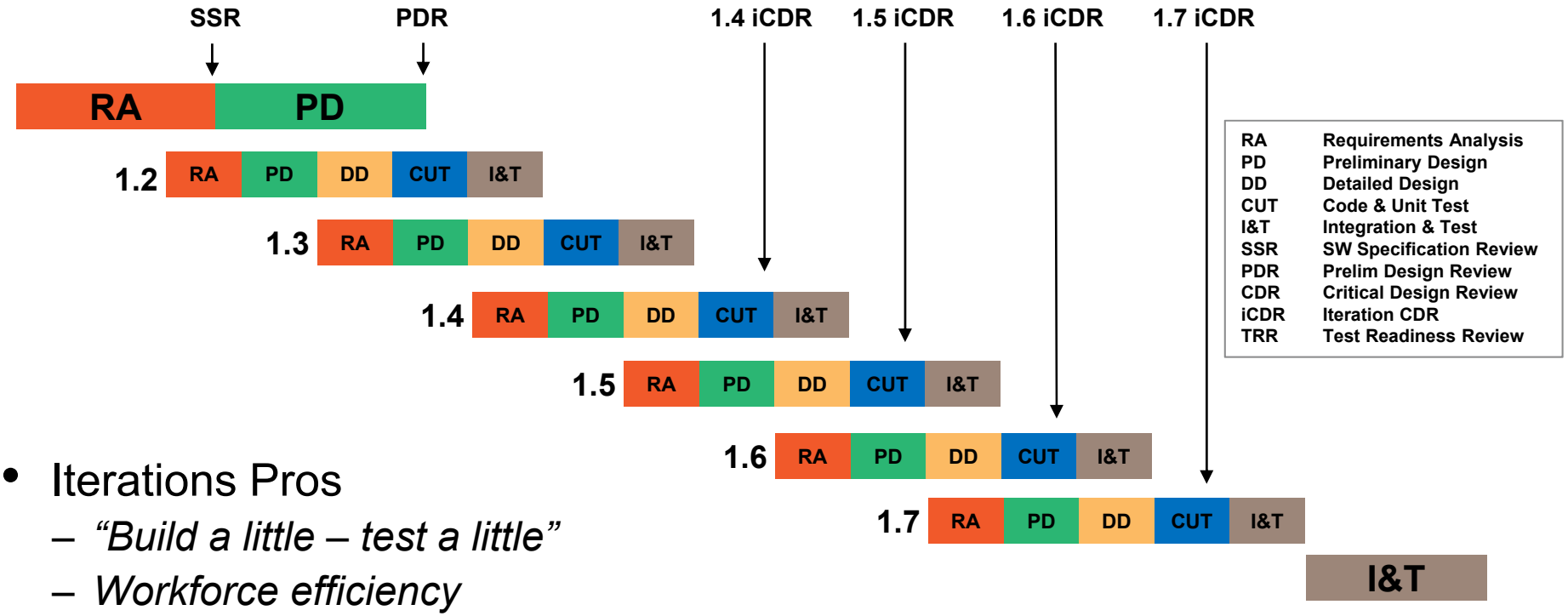
- Contract awarded 25 Feb 10 to Raytheon IIS, Aurora, CO
- Software intensive system ~ 1.1 M ESLOC
- Multiple Blocks with multiple iterations within each block
- “Back to basics” for Government oversight and insight
- Mixed iterative/“waterfall” approach for Systems Engineering
- Traditional “waterfall” approach to contractual oversight
  - *SSR, PDR*
- Hybrid (waterfall/iterative) approach for CDR
  - *iCDR: Software Development (Iterative)*
  - *COTS CDR: Software/Hardware COTS (Waterfall)*
  - *Hardware CDR: Developed Hardware (Waterfall)*
  - *VCDR: Verification Program (Waterfall)*
  - *TCDR: Transition Program (Waterfall)*

Waterfall vs. Iterative Software Development Models



Iterations: Reduced Risk via Opportunity for Adjustments and Early Integration

OCX Iterative Software Development – Block 1



- Iterations Pros
  - *"Build a little – test a little"*
  - *Workforce efficiency*
  - *Flexibility and Early Integration*
- Iteration Challenges
  - *Level of upfront systems engineering effort*
  - *Need for solid SW architecture*
  - *System perspective – requirements movement between iterations*

Planning is crucial

PDR Campaign Approach

- PDR campaign designed to minimize PDR event risks
- “Raw” Artifact Reviews
- Planned PDR Focused Reviews (PFRs)
  - *Special topic “Deep Dives”*
  - *Architecture, Nav, IA, C2/Sim, E2E, RMA, HSI, EIS, PSICA*
- Draft CDRLs review
- Formal CDRLs review
  - *133 CDRLs*
  - *Assess all CDRLs with respect to tailored MIL-STD-1521B criteria*
- PDR readiness assessment gates
- Defined CRM Process
  - *Start with a standard CRM Process*
  - *Multiple related comments binned into PDR Issue Notices (PINs)*
  - *PINs linked to tailored MIL-STD-1521B(T) criteria (114)*
  - *PINs and CRMs delivered to Contractor for resolution*
- Clearly defined PDR Exit criteria

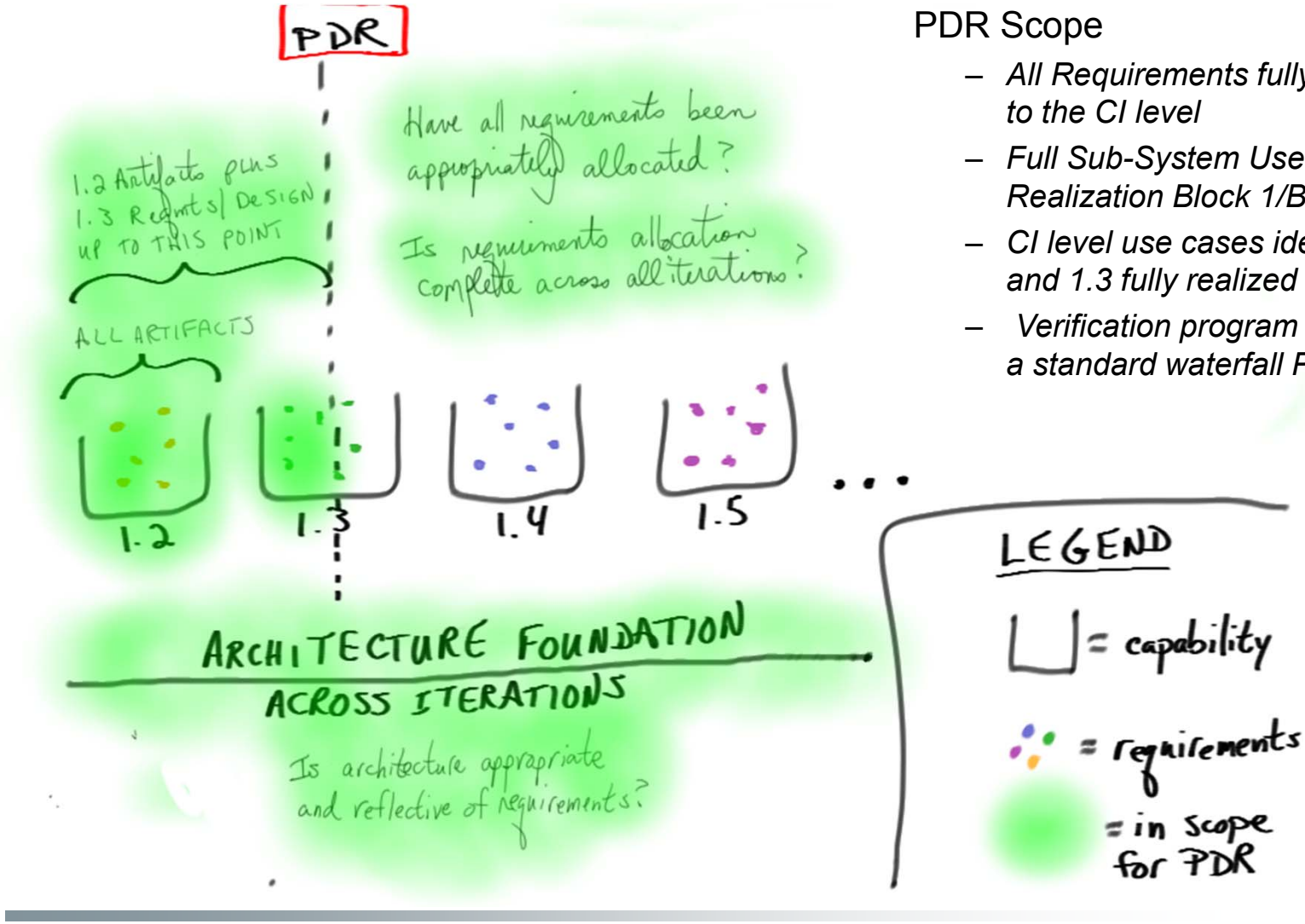
Rigorous, Methodical Approach with Clearly Defined Success Criteria

Design Review Issue Notices

- Issue Notices exist in two flavors
  - *Individual finding that causes failure of one or more 1521B criteria*
  - *Systemic issue that collectively causes failure of one or more 1521B criteria*
- Initiated by Technical POC and validated by Design Review Board before submission to Contractor
- All Issue Notices are supported by one or more findings in the Integrated Comment Review Matrix (CRM)
- Each Issue Notice requires joint agreement to resolution plan
- Closure upon completion of resolution plan and successful review by the GPG technical team of redelivered artifacts
  - *Risk mitigation assessments at each step in resolution plan*

- Developing Resolution Plans
  - Define Issue
  - Scope Issue
    - Effort to fix
    - How widespread
  - Assess Design Impact
    - Critical to current iteration
    - Rework risk
    - Cumulative future risk
  - Document Plan
    - Master CDRIN file

Iterations are at Different Stages for PDR



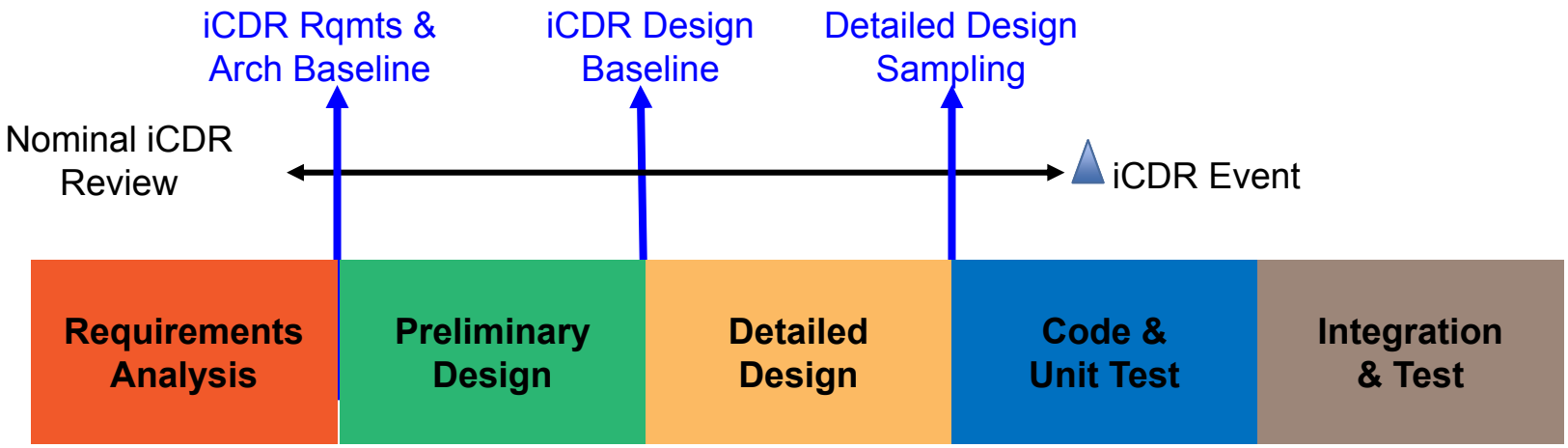
PDR Scope

- *All Requirements fully decomposed to the CI level*
- *Full Sub-System Use Case Realization Block 1/Block2*
- *CI level use cases identified. IT 1.2 and 1.3 fully realized*
- *Verification program to the level of a standard waterfall PDR*

Iterative Development & Classic CDR

- CDR Objectives:
  - *Verify detailed design meets requirements*
  - *Timely opportunity for course correction*
- Issue:
  - *Nearly all code will be implemented or at preliminary design level at any point in iteration development lifecycle*
- OCX Approach:
  - *Detailed design verified thru CDR campaign comprised of focused events*
  - *iCDR – Iteration CDRs with focus on software design*
  - *HCDR – Hardware CDR for developed hardware and adequacy of all HW and SW COTS*
  - *Verification and Transition CDR*
- Advantages:
  - *Satisfies all CDR objectives*
  - *Mitigates schedule risk by reducing distraction to ongoing software development*
  - *Each event scheduled for maximum effectiveness to related program events*

iCDR Campaign Plan



- Verify design maturity for low risk implementation of software planned for the iteration as defined in the Master Software Build Plan
- Assessment based on design artifacts produced to support development
  - *Minimal impact to developers*
- iCDR Event
  - *Planned as a summary of issues and their closure plans*
  - *Compliance to tailored MIL-STD-1521B criteria*
  - *Identification of residual program risk*

MIL-STD-1521B/T Criteria Distribution to CDR Events

CDR Event	# of 1521B Criterion
Software Iteration CDR (iCDR)	54
Hardware and COTS CDR	49
Verification CDR	31
Transition CDR	32

OCX Acquisition – Lessons Learned (so far)

- Program insight and oversight requires detailed planning
- "Traditional" oversight models must be tailored and adapted
- Iterative approach is useful for development of large software systems
  - *Need mature processes*
  - *Control relationship and sequence of iterations*
  - *Maintain rigorous Systems Engineering perspective*
  - *Test program needs to maintain "an Enterprise" perspective*
  - *Acknowledge and address "schedule-driven" program tendencies*
- Place on contracts what is needed & enforce contractual language
- Experienced Government technical team with practical knowledge is critical to success

Maintain discipline throughout