



# Software Development under DO-178B

John Joseph Chilenski  
Associate Technical Fellow  
Airborne Software Engineering  
Boeing Commercial Airplanes

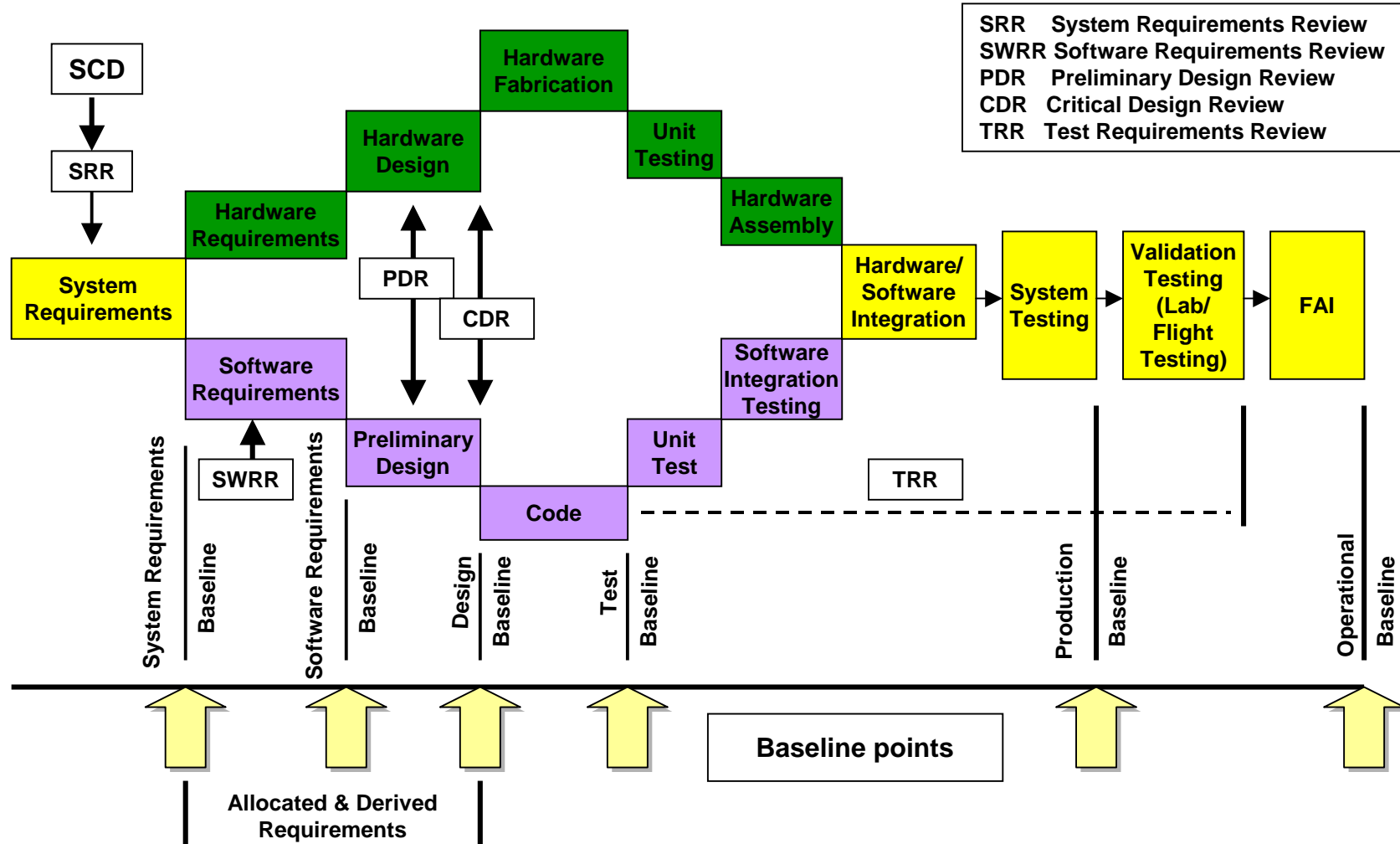
January 2002

# Agenda

- Development Process Model
- Software Lifecycle Processes
- Software Verification Process Framework
- Traceability Requirements
  - Coverage Analysis
- COTS?
- Concluding Remarks

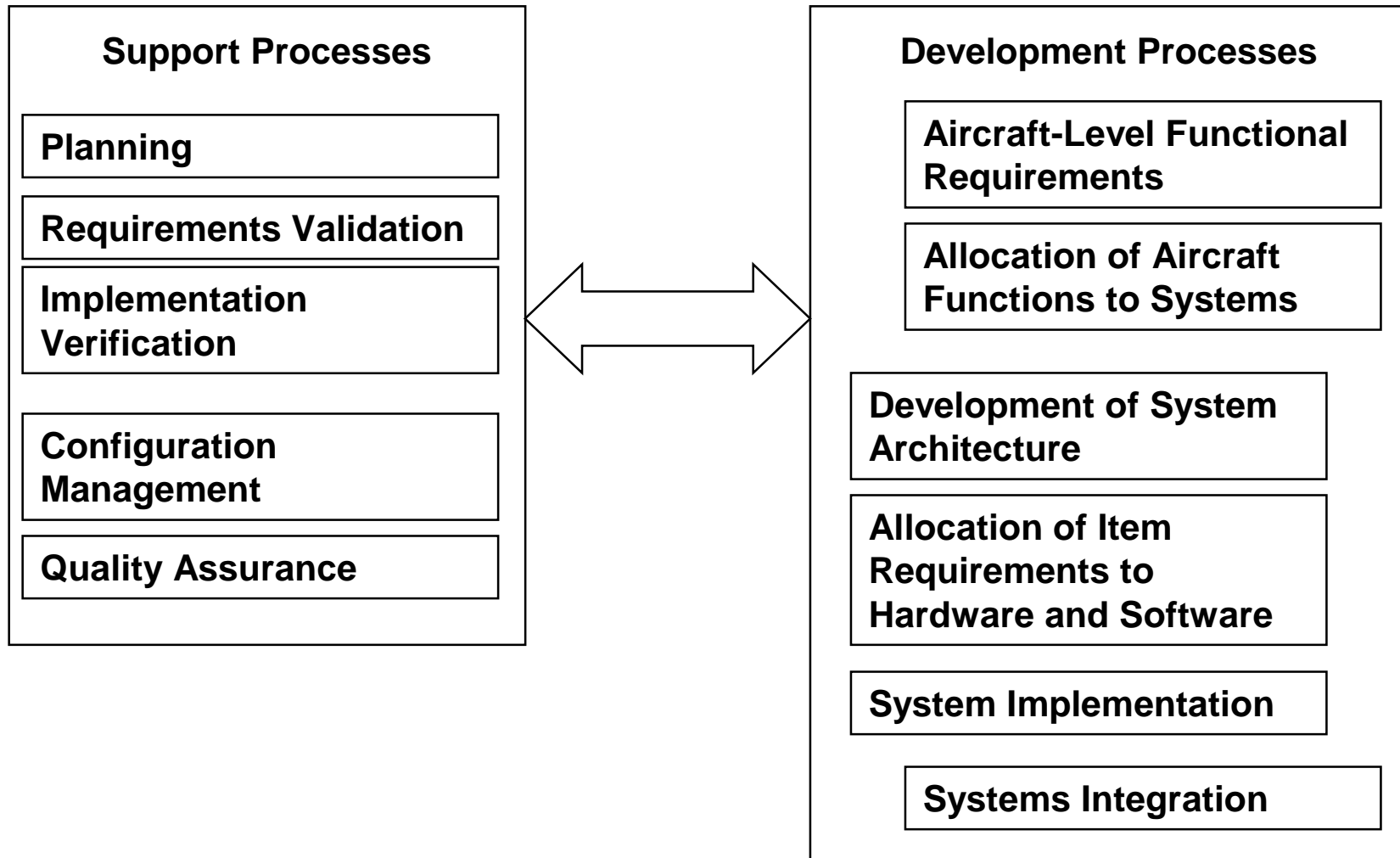
# Development Process Model

## (1 of 4)



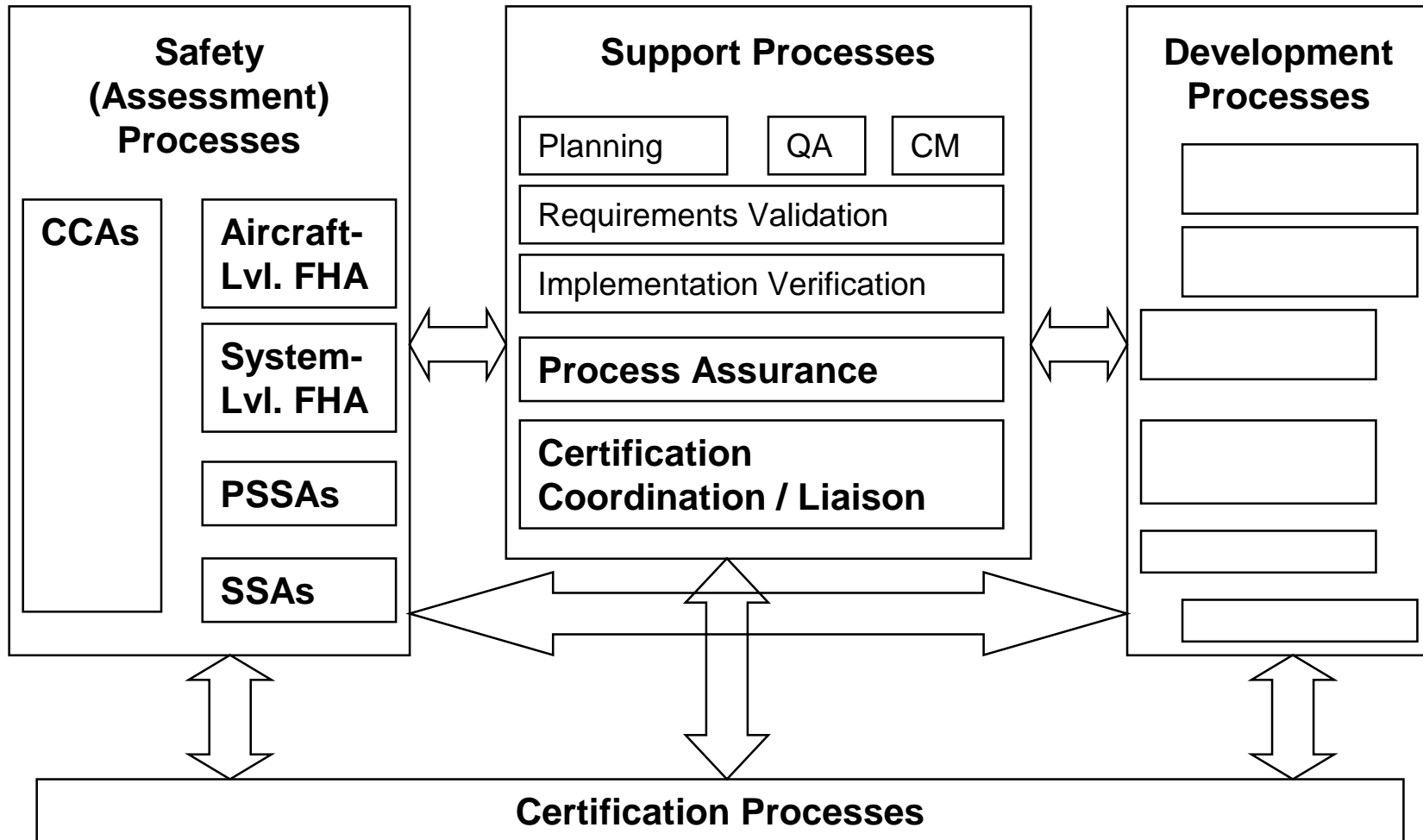
# Development Process Model

(2 of 4)



# Development Process Model

(3 of 4)



# Development Process Model

(4 of 4)

- In addition to Interaction and Feedback between Processes, some Processes impose Constraints upon other Processes
  - The Certification Process imposes constraints on all the other Processes
    - Especially the Verification Process
  - The Verification Process imposes constraints on most of the other Processes
    - Especially the Development Processes
    - If it can't be Verified, it can't be Used
      - Results in only a subset of Architectures, Designs and Implementations being Acceptable
- Traceability is very Extensive, and very Important, in the Safety Critical arena
  - Especially Traceability to Verification and the Safety Case

# Software Life Cycle Processes

Software Planning Process

Software Development Processes

Software  
Requirements  
Process

Software  
Design  
Process

Software  
Coding  
Process

Software  
Integration  
Process

Software Integral Processes

Software Verification Process

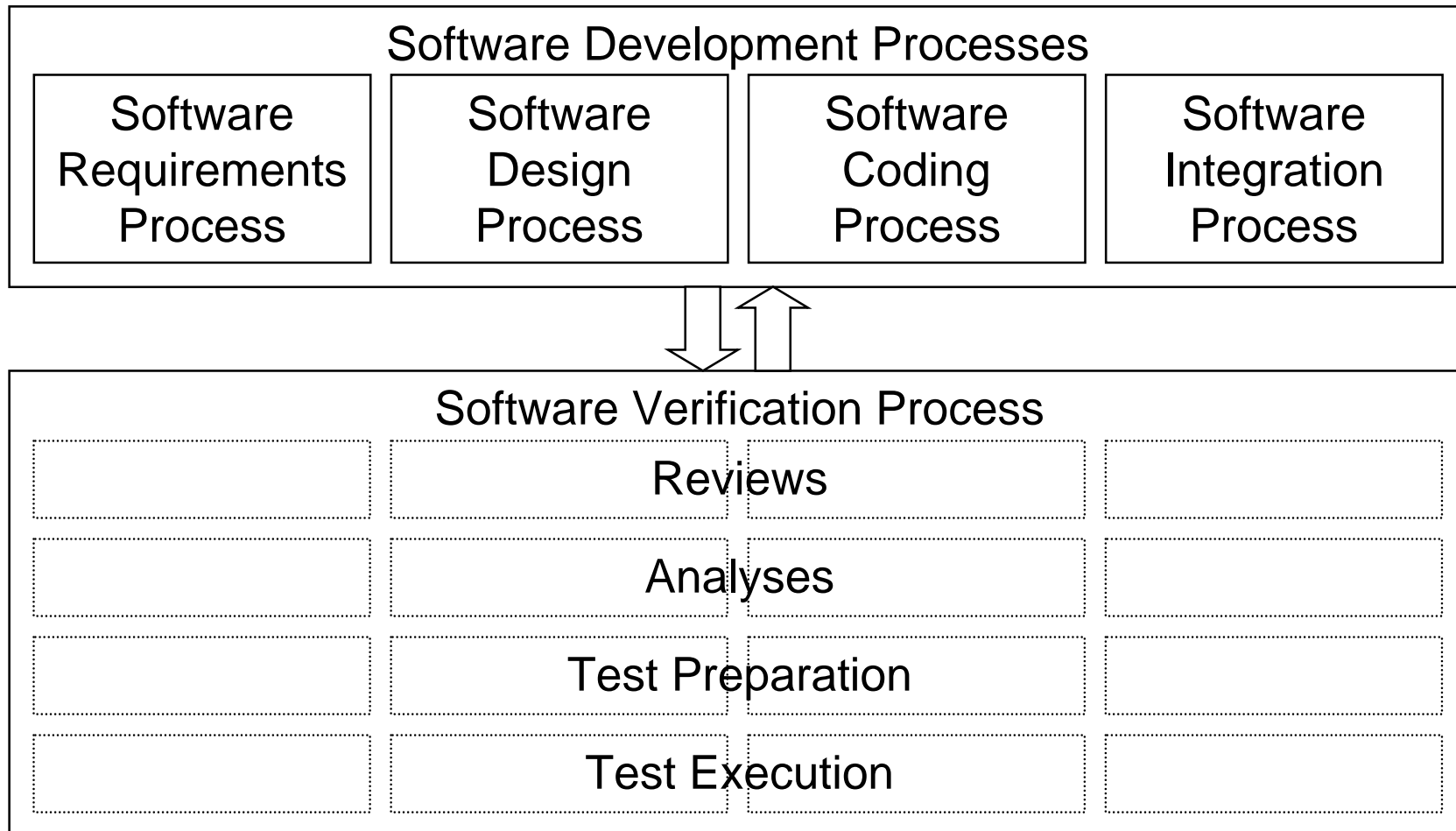
Software Configuration Management Process

Software Quality Assurance Process

Certification Liaison Process

# Software Verification Process Framework

(1 of 4)



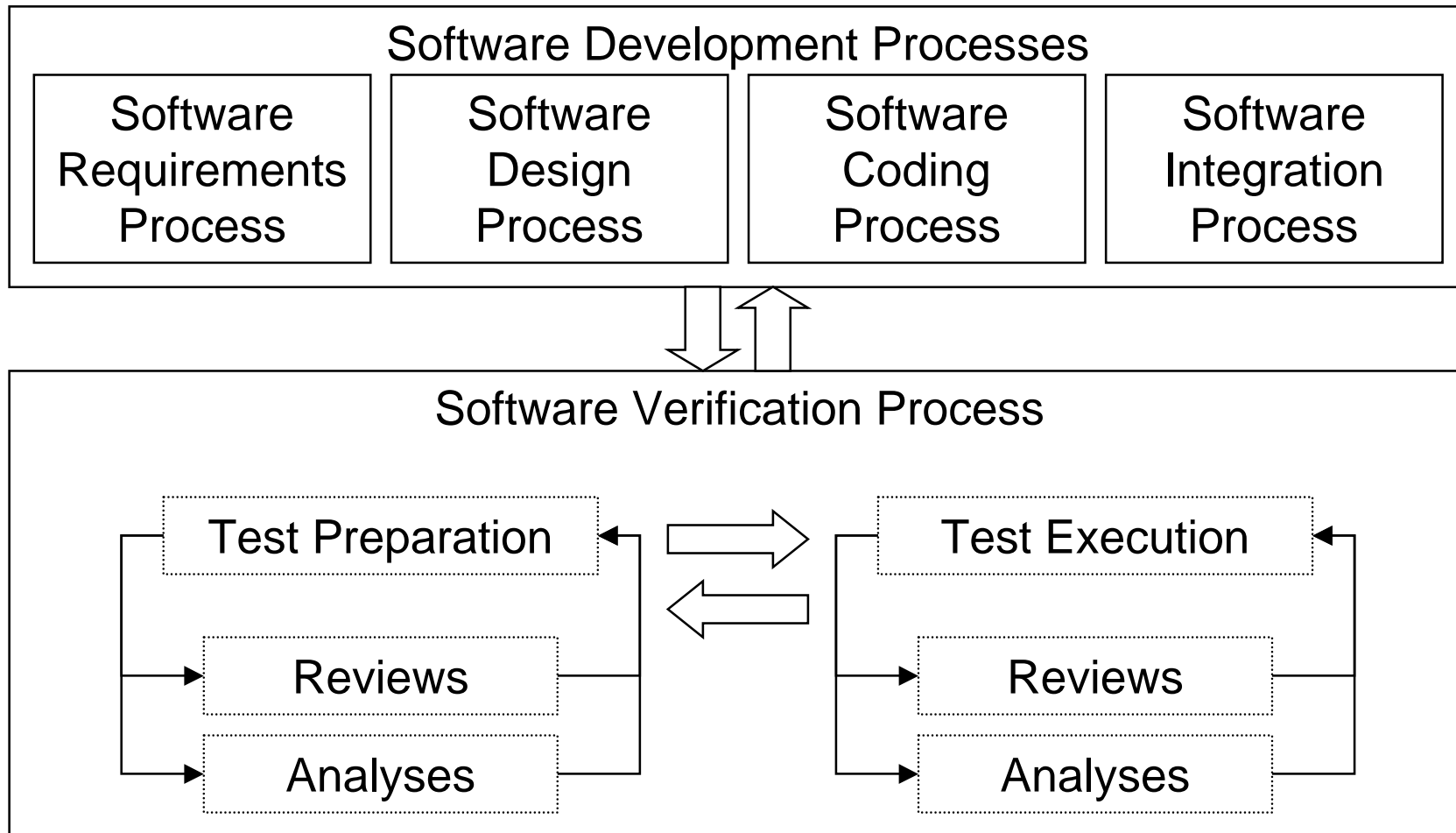
# Software Verification Process Framework

## (2 of 4)

- Reviews
  - Provide a qualitative assessment of correctness
- Analyses
  - Provide repeatable evidence of correctness
- Tests
  - Demonstrate that the software satisfies its high and low level requirements
  - Provide a high degree of confidence that errors which could lead to unacceptable failure conditions have been removed
  - DO-178B acknowledges that test preparation can be as effective as test execution

# Software Verification Process Framework

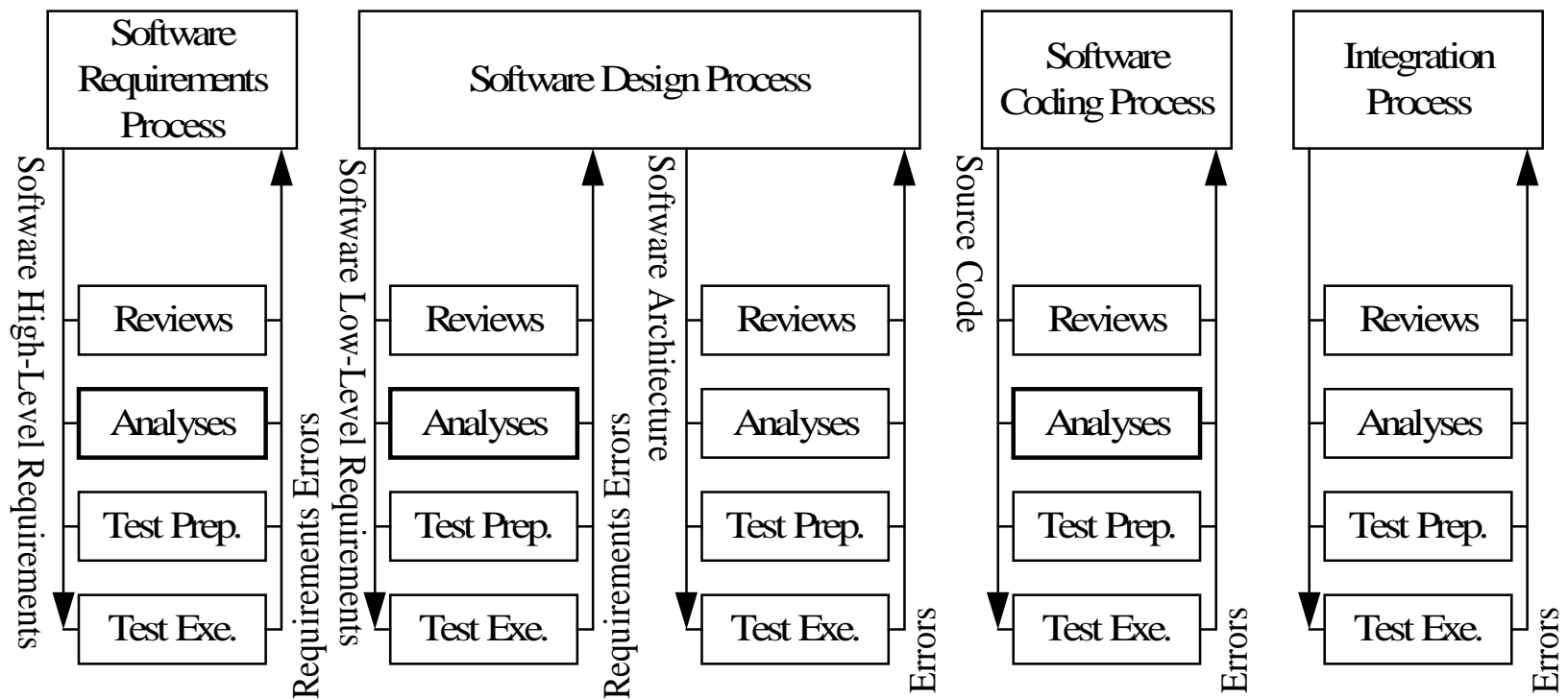
(3 of 4)



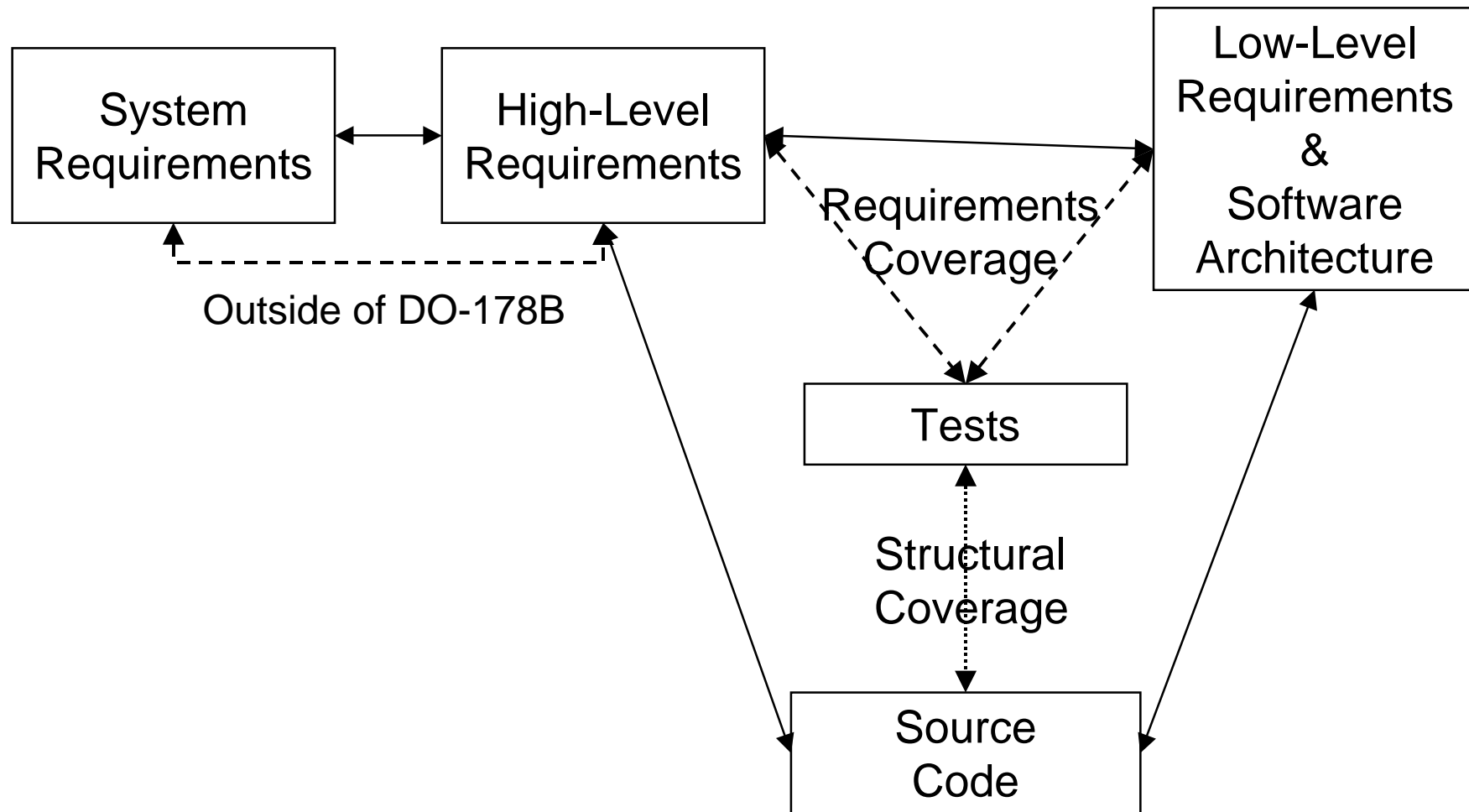
# Software Verification Process Framework

(4 of 4)

- Extensive Verification required in the production of Safety-Critical Software



# Traceability Requirements



# Traceability - Coverage Analysis

## (1 of 2)

- Well defined way to (objectively) assess (i.e., measure) the quality of your System and Software Requirements (e.g., completeness, consistency, testability)
- Measure how well the Implementation reflects the Requirements
  - Requirements Coverage: Requirements  $\Leftrightarrow$  Tests
  - Structural Coverage: Tests  $\Leftrightarrow$  Source Code
- Objective completion criteria for the (System and Software) Testing Process
  - Determine how much Testing is enough
    - All Requirements (system and software) tested and verified
    - All Source Code exercised and verified
    - All Object Code exercised and verified (Level A only)

# Traceability - Coverage Analysis

## (2 of 2)

- Objective assessment of the adequacy of the (overall) Verification Process
  - All Requirements (system and software) implemented (satisfied) in (by) the resulting System
  - All Source Code implements the (software) Requirements
  - All Object Code implements the Source Code (Level A only)
  - Implementation of Requirements (system and software) is verified
- Helps with the identification of
  - Dead (“untest-able”) Code
  - Deactivated Code
  - Unspecified (“unintended”) Function
  - Unverified Function

# COTS?

(1 of 2)

- Acceptance of software by the certification authorities is done on a system by system basis
  - No universal acceptance
    - Even between different versions of the same system
- All software must satisfy the objectives of DO-178B
  - This presentation has concentrated on airborne software
    - Development tools are treated the same
    - Verification tools have different requirements
- All airborne software “aspects” (documentation, code, tests, etc.) must ultimately trace to the system requirements

# COTS?

(2 of 2)

- Strictly speaking then, there will be no “pure” COTS for commercial airborne software/systems
  - Instead, we can have “developmental” COTS
- COTS vendor will need to work with the system developer to ensure acceptance of the component
  - Follow an (DO-178B) acceptable development process
    - Includes change control and configuration management
  - Produce all the required documentation/evidence
    - With “hooks” for the developer to link into
  - Provide all materials for review and approval
    - Including possible submittal to the FAA and/or DER

# Concluding Remarks

- Rigorous process for the development of commercial airborne software
  - No *separate* product assurance for software
    - Evidence of product conformance required
- Lifecycle data required
- Conformance to standards substantiation required
- COTS is possible
  - But different from what is generally done today