Presented by

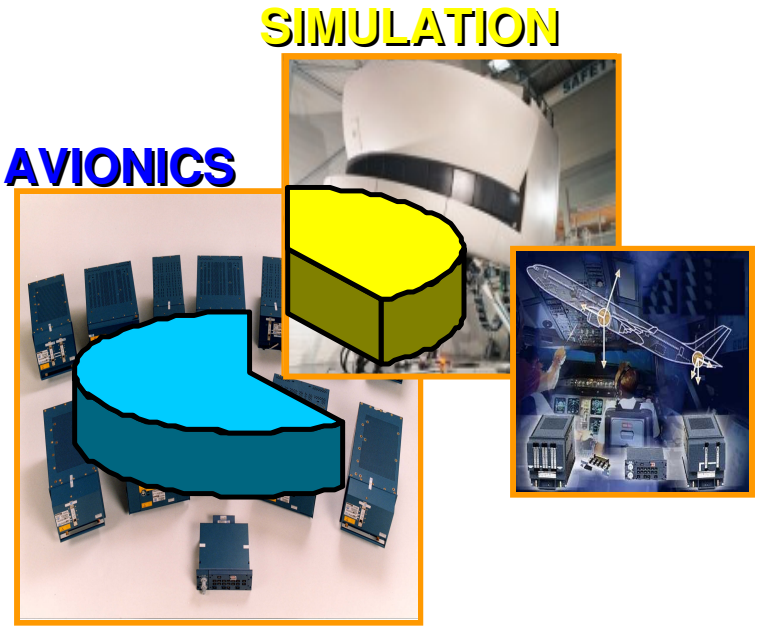
Famantanantsoa Randimbivololona

Avionics and Simulation Products
AIRBUS FRANCE

Abstract interpretation in avionics engineering …

AIRBUS

Who we are ?



Avionics and Simulation Products

Center of Competences for :

- Electronics and on board Software in real time applications
- Avionics and Simulation

Business Center

- Developing and selling products

AIRBUS

Our avionics products

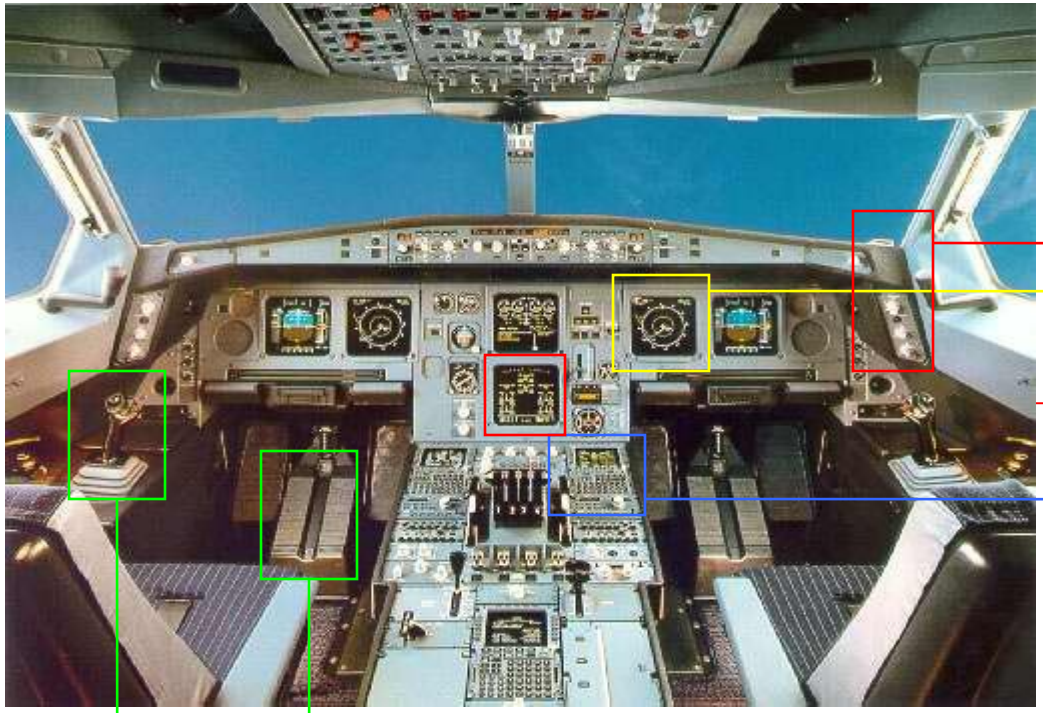
Products / Equipment's sets

DOMAINS

- Flight control
- Warnings
- Maintenance
- Communication

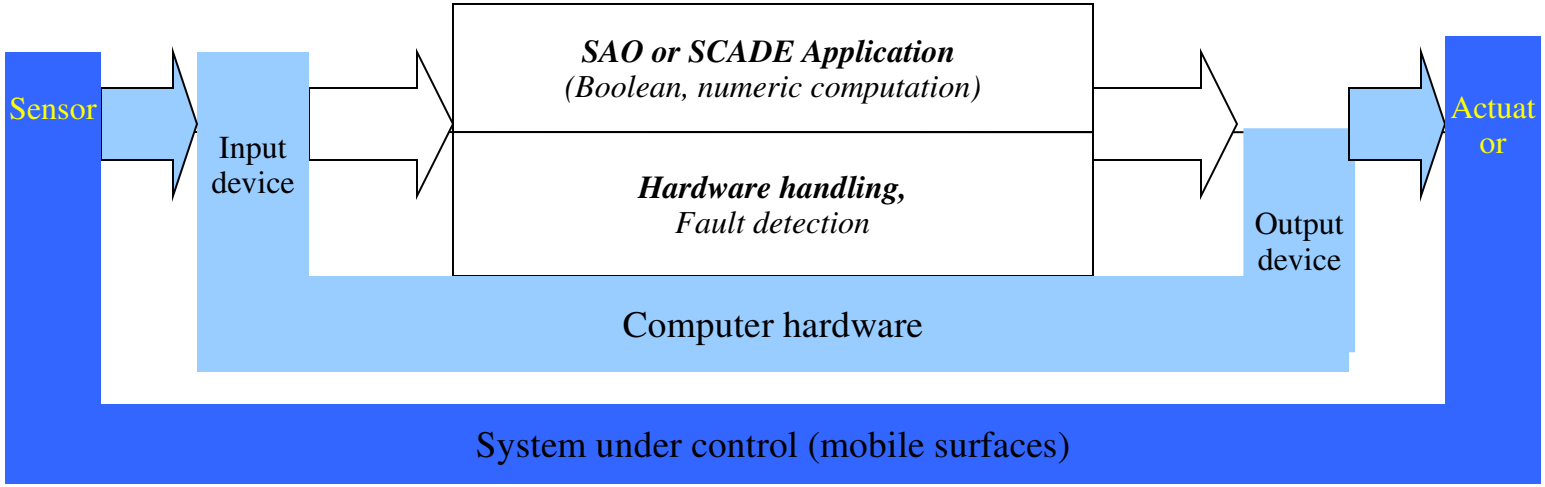
Elements on avionics

Architecture overview for the A330/340

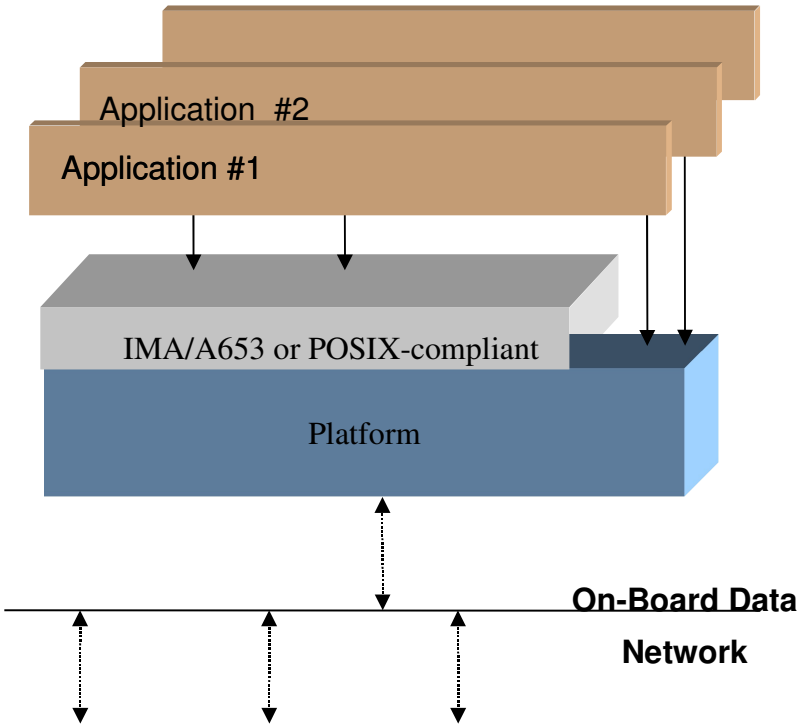


From safety critical ...

- Electrical Flight Control
- Safety level: critical (A)
- Synchronous mono-application
- Hard realtime constraints
- From 150 kloc to 1000 kloc



… To safety essential



- Air Traffic Control communication
- Flight Warning function
- Safety level: C/B (DO-178B)
- Asynchronous multitask application
- « Soft » realtime constraint (communication timeout)
- Large application: upto 4000kloc
- Run on top of a multiappli OS platform:
  - Integrated Modular Avionics A653
  - embedded POSIX

Today's engineering ...

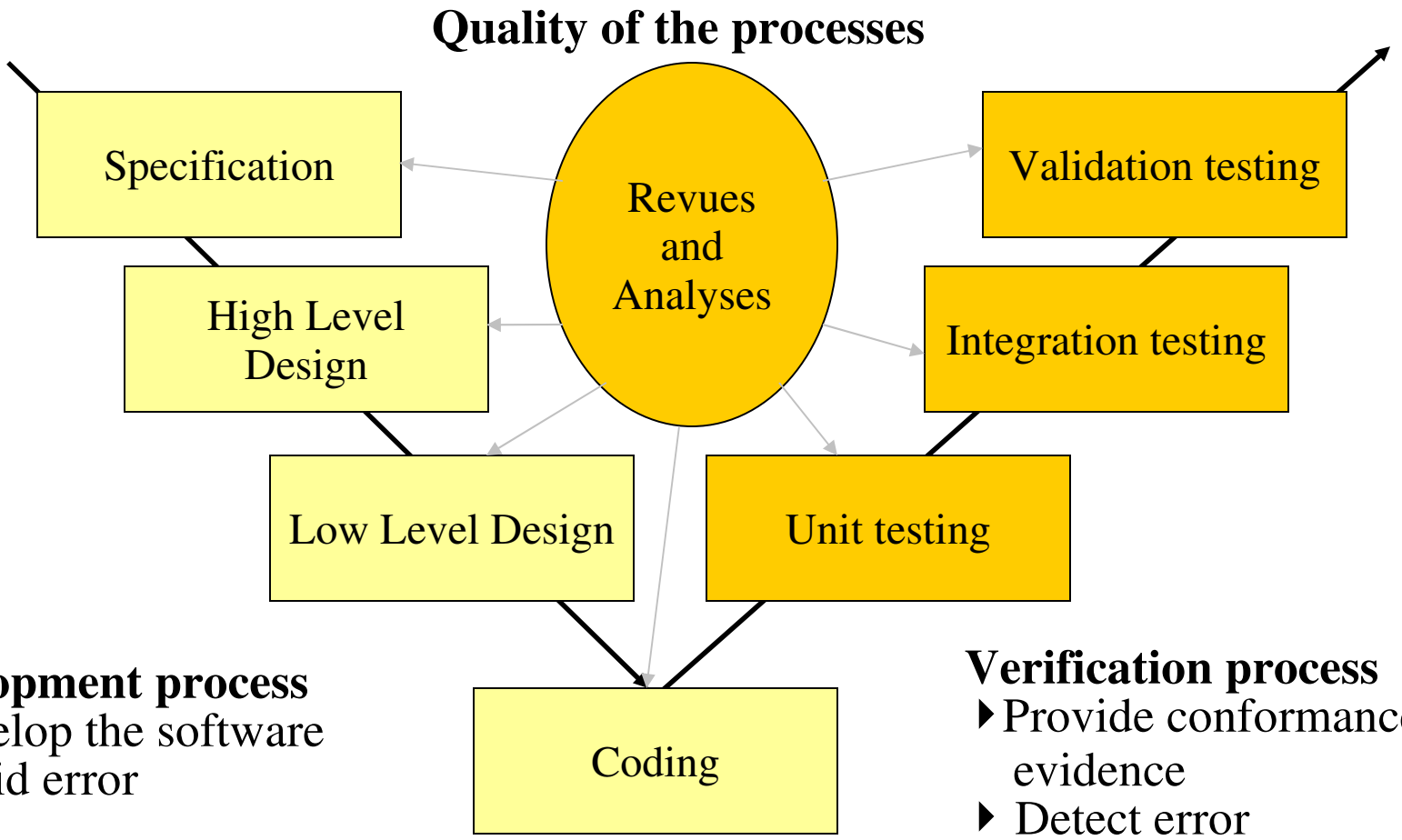
DO-178B/ED-12B certification compliance

- Guidance for satisfying airworthiness requirements
- Define software life cycle, processes and assurance criteria
- Level of assurance and completion criteria depend on software level
- Industry-accepted techniques and methods
- Otherwise equivalence demonstration for alternative means

Regular revision of DO-178_/ED-12_

- Next coming will be DO-178C

… Software life cycle: the “V” model



Development process
- Develop the software
- Avoid error

Verification process
- Provide conformance evidence
- Detect error

Motivation for static analysis technique

- Limits of tests:
  - Costs: test means and tools, test software, coverage completion
  - Intrinsic difficulties on: robustness checks, determination of computer-resources upper-bounds, computation safety => suboptimal architecture, resources-consuming fault-tolerance mechanisms

- The problems are increasing
  - Trend towards software-intensive systems: more functions implemented in software, more sophisticated functions, new functions
  - Evolution of underlying hardware technology: integration level, modern processor architecture, floating-point operators

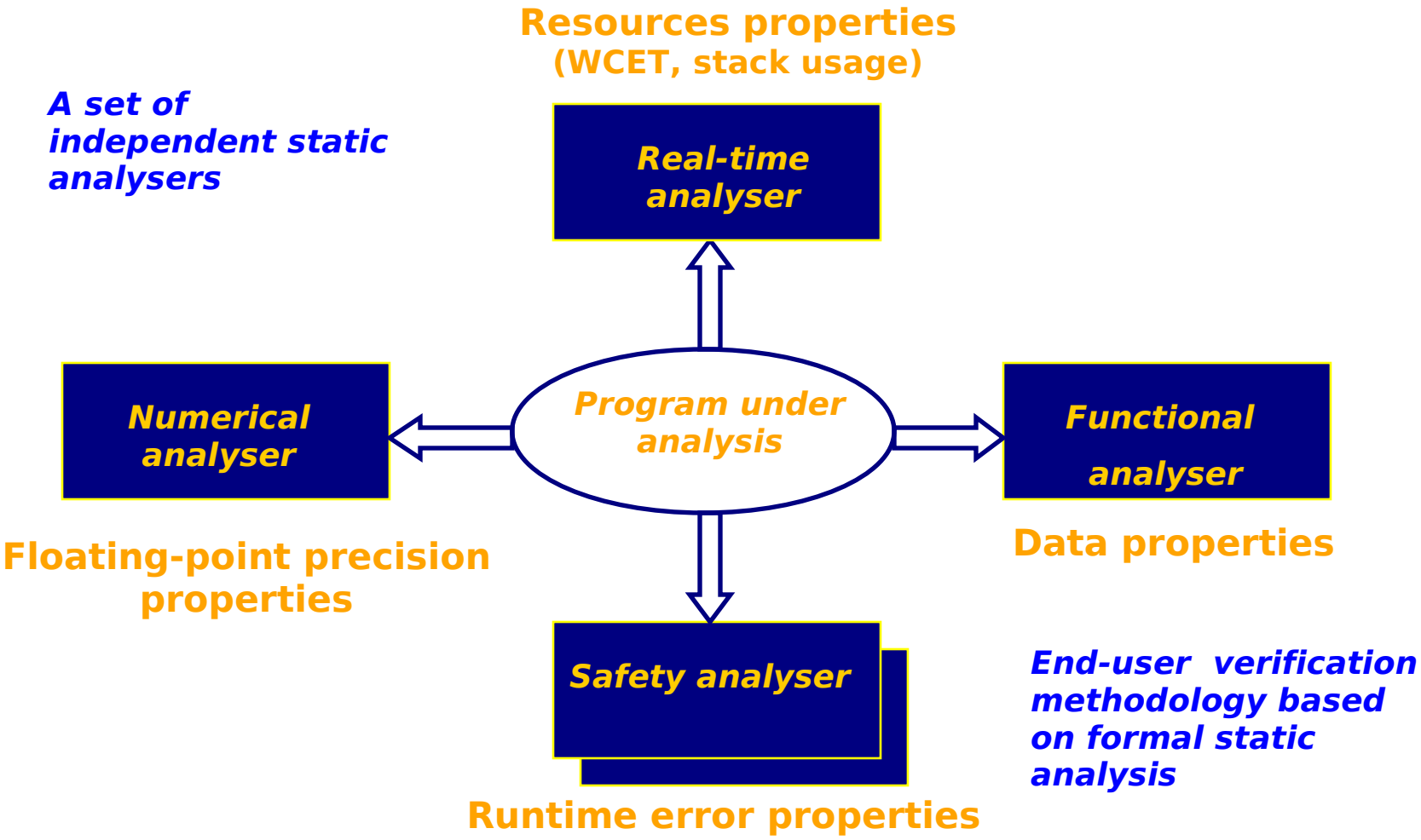
… Motivation for static analysis technique

- Introduction of static analysis
  - Main idea to avoid limits induced by test execution:
    - Dynamic properties are « present » in the code of the program
  - **Analyse the source code - at compilation-time - to check execution-time properties**
    - Exhaustive (notion of proof => maximum coverage)
    - Highly automatized
  - **Well-founded on scientific theory**
    - Abstract interpretation

Acceptability criteria for static analysis technique

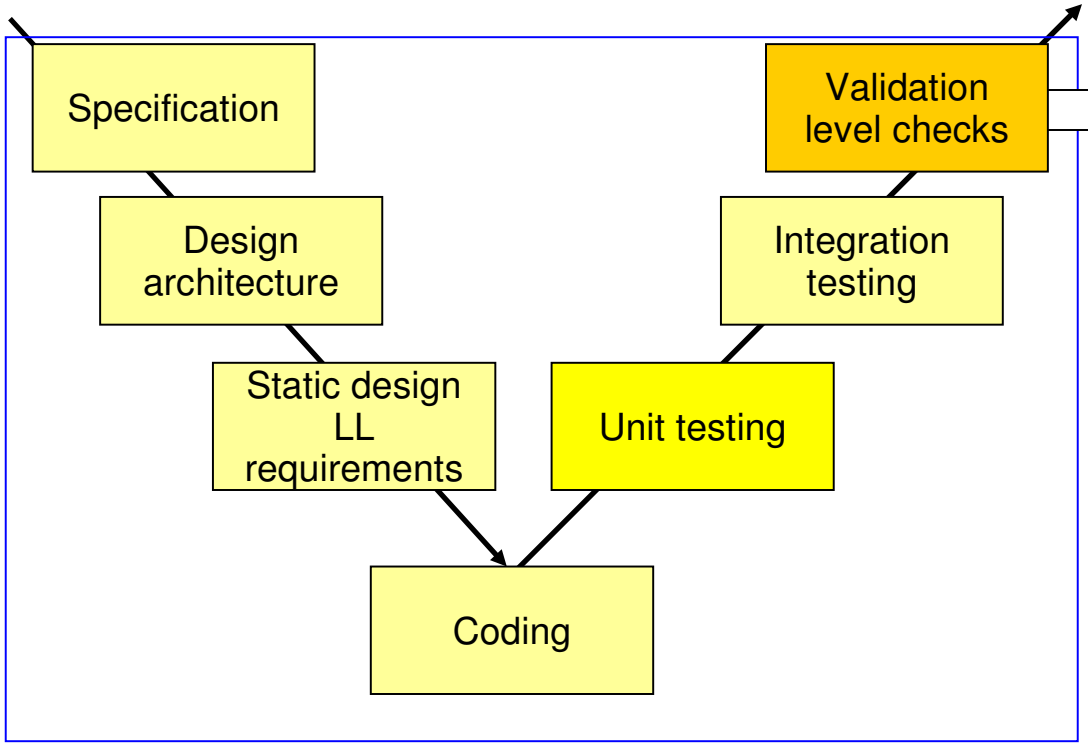
- Ease of learning and use.
  « Standard » avionics software developers must be able to use the tools
- Early payback.
  New process must have better characteristics (productivity, quality-effectiveness)
- Easy integration.
  The use of the tools should not break down the actual verification process and environment
- Ability to cope with real program.
  Analyze unmodified program
    at the source level -> C programming language
    at the binary level(if required) -> X86, PPC
- Scale-up to real size program.
  Synchronous program -> 1000kloc
  Asynchronous program -> 4000kloc

Properties of interest: a first set



Resources properties

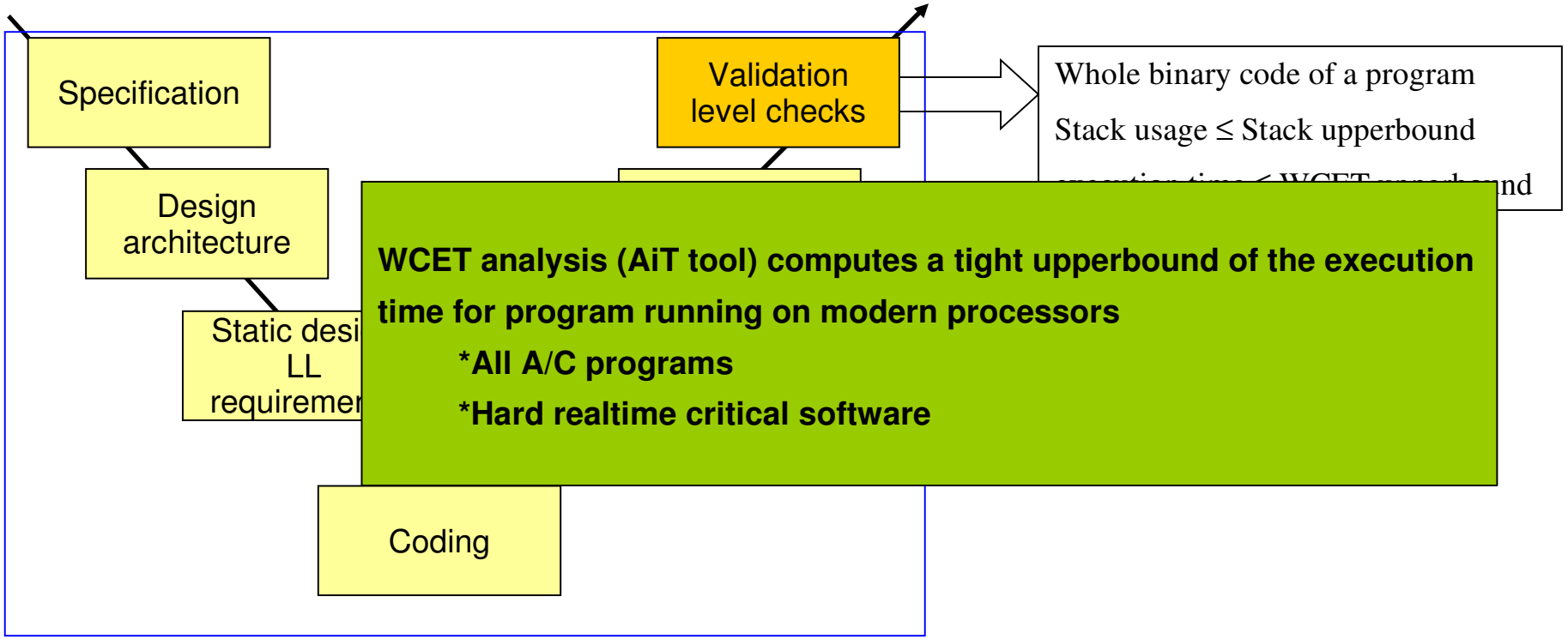
- Abstract interpretation-based WCET and Stack analyzers (ABSINT)
  - Fully part of the software production workbench
  - Qualified (DO-178B) as verification tools
  - Both analyse the binary executable code



Place in the development cycle

Resources properties

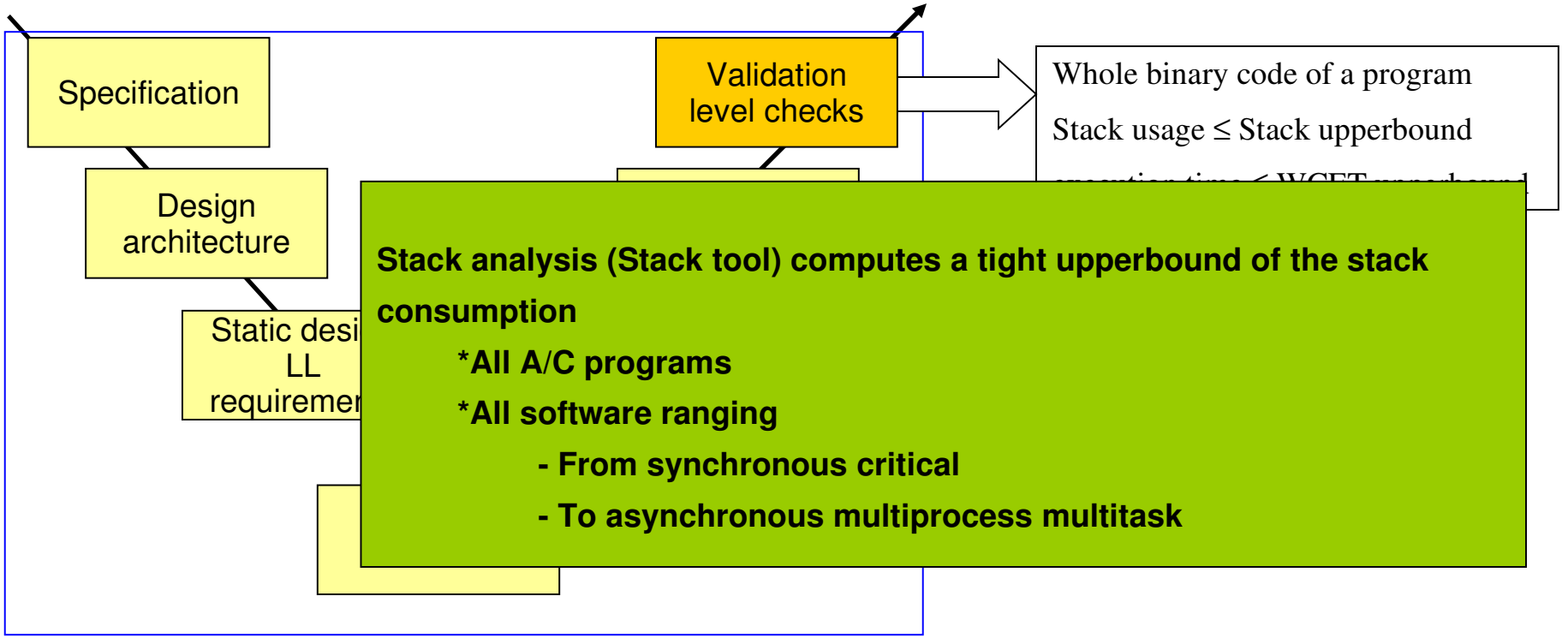
- Abstract interpretation-based WCET and Stack analyzers (ABSINT)
  - Fully part of the software production workbench
  - Qualified (DO-178B) as verification tools
  - Both analyse the binary executable code



Place in the development cycle

Resources properties

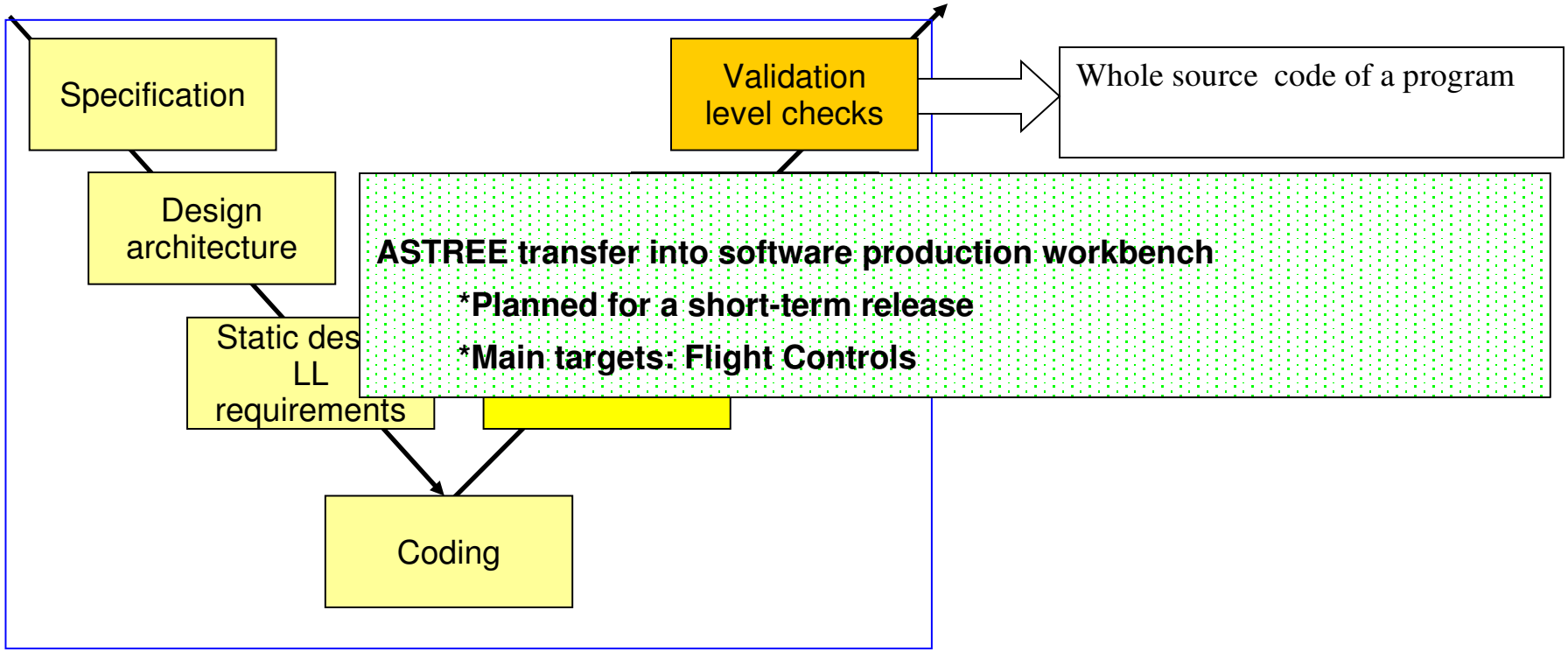
- Abstract interpretation-based WCET and Stack analyzers (ABSINT)
  - Fully part of the software production workbench
  - Qualified (DO-178B) as verification tools
  - Both analyse the binary executable code



Place in the development cycle

Runtime error properties

- ASTREE (ENS Paris) aims at proving the absence of runtime errors for synchronous program
  - Analyzes successfully real, full flight control programs (340, 380)
  - Achieves zero false alarm through parametric analysis
  - Analyses C source code. Early use by Airbus verification specialists



Place in the development cycle

The future

- Emphasis on integration/verification
- Static analysis opens unprecedented perspectives for software engineering
  - Assurance-based on the product vs development process
  - Computation-based engineering vs « human force »
- Developments are required
  - Generalization
    - Classes of programs
    - Languages support
  - Extension
    - Classes of properties
    - Software failure analysis

AIRBUS

AN EADS COMPANY