

---

# Using MathWorks Tools to Develop DO-178B Certified Code

# Tools Used for Development of DO-178B Certified Code

---

- The following tools from The MathWorks have been used by Honeywell to develop code for the PRIMUS 2000 and PRIMUS EPIC<sup>®</sup> systems:
  - MATLAB<sup>®</sup> a high-performance language for technical computing
  - Simulink<sup>®</sup> - a tool for modeling, simulating, and analyzing dynamic systems
  - Stateflow<sup>®</sup> - a graphical design and development tool for control and supervisory logic
  - Real Time Workshop<sup>®</sup> Embedded Coder - an automatic code generator that generates embedded C Code from Simulink and Stateflow models

# Aircraft & Systems Certified Using The MathWorks Tools

---

- Aircraft:
  - Bombardier Global Express
  - Gulfstream IV & V
  - Dassault Falcon 900 & 2000
  - Cessna Sovereign
  - Augusta Bell 139 Helicopter
  - Embraer 170
- Systems:
  - Automatic Flight Controls
  - Monitor Warning System
  - Fly-By-Wire Control System
- Software levels - A, B, C, D

# Qualified Verification Tools Developed by Honeywell to Compliment MathWorks Products

---

- **Model Compliance Checker**
  - Verifies safe subset of blocks used
  - Verifies model meets design standards
  - Partially eliminates software requirements reviews
- **Software Test Case Generator**
  - Verifies executable object code complies with model
  - Automates development of software test cases
  - Eliminates software test case reviews
- **Source Code Verifier (Patent Pending)**
  - Verifies source code complies with model
  - Eliminates code reviews

# Documentation Developed for Tool Qualification

---

- Tool Qualification Plan
- Tool Operational Requirements
- Tool User Guide
- Tool Design Document
- Requirements Based Test Cases
- Tool Partitioning Analysis
  - Required because some of the unqualified parts of the tool are development tools (e.g. RTW Embedded Coder)
- Tool Accomplishment Summary

# Benefits of Using COTS Tools for Model Based Development

---

- High quality code
  - Over 1 million lines of code have been certified just in the last year.
  - One code generator option error was found (and corrected), although the generated code actually performed correctly and passed testing with 100% MCDC coverage.
  - No compiler errors have been found when using an unqualified COTS compiler with a limited subset of model-based C code.
- High quality design
  - Defect leakage rates at integration are reduced by at least one order of magnitude.
  - Designs are proven prior to code generation.
  - Model based testing provides more thorough and rigorous method of validating and verifying system design and software requirements.

# Benefits of Using COTS Tools for Model Based Development

---

- Model based development puts the emphasis on improving quality of the design
  - Error rates for design errors are still relatively high compared to implementation errors. Model based development emphasizes design error prevention and detection.
  - Error rates for code and compilation are relatively low, zero defects is achievable when safe subsets are used for COTS code generators and compilers.
- Achieves two important industry goals
  - Reduces cost
  - Improves safety

# Qualification Issues - Reuse of Tools and Artifacts

---

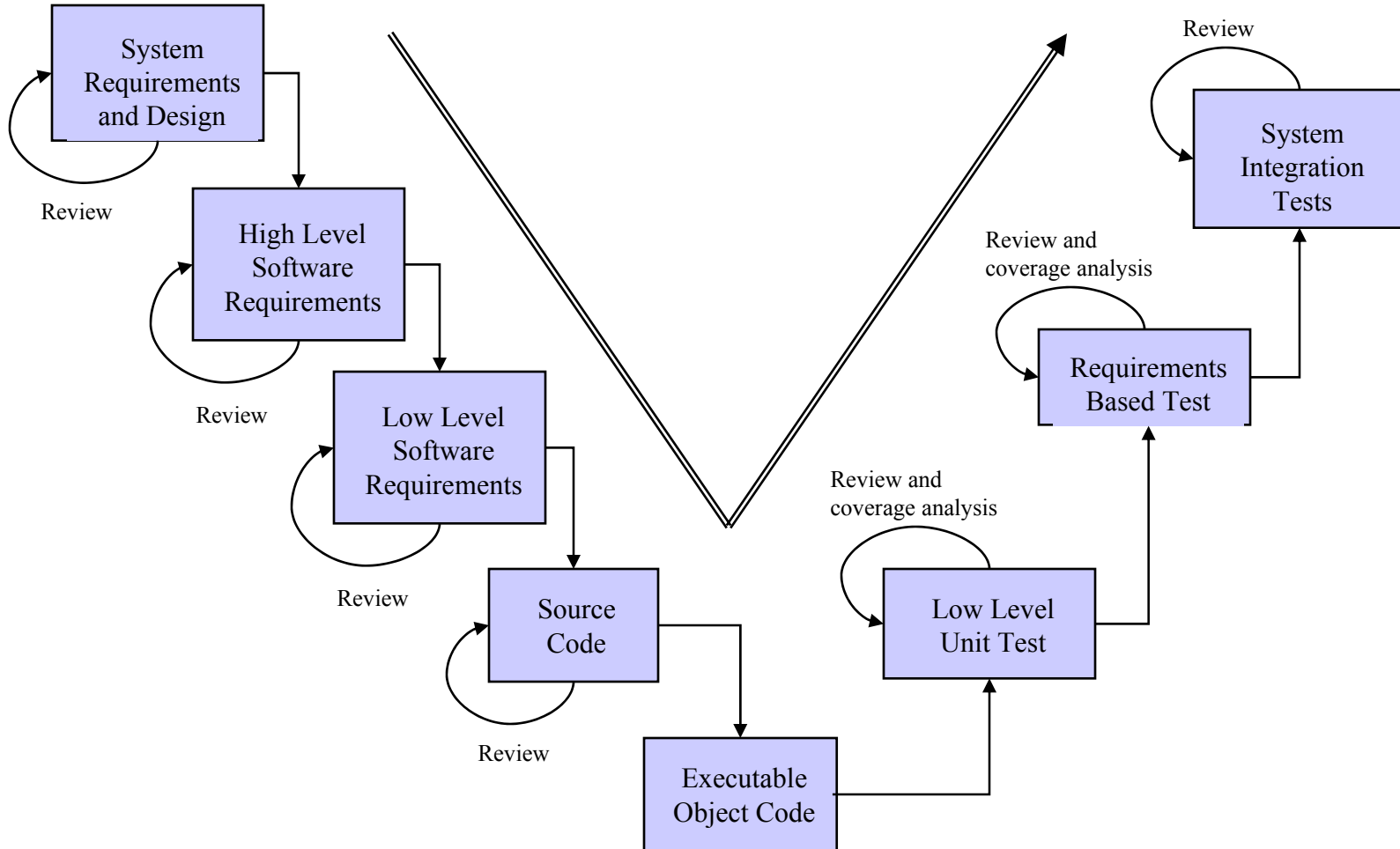
- Positives:
  - Tool qualification data has been successfully reused across multiple aircraft programs and multiple software applications on each program.
- Negatives:
  - Tool ends up being audited multiple times by different certification agencies and individual auditors.
  - Each application and program using the tool must rejustify the tool qualification.

# Model Based Development Issues - DO-178B Mapping

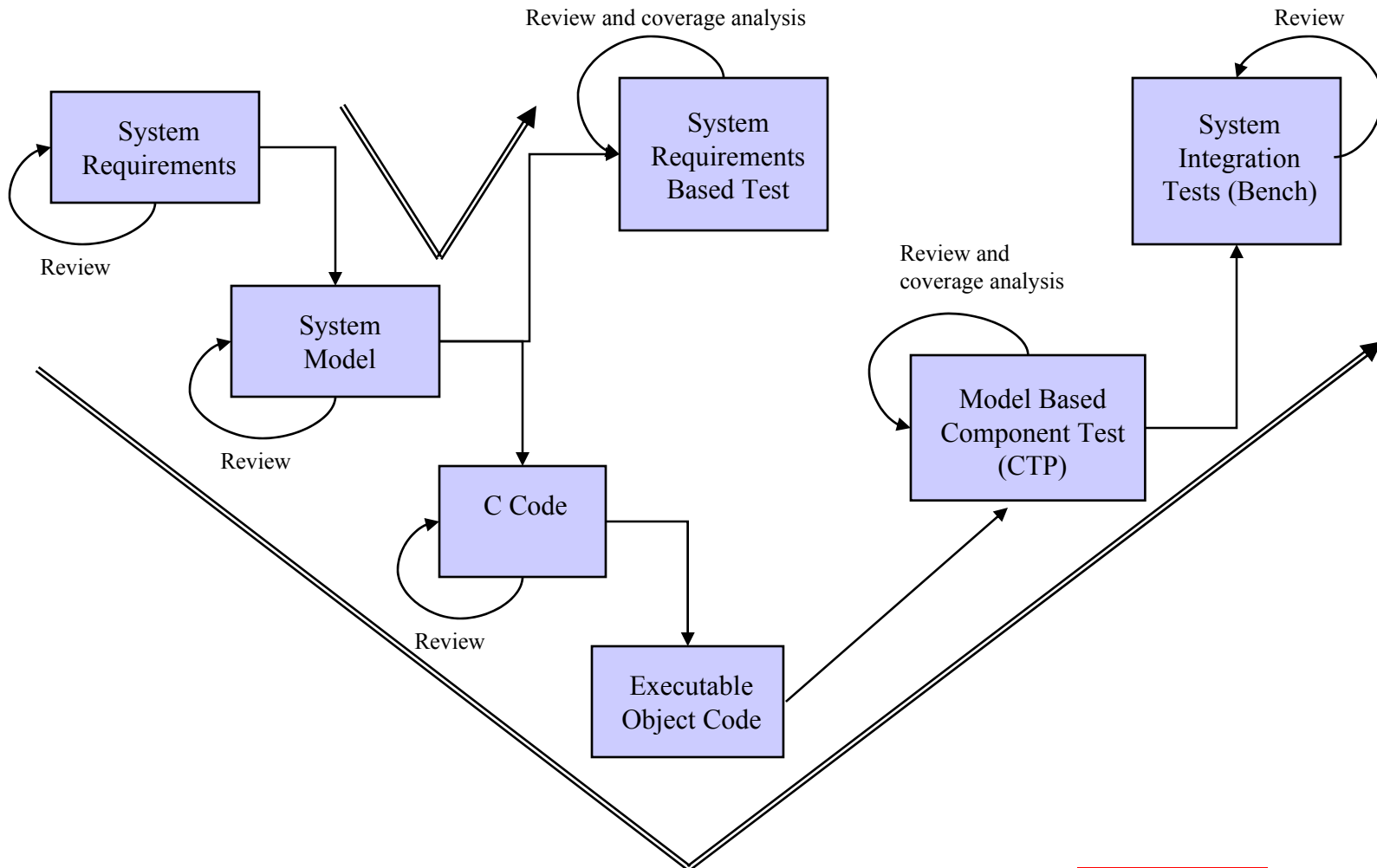
---

- Positives:
  - Model based development eliminates steps, which inherently reduces error injection points in the development process resulting in higher quality.
  - Model based development puts emphasis on detecting design errors (high leakage rate) rather than implementation errors (low leakage rate).
- Negatives:
  - DO-178B is obsolete with respect to model based development processes
  - SOI audits become difficult when development steps are merged
  - Can end up adding non-value added tasks to the development process just to fit into DO-178B

# Classical DO-178B Software Development Process



# Model Based DO-178B Software Development Process



# Model Based Development Issues - DO-178B Mapping

---

- Model based development merges the following development steps:
  - System Design
  - High-Level Software Requirements
  - Low-Level Software Requirements
- Some people view the graphical model as a higher level software language, the code generator is simply another compiler stage
- DO-178B and ARP-4754 do not address this possibility
  - Currently it is difficult to get consensus about mapping to DO-178B
  - Certification authorities need to start addressing this issue

# Tool Qualification Issues - Development versus Verification

---

- Internal trade studies have shown that the cost of development tool qualification is at least 20x the cost of verification tool development.
  - Use of qualified verification tools results in savings on first program where it is introduced
  - Use of qualified development tools takes several programs to make up the cost
- COTS tools, and the platforms they run on, are updated at a faster rate than aircraft development life cycles (18 to 24 months versus 3 to 5 years).
- Qualified development tools are typically either very simple tools or end up way behind the technology curve.

## Tool Qualification Issues - Service History

---

- There is currently no good definition or guidance for acceptable service history.
- Typically, by the time service history is achieved, the tool has been updated or modified in some way.
  - The tool that generated 1 million plus lines of defect-free code last year is being updated to a new version this year.
  - Between the certification of the Global Express in 1998 and the Gulfstream V in 2002, MATLAB was updated three times, making it virtually impossible to use service history.

## Tool & Technology Issues - What Industry Needs

---

- Set up methods to audit the tool independent of the programs and applications using it
- Update guidance materials (DO-178B & ARP-4754) to properly cover model based development processes
- More streamlined method to qualify development tools and to keep them current as technology advances
  - Why are there two extremes? (Level A versus Service History)
  - Can we find a more reasonable middle ground?
- Better guidance on how to apply service history, and to address what has to be done for incremental tool changes