

FAA National Software Conference

Software Verification



 *Digital Flight*

FAA National Software Conference

Verification Within An RTCA DO-178B Framework

Presented by: Cheryl Dorsey
Digital Flight, Principal
email: dorsey@erols.com

Slide 1



 *Digital Flight*

Verification—Within a DO-178B Framework

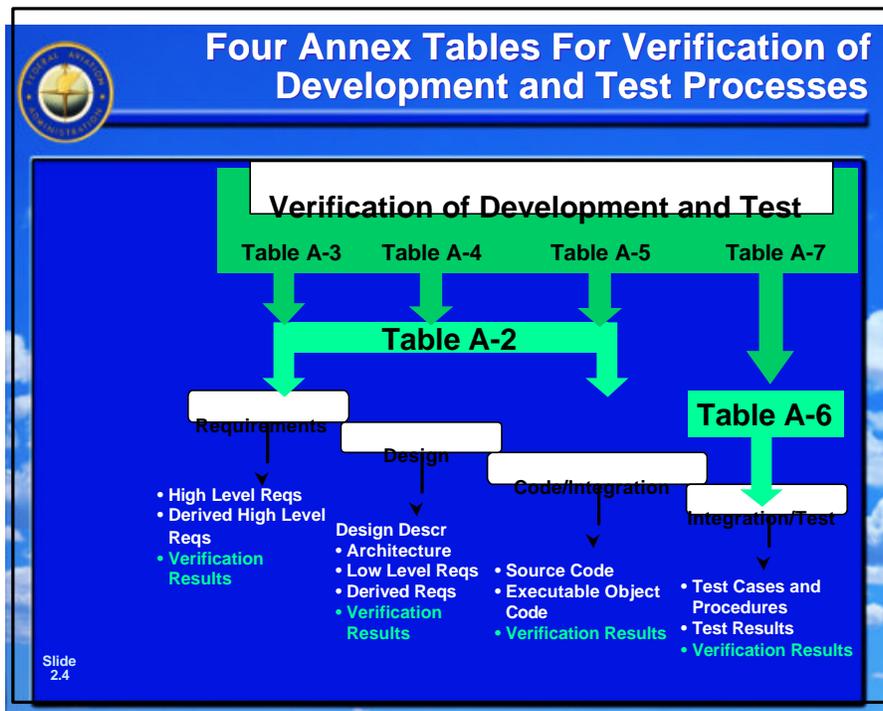
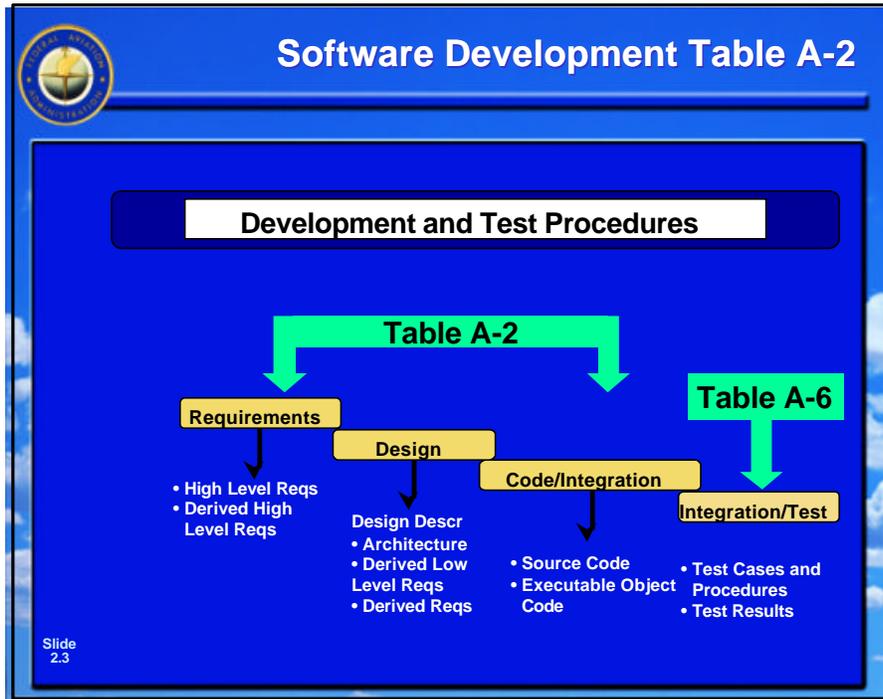
Presentation Objectives:

- Understand the relationship of DO-178B verification tables A-3, A-4, A-5, A-6 to the software development and test tables A-2 and A-6
- Understand the objectives of tables A-2 through A-7
- Setting up for success

Slide 2

FAA National Software Conference

Software Verification



FAA National Software Conference

Software Verification



 *Digital Flight*

Terminology

High level:	A requirement that is traceable to a system requirement states what the software must do, not how it will do it.
Derived:	A requirement that stemmed from a design decision—may not directly trace backwards, but should trace forwards.
Decomposed:	A requirement (system, or high-level) that is satisfied by the combination of two or more lower-level requirements.

Slide 5



 *Digital Flight*

Terminology (Cont'd)

Design Description:	The combination of low-level requirements and architecture. May include a data dictionary.
Low-level Req.:	States how the high-level requirement is met.
Architecture:	Diagram showing how the components of a system are tied together. Usually presented from a control and a data flow viewpoint.

Slide 6

FAA National Software Conference

Software Verification



Developer's Road Map

Planning

Requirements

- SW Requirements Data
 - High Level Requirements
 - Derived High Level Requirements
- High-level requirements
 - Comply with systems requirements
 - Accurate and consistent
 - Compatible with target
 - Verifiable and traceable to system
 - Conform to standards
 - Algorithms accurate

Slide 7



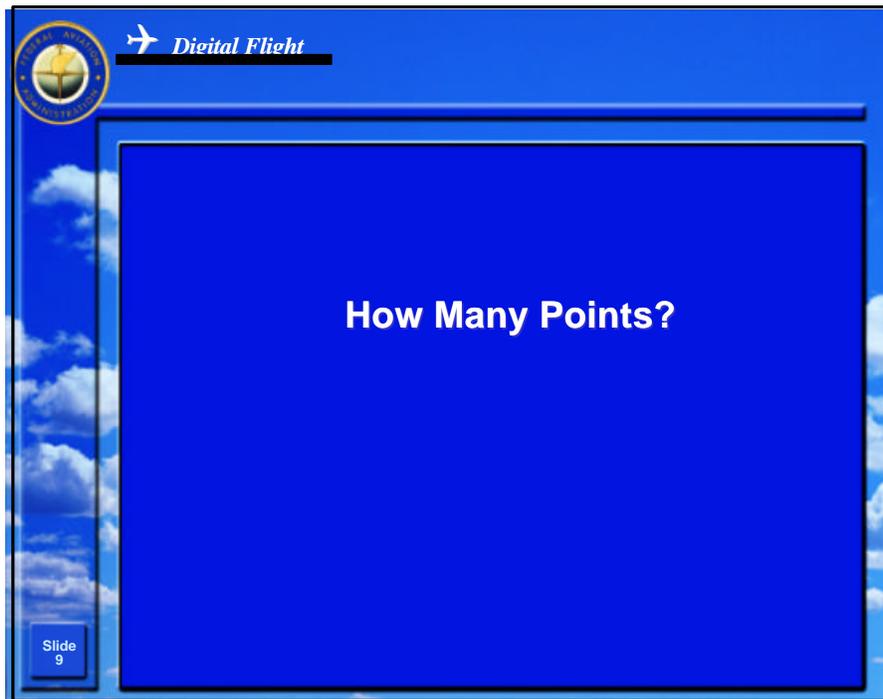
How Many Points?



Slide 8

FAA National Software Conference

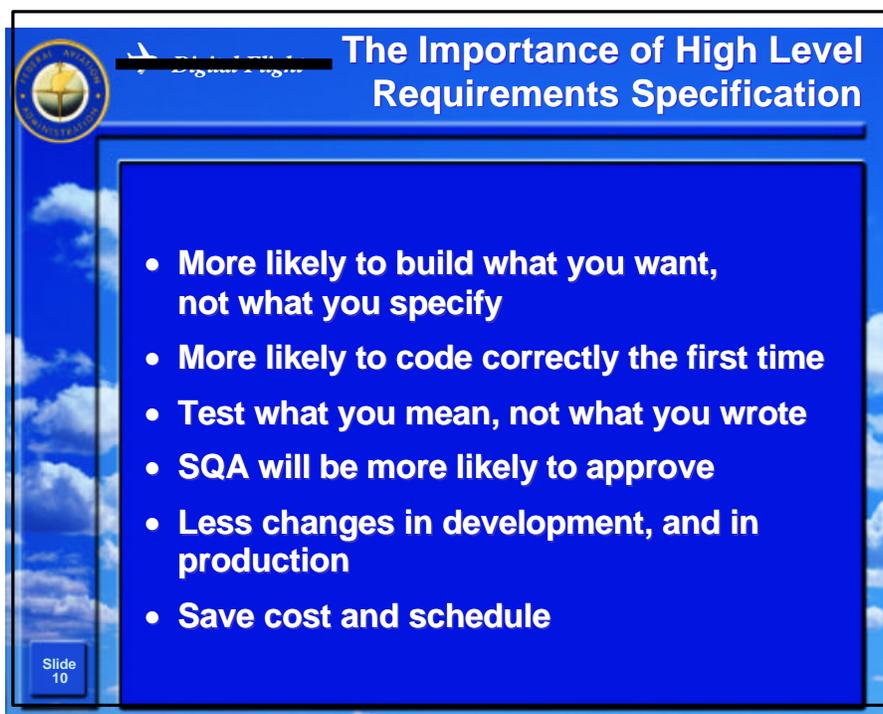
Software Verification



Slide 9

How Many Points?

The slide features a blue background with a white border. In the top left corner, there is a circular logo for the Federal Aviation Administration (FAA) and the text "Digital Flight" next to a small airplane icon. The main content is the question "How Many Points?" centered in white text. A small box in the bottom left corner contains the text "Slide 9".



Slide 10

The Importance of High Level Requirements Specification

- More likely to build what you want, not what you specify
- More likely to code correctly the first time
- Test what you mean, not what you wrote
- SQA will be more likely to approve
- Less changes in development, and in production
- Save cost and schedule

The slide features a blue background with a white border. In the top left corner, there is a circular logo for the Federal Aviation Administration (FAA) and the text "Digital Flight" next to a small airplane icon. The main content is the title "The Importance of High Level Requirements Specification" and a bulleted list of six points. A small box in the bottom left corner contains the text "Slide 10".

FAA National Software Conference

Software Verification

FAA Logo

Digital Flight

Software
High Level Requirements (Cont'd)

Requirements address:

- Functionality
- External Interfaces
- Performance
- Quality Attributes
- Design Constraints
- Safety/security

Slide 11

FAA Logo

Digital Flight

Software
High Level Requirements . . .

Should NOT address design

- Partitioning software into modules
- Allocating functions into modules
- Flow of information or control between modules

Should NOT address project management

- Cost
- Schedule
- Quality assurance
- Software development methodology

Slide 12

FAA National Software Conference

Software Verification



 *Digital Flight*

Criteria for Good High Level Requirements

Assure high-level requirements are:

- Correct
- Uniquely identified
- Non-ambiguous
- Consistent
- Traceable
- Verifiable

Slide 13



 *Digital Flight*

Example of Ambiguity

- “Aircraft that are threats have either a known status or the potential to enter restricted airspace within 5 minutes shall raise an alert.”
- “Aircraft that are either threats and have a known status or have the potential to enter restricted . . .”
- “Aircraft that are threats and either have an unknown status or the “potential to enter restricted . . .”

Slide 14

FAA National Software Conference

Software Verification



Digital Flight

Problems
Translating English to PDL

Not WOW or Wheel Speed greater than 90 mph

DeMorgans Theorem

Not (A or B) = Not A and Not B

Not (A) or B?

Slide 15



Qualities of High Level Requirements

<p>Correct</p> <ul style="list-style-type: none"> • No tool or procedure assures correctness (a fool with a tool is still a fool) • Mathematical algorithms • Complete (No TBDs) 	<p>Consistent</p> <ul style="list-style-type: none"> • Conflicting characteristics (blue/green) • Conflicting logic (A or B, A and B) • Different terms—(cue, prompt)
<p>Traceable</p> <ul style="list-style-type: none"> • Trace-up to system level req, TSO req • Trace down to data derived from high level req 	<p>Verifiable</p> <ul style="list-style-type: none"> • Each requirement is stated in quantifiable terms • For each requirement, can a test be formulated that will unambiguously answer whether the requirement has been met?

Slide 16

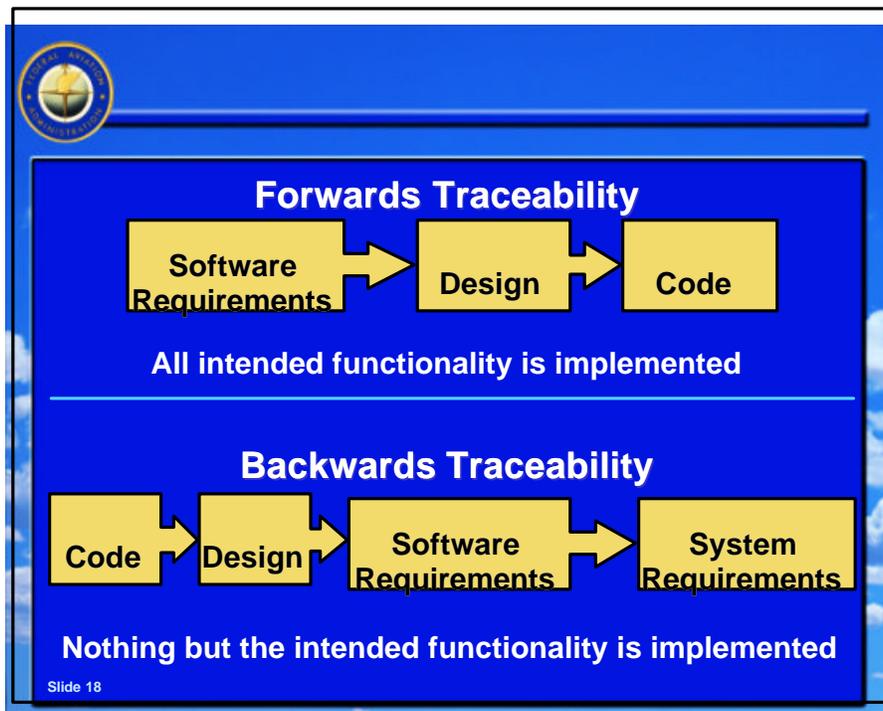
FAA National Software Conference

Software Verification

Requirements Traceability

System Req'mts	Software Req'mts	Design	Code/Module	Test
3.1.4	2.4.3	4.2.1	Correlate	4
			Associate	4
	4.1.2	2.3.6	Track update	4
			Track initiate	

Slide 17



FAA National Software Conference

Software Verification



Digital Flight Requirements Verifiability

Non-verifiable	Verifiable
<p>The software shall provide accuracy sufficient to support terrain navigation</p> <p>The software shall provide response to pilots immediately</p>	<p>The software shall compute aircraft position with an accuracy of:</p> <ul style="list-style-type: none">± 20 ft. in the horizontal± 10 ft. in the vertical <p>The system shall respond to:</p> <ul style="list-style-type: none">• Safety critical pilot actions in less than 40 milliseconds• Non safety critical pilot actions in less than 30 seconds

Slide 19



Digital Flight Non-testable Requirements

- The software shall be robust
- The software shall degrade gracefully under stress
- The software shall be developed in accordance with modern programming practices
- The software shall provide the necessary processing under all modes of operations
- Computer memory utilization shall be minimized to accommodate future growth
- The software shall be easy to use

Slide 20

FAA National Software Conference

Software Verification



Digital Flight

Words Difficult to Quantify

- Sufficient
- Modular
- Achievable
- Adequate
- Efficient
- Accomplished
- Possible
- Better/higher/slower
- Generally/normally
- Earliest/latest/highest
- Simultaneous
- Nominal/normal/average

Slide 21



Digital Flight

Requirements Quiz

“The System shall as a goal, calculate aircraft heading, altitude, and global position to a sufficient accuracy.”

Is this acceptable -- Why or Why Not?

Slide 22

FAA National Software Conference

Software Verification



Digital Flight

Problems Found in Requirements Specifications

- Design details given too early
- Data flow inconsistent
- Desired outputs not derived from the given inputs
- All scenarios not identified and thought through
- Engineers don't specify areas they know tightly (assume others know it too)
- Engineers specify only generally, areas not known or understood
- Usual ambiguities, errors of omission, etc.

Slide 23



Digital Flight

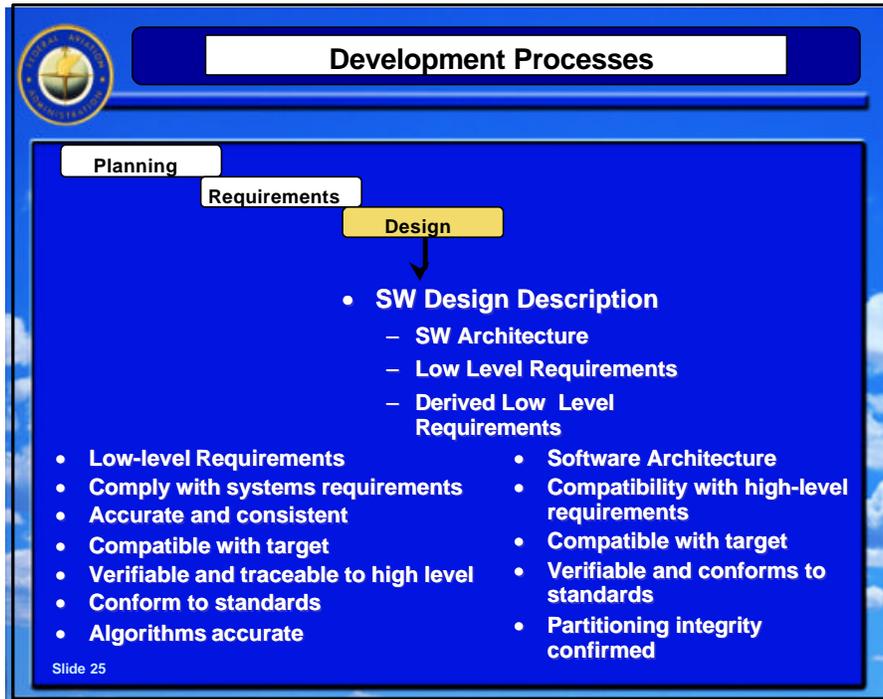
Requirements Summary

- Requirements are foundation for the development
- Requirements specification is an art even if you know what you want to build
- Requirements review (scrub) is important
- Most software errors are due to poorly specified requirements

Slide 24

FAA National Software Conference

Software Verification



Example of Low Level Requirements

Description: The **check__logic()** function compares commanded versus sensed states of the ACrvdt1, Accrvdt3, Ervdt2 and the Bus Relay. When failures are detected, this procedure sets failure bits in the ACR failure array. The types of failures can be in either the Logic card, the 101 (ACRVDT1, ERVDT2, ACCRVDT3 or Bus Relay) or a continuity failure.

Slide 26

FAA National Software Conference

Software Verification



Example of Low Level Requirements (Cont'd)

Detailed Requirements:

10. Determine if it is necessary to perform this test by determining whether the processor is the active processor.
20. Perform this test for all 4 Logic Cards (ACrvdt1, Accrvdt3, Ervdt2 and the Bus Relay)
30. Obtain the Auxiliary contact state, from memory, for the current RVDT being evaluated.
40. Compare the RVDT's commanded and sensed states to determine if a failure condition exists.
50. If the comparison of 4. yields that there is a failure, signify the failure, otherwise signify no failure.
60. Given a failure, for the current SLC being evaluated, determine whether or not it is persistent.
70. If a failure is determined to be persistent report the failure in ARINC label 101.

Slide 27



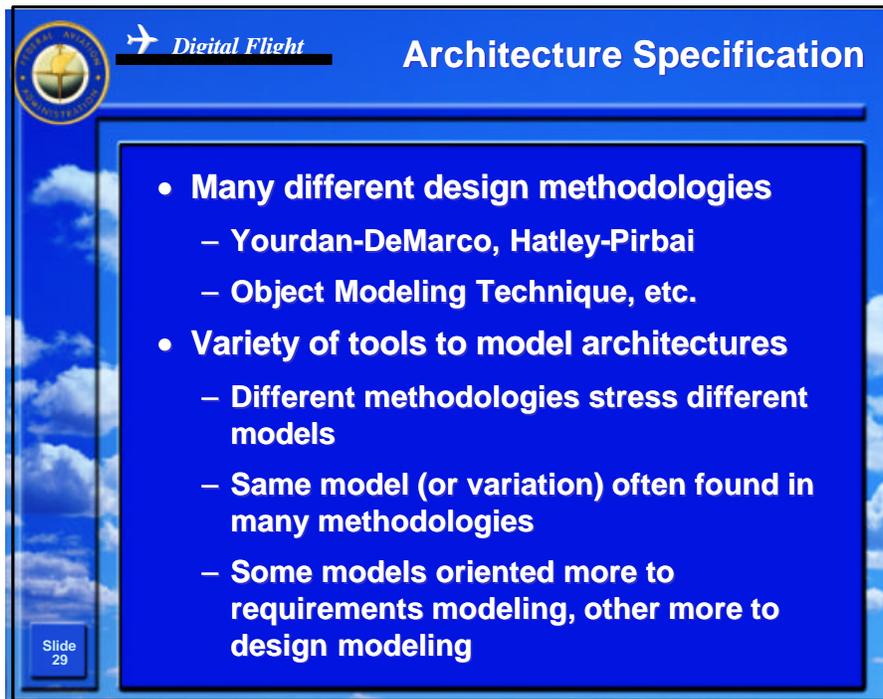
Architecture

- Shows physical partitioning of the system into components and flow of data between them
- Allocates functional processes to physical units
- Shows control flow (what process activates another)

Slide 28

FAA National Software Conference

Software Verification



Slide 29: Architecture Specification

Digital Flight

- Many different design methodologies
 - Yourdan-DeMarco, Hatley-Pirbai
 - Object Modeling Technique, etc.
- Variety of tools to model architectures
 - Different methodologies stress different models
 - Same model (or variation) often found in many methodologies
 - Some models oriented more to requirements modeling, other more to design modeling



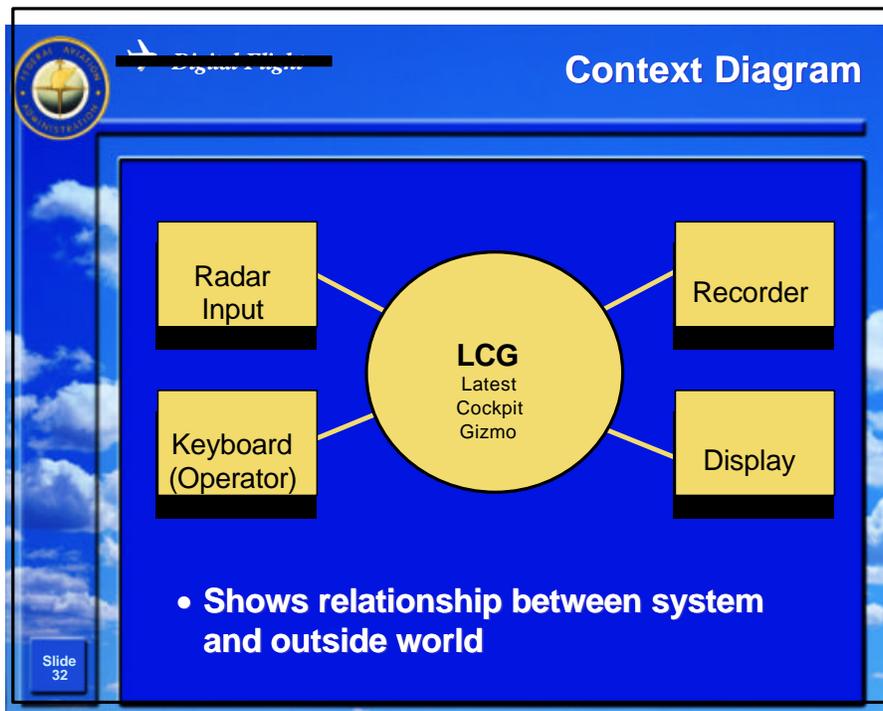
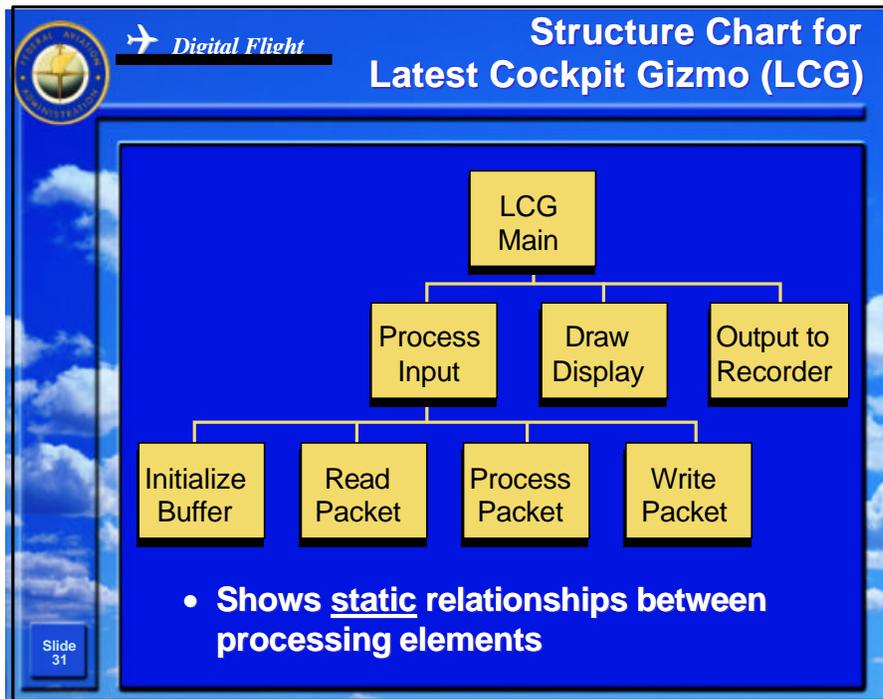
Slide 30: Looking for a Few Good Design Models . . .

Digital Flight

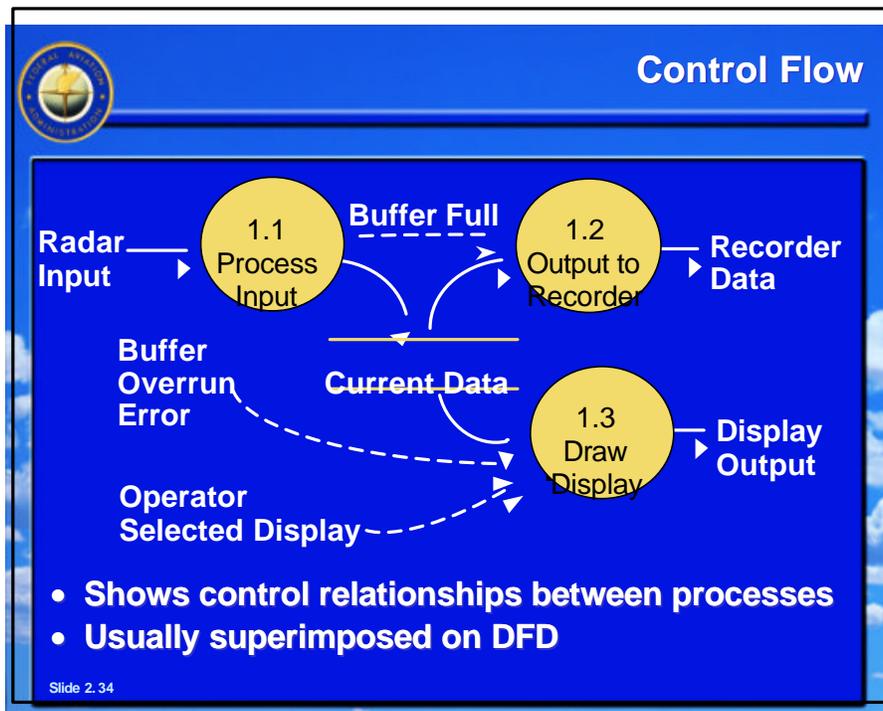
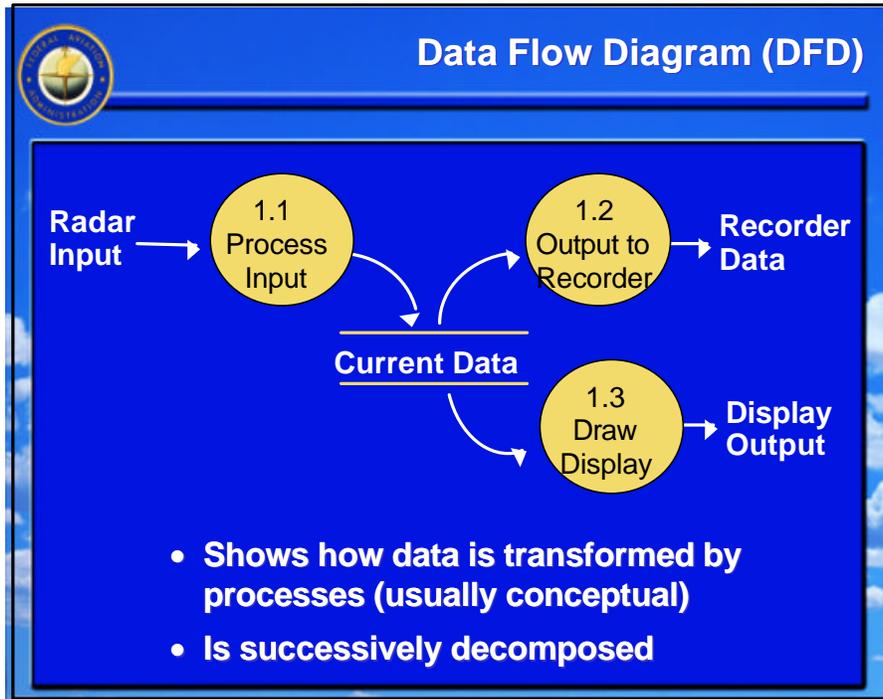
- Structured
- Context Diagram
- Data Flow
- Control Flow
- State Diagram
- Object

FAA National Software Conference

Software Verification

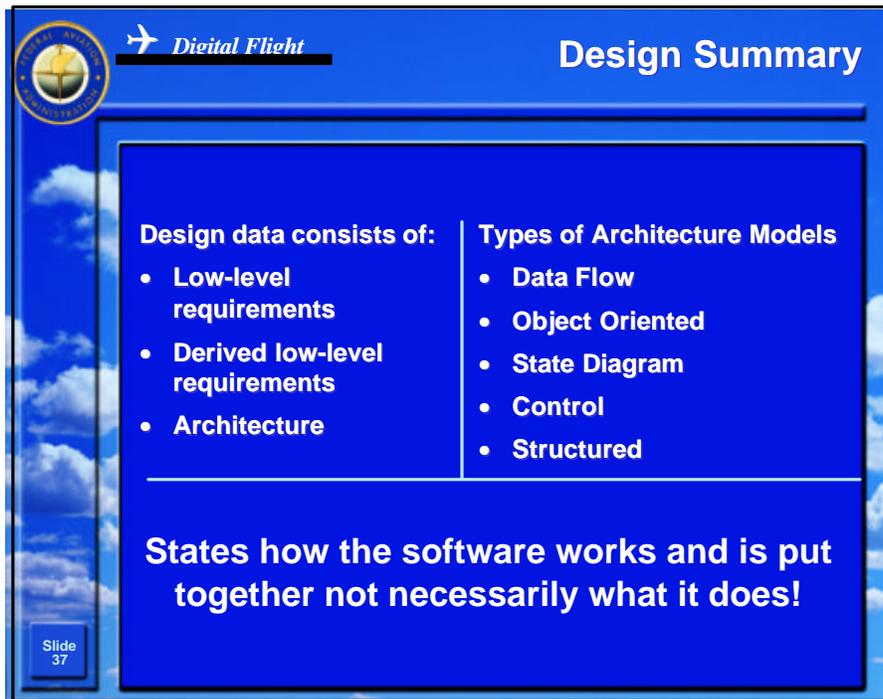


FAA National Software Conference Software Verification



FAA National Software Conference

Software Verification



FAA Logo

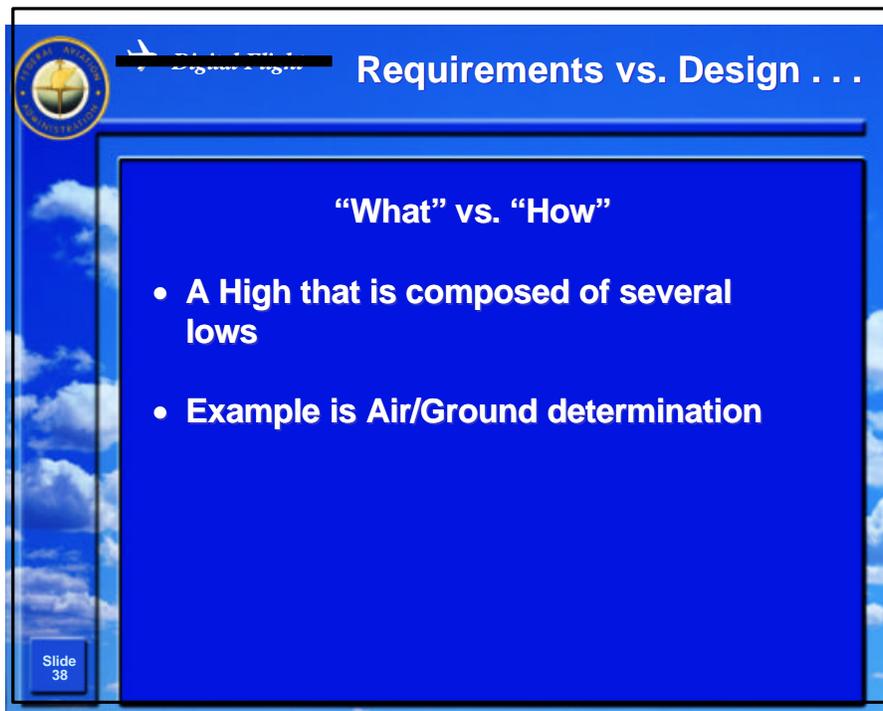
Digital Flight

Design Summary

Design data consists of:	Types of Architecture Models
<ul style="list-style-type: none">• Low-level requirements• Derived low-level requirements• Architecture	<ul style="list-style-type: none">• Data Flow• Object Oriented• State Diagram• Control• Structured

States how the software works and is put together not necessarily what it does!

Slide 37



FAA Logo

Digital Flight

Requirements vs. Design . . .

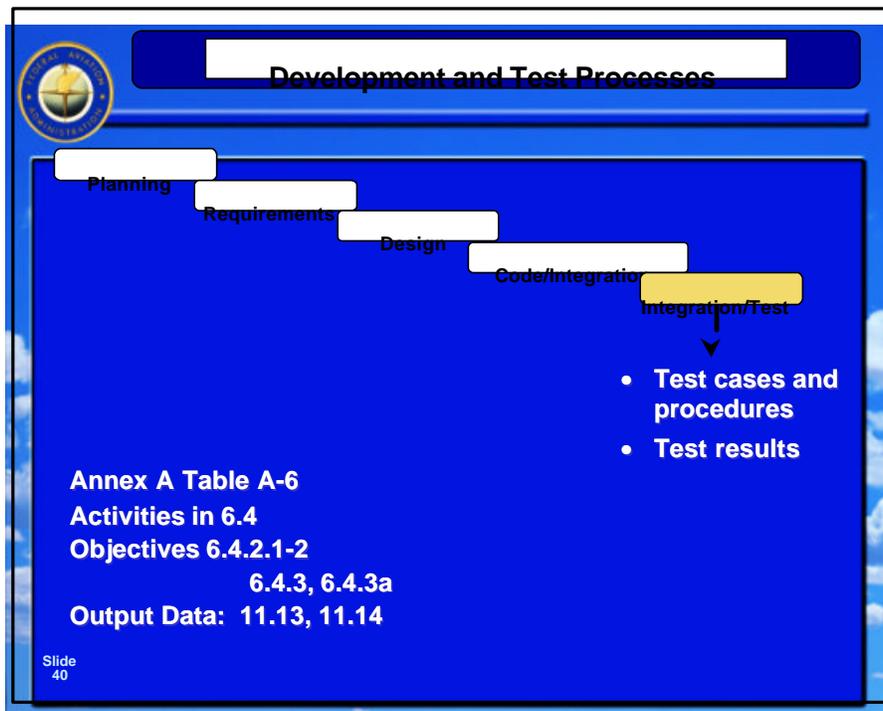
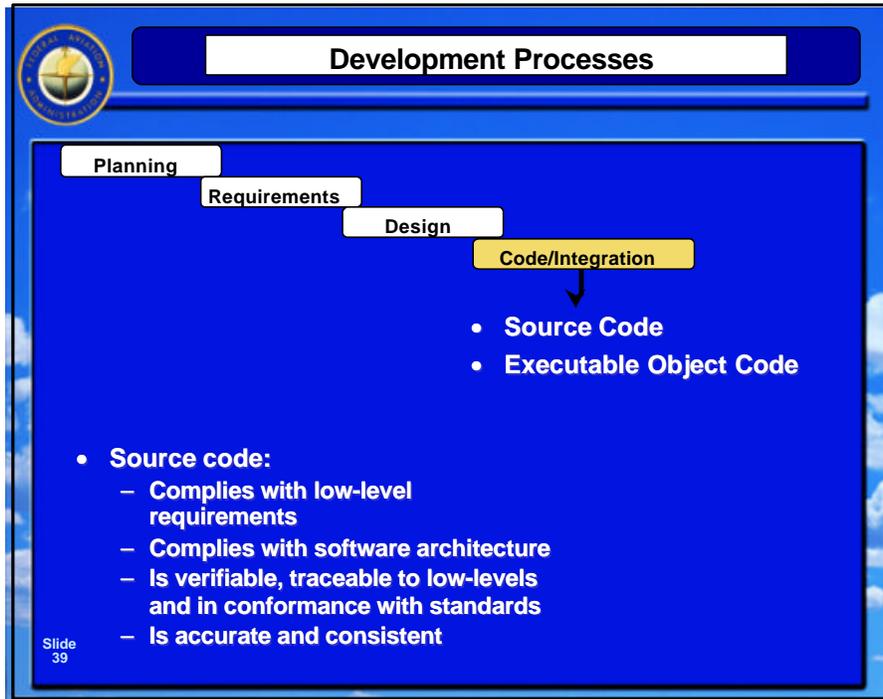
“What” vs. “How”

- A High that is composed of several lows
- Example is Air/Ground determination

Slide 38

FAA National Software Conference

Software Verification



FAA National Software Conference

Software Verification

 **Digital Flight** **Testing Of Outputs of Integration Process—Table A-6**

Objective	Objective Number
• Executable meets:	
– High-level requirements	1
– Low-level requirements	3
• Executable is robust with:	
– High-level requirements	2
– Low-level requirements	4
• Executable	
– Compatible with target	5

Slide 41

 **Digital Flight** **Avoiding Testing On The Target**

Some target environments may not support the invasive testing techniques (e.g., peeking and poking) needed to determine if the software meets the requirements.

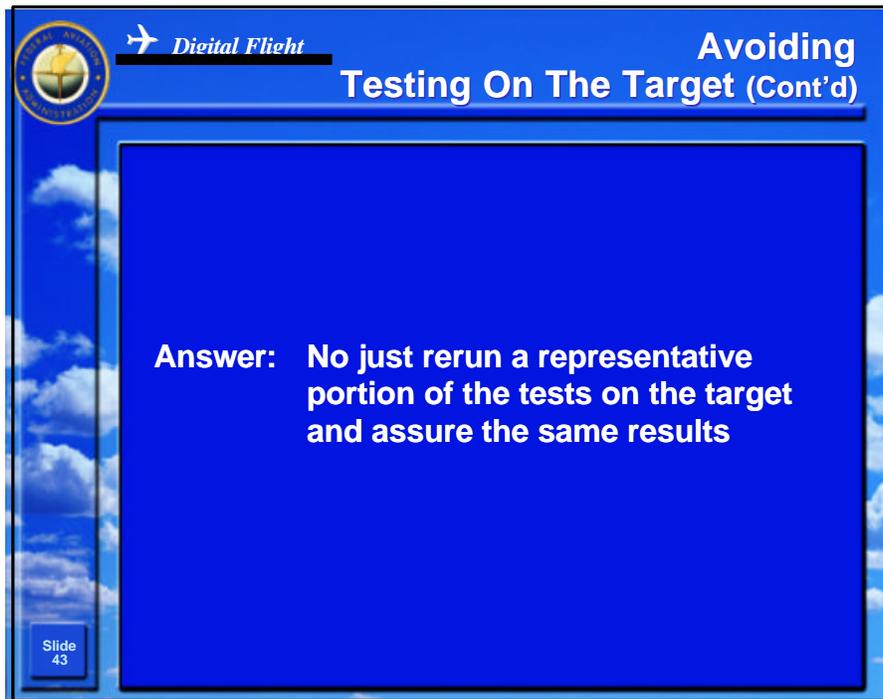
Question:
Before an applicant uses an In Circuit Emulator for verification:

- Must they qualify it (i.e. assure it works correctly)?
- Do they have to re-run all the tests on the target?

Slide 42

FAA National Software Conference

Software Verification

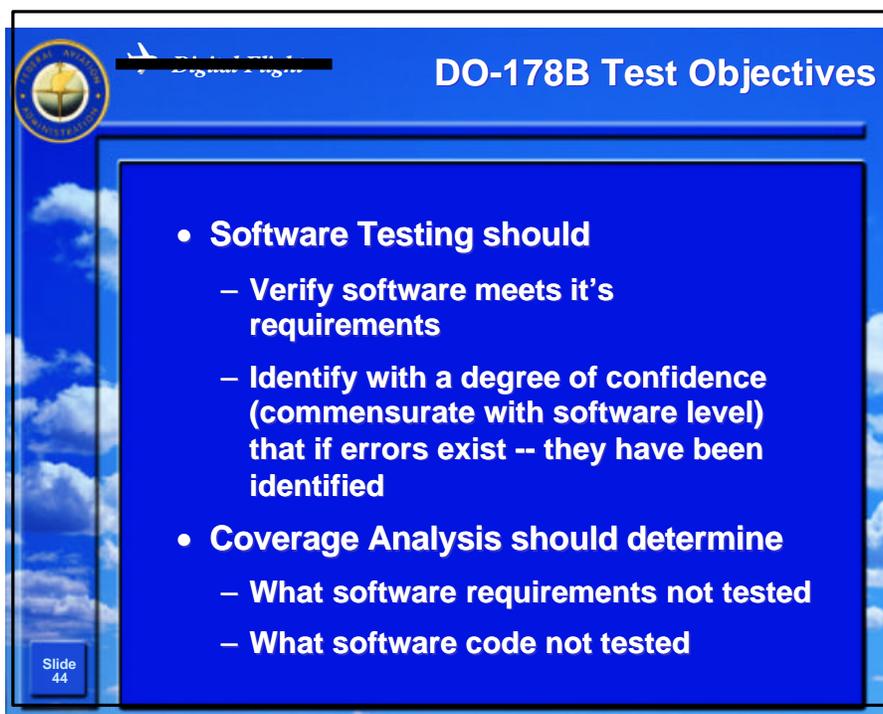


FAA Logo

~~Digital Flight~~ **Avoiding Testing On The Target (Cont'd)**

Answer: No just rerun a representative portion of the tests on the target and assure the same results

Slide 43



FAA Logo

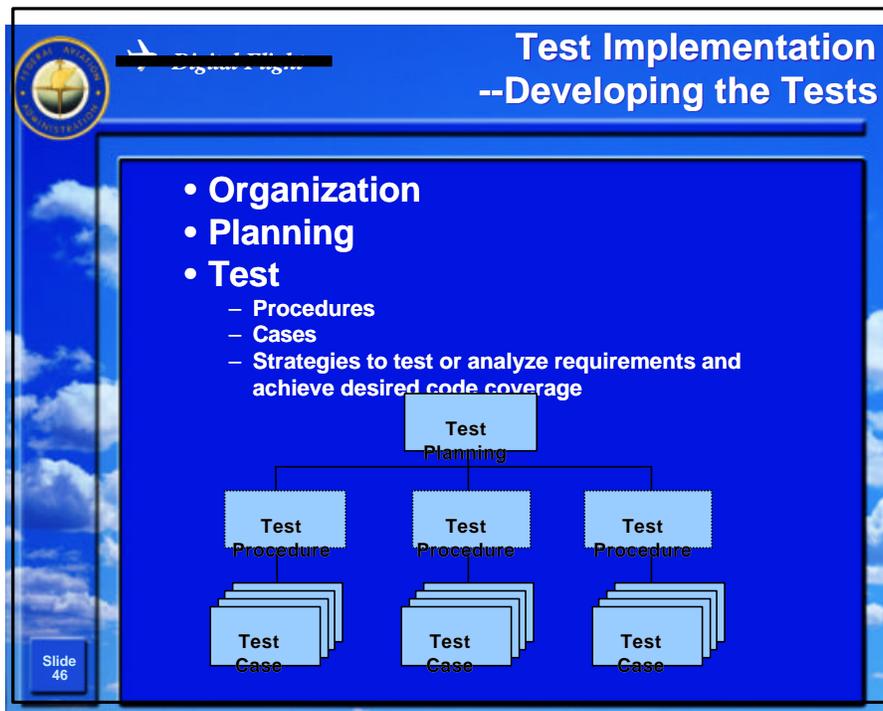
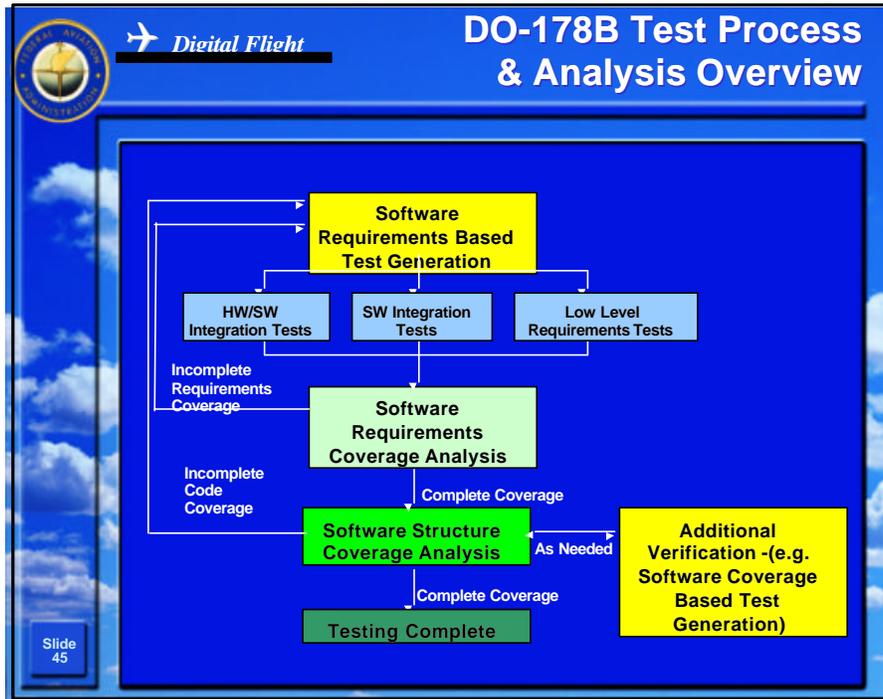
~~Digital Flight~~ **DO-178B Test Objectives**

- **Software Testing should**
 - Verify software meets it's requirements
 - Identify with a degree of confidence (commensurate with software level) that if errors exist -- they have been identified
- **Coverage Analysis should determine**
 - What software requirements not tested
 - What software code not tested

Slide 44

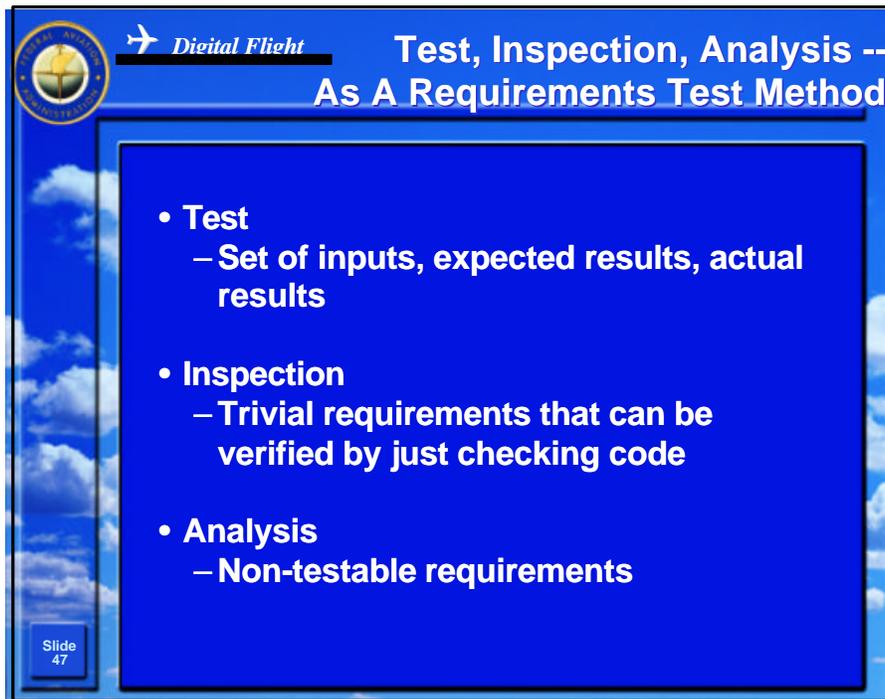
FAA National Software Conference

Software Verification



FAA National Software Conference

Software Verification

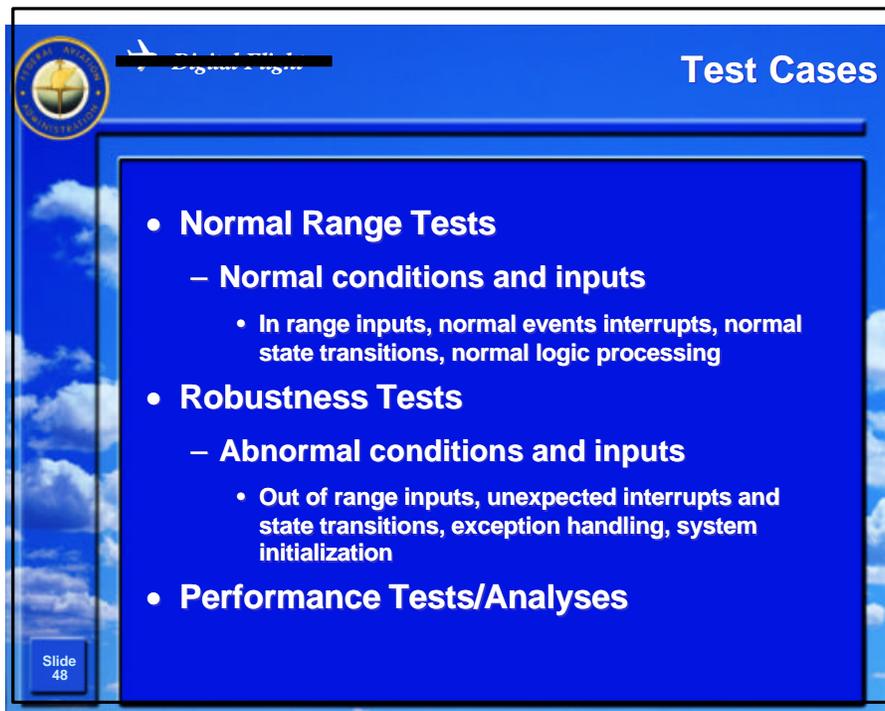


FAA National Software Conference
Digital Flight

Test, Inspection, Analysis -- As A Requirements Test Method

- **Test**
 - Set of inputs, expected results, actual results
- **Inspection**
 - Trivial requirements that can be verified by just checking code
- **Analysis**
 - Non-testable requirements

Slide 47



FAA National Software Conference
Digital Flight

Test Cases

- **Normal Range Tests**
 - Normal conditions and inputs
 - In range inputs, normal events interrupts, normal state transitions, normal logic processing
- **Robustness Tests**
 - Abnormal conditions and inputs
 - Out of range inputs, unexpected interrupts and state transitions, exception handling, system initialization
- **Performance Tests/Analyses**

Slide 48

FAA National Software Conference

Software Verification

 *Digital Flight* **Normal and Robust Testing**

- **Normal range test cases**—how software responds to normal inputs
 - Valid input equivalence classes
 - Performance related functions
 - Algorithms (logic and arithmetic)
 - State transitions
- **Robustness test cases**—how software responds to abnormal inputs
 - Invalid input equivalence classes
 - System initialization—abnormal conditions
 - Out of range loop variants
 - Provoke invalid state transitions

Slide 49

 *Digital Flight* **Test Case Levels**

Low Level

High Level

R
e
q
u
i
r
e
m
e
n
t
s

Low

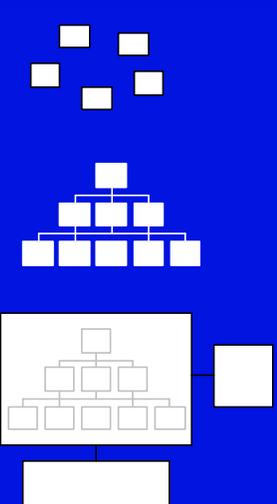
- Low Level Requirements Testing
- Unit/Module Testing

Intermediate

- Software Integration Tests
- Test Simulators

High

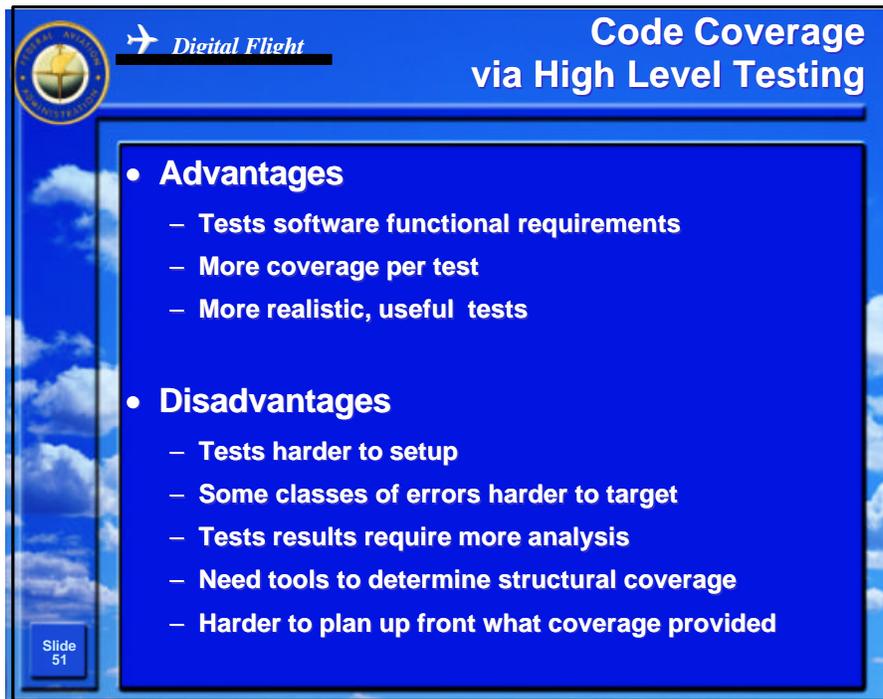
- Hardware Software Integration Tests
- Target Environment



Slide 50

FAA National Software Conference

Software Verification



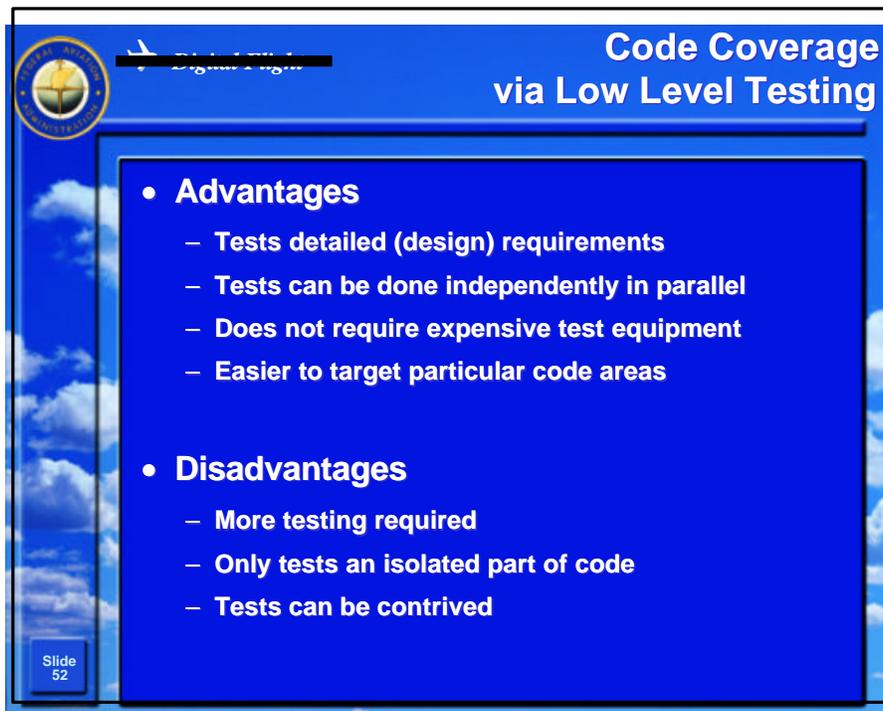
FAA Logo

Digital Flight

Code Coverage via High Level Testing

- **Advantages**
 - Tests software functional requirements
 - More coverage per test
 - More realistic, useful tests
- **Disadvantages**
 - Tests harder to setup
 - Some classes of errors harder to target
 - Tests results require more analysis
 - Need tools to determine structural coverage
 - Harder to plan up front what coverage provided

Slide 51



FAA Logo

Digital Flight

Code Coverage via Low Level Testing

- **Advantages**
 - Tests detailed (design) requirements
 - Tests can be done independently in parallel
 - Does not require expensive test equipment
 - Easier to target particular code areas
- **Disadvantages**
 - More testing required
 - Only tests an isolated part of code
 - Tests can be contrived

Slide 52

FAA National Software Conference

Software Verification



Digital Flight Code Coverage Via Analysis

- **Advantages**
 - May be less expensive to setup
 - Does not require tools or code instrumentation
- **Disadvantages**
 - More labor intensive
 - Needs to be repeated each time code changes/tests rerun
 - Can be less rigorous (error-prone and tedious process)
 - few people do it right -- risk!!

Slide 53



Digital Flight Error Detection Objectives*

Low Level Testing	Software Integration Testing	Hardware/Software Integration Tests
<ul style="list-style-type: none">• Algorithm Failures• Incorrect Loop Operations• Incorrect Logic Decisions• Failure to Process Correct Input Combinations• Incorrect Response to Bad Input Data• Incorrect Exception Handling• Incorrect Computation Sequence• Inadequate Algorithm Precision, Accuracy, Performance	<ul style="list-style-type: none">• Incorrect Initialization of Variables• Parameter Passing Errors• (Global) Data Corruption• Inadequate Numerical Resolution• Incorrect Sequencing of Events and Operations	<ul style="list-style-type: none">• Incorrect Interrupt Handling• Miss Timing Requirements• Hardware Transient Errors• Resource Contention• BIT Detection Errors• Bad Feedback Loops• Incorrect Device Control• Stack Overflow• Incorrect Load Version Verification• Software Partitioning Violations

Slide 54

* DO-178B 6.4.3

FAA National Software Conference

Software Verification

 *Digital Flight* Test Summary

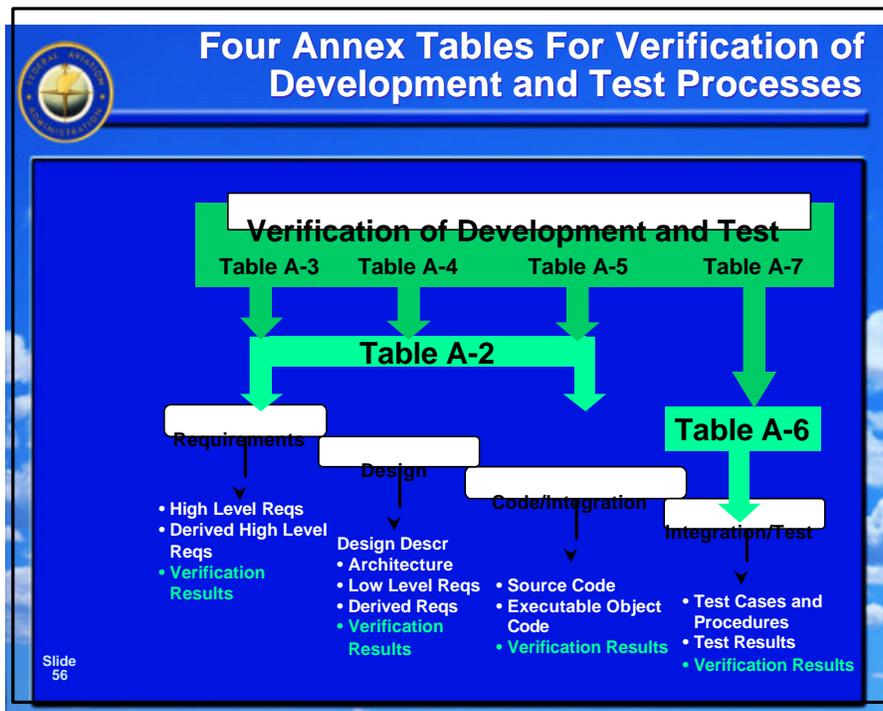
The test process is non-trivial.

Analysis within the test context is confusing since it can have several meanings.

There are many strategies to testing, decisions are made based on the type of development.

You will see many different approaches to achieve the same objectives and will be required to assess them.

Slide 55



FAA National Software Conference

Software Verification

Definitions in DO-178B

- **Verification**—evaluation of the results of a process to ensure correctness and consistency with respect to inputs and standards (Glossary)

Verification:

- **Analysis**—repeatable assessment (par 6.3)
- **Review**—qualitative assessment of correctness (par 6.3)
- **Test**—process of exercising a system or system component to verify it satisfies specific requirements and to detect errors (Glossary)

Slide 57

Verification

Planning Requirements Design Code/Integration Integration/Test

- **SW Requirements Data**
 - High Level Requirements
 - Derived requirements

Requirements Verification

Annex A Table A-3
Objectives In 6.3.1a-g
Activities in 5.1.2
Data Description 11.9
Output Data in 11.14

Slide 58

FAA National Software Conference

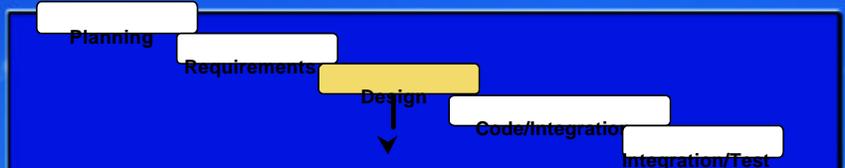
Software Verification

 *Digital Flight* Verification of Outputs Of Software Requirements Process—Table A-3

Objective	Objective Number
• High-level requirements:	
– Comply with systems requirements	1
– Accurate and consistent	2
– Compatible with target	3
– Verifiable	4
– Conform to standards	5
– Traceable to system	6
• Algorithms accurate	7

Slide 59

 Verification



- SW Design Description
 - SW Architecture
 - Low Level Requirements
 - Derived Requirements

Design Verification
 Annex A Table A-4
 Objectives In 6.3.2 a-g
 6.3.3 a,b-f
 Activities in 5.2.2
 Data Description 11.10
 Output Data 11.14

Slide 60

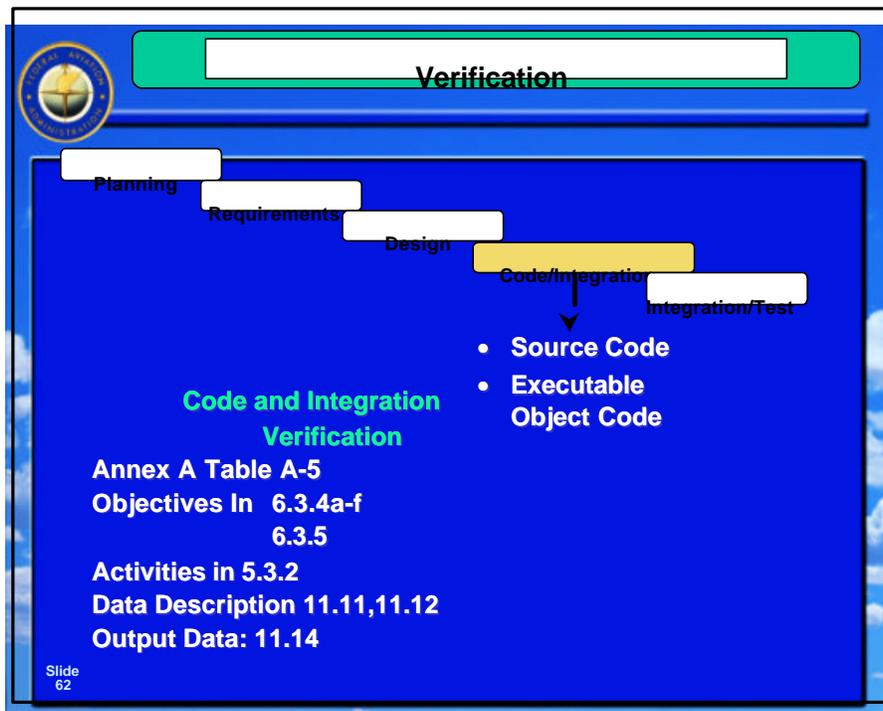
FAA National Software Conference

Software Verification

Verification of Outputs of Software Design Process—Table A-4

Objective	Objective Number
• Low-level requirements:	
– Comply with high-level	1
– Accurate and consistent	2
– Compatible with target	3
– Verifiable and traceable to high-level	4, 6
– Conform to standards	5
– Algorithms accurate	7
• Software Architecture	
– Compatible with high-level	8
– Consistent	9
– Compatible with target	10
– Verifiable and conforms to standards	11, 12
– Partitioning integrity confirmed	13

Slide 61



FAA National Software Conference

Software Verification



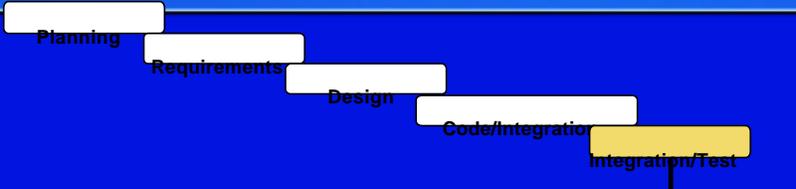
Digital Flight Verification of Outputs of Coding and Integration—Table A5

Objective	Objective Number
• Source Code:	
– Complies with low-level	1
– Complies with architecture	2
– Verifiable and conforms to standards	3, 4
– Traceable to low-level	5
– Accurate and consistent	6
– Complete and correct	7

Slide 63



Verification



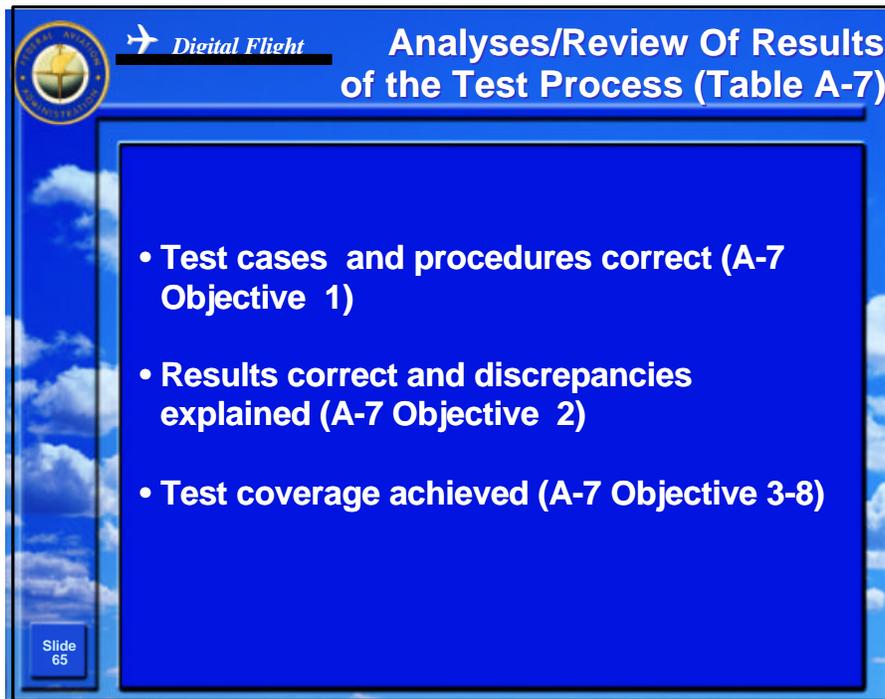
• Test cases and procedures
 • Test results

Verification of Verification
 Annex A Table A-7
 Objectives In: 6.3.6b-c, 6.4.4.1, 6.4.4.2 a-c
 Data Description 11.13, 11.14
 Output Data: 11.14

Slide 64

FAA National Software Conference

Software Verification



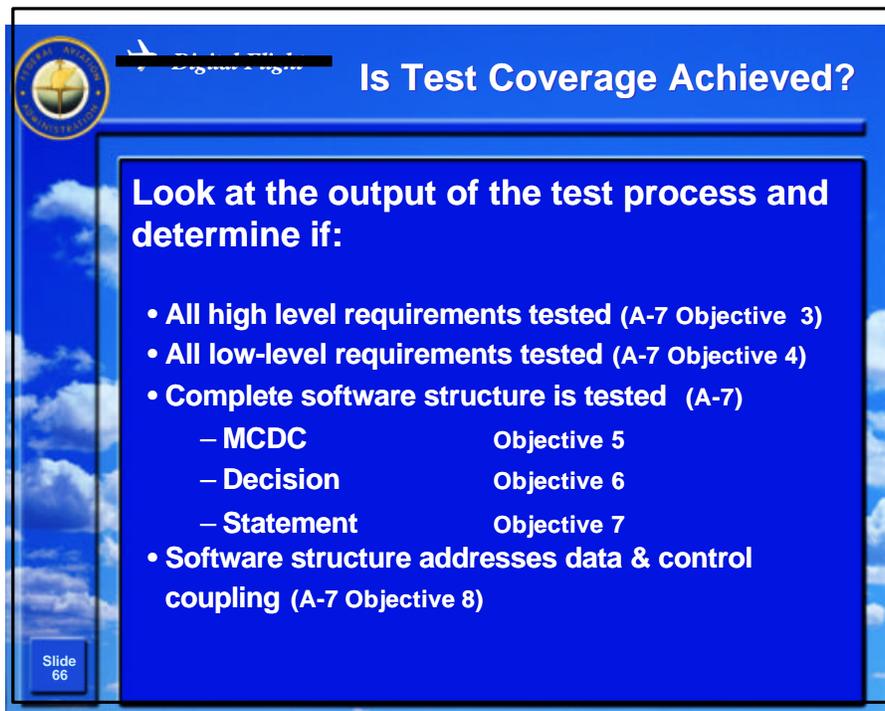
FAA Logo

Digital Flight

Analyses/Review Of Results of the Test Process (Table A-7)

- Test cases and procedures correct (A-7 Objective 1)
- Results correct and discrepancies explained (A-7 Objective 2)
- Test coverage achieved (A-7 Objective 3-8)

Slide 65



FAA Logo

Digital Flight

Is Test Coverage Achieved?

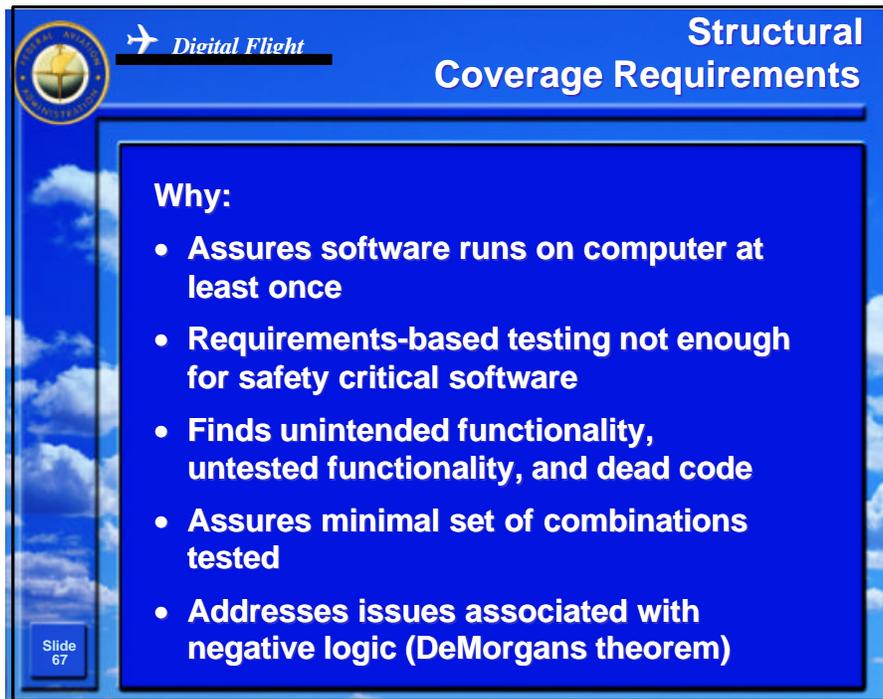
Look at the output of the test process and determine if:

- All high level requirements tested (A-7 Objective 3)
- All low-level requirements tested (A-7 Objective 4)
- Complete software structure is tested (A-7)
 - MCDC Objective 5
 - Decision Objective 6
 - Statement Objective 7
- Software structure addresses data & control coupling (A-7 Objective 8)

Slide 66

FAA National Software Conference

Software Verification

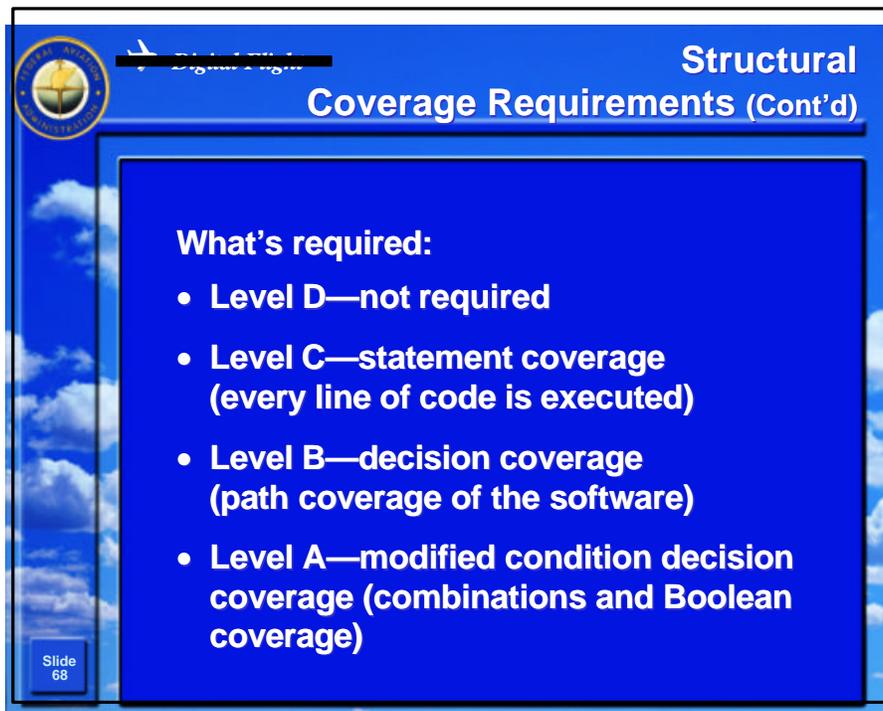


FAA National Software Conference logo and "Digital Flight" text are visible in the top left corner. The slide title is "Structural Coverage Requirements".

Why:

- Assures software runs on computer at least once
- Requirements-based testing not enough for safety critical software
- Finds unintended functionality, untested functionality, and dead code
- Assures minimal set of combinations tested
- Addresses issues associated with negative logic (DeMorgans theorem)

Slide 67



FAA National Software Conference logo and "Digital Flight" text are visible in the top left corner. The slide title is "Structural Coverage Requirements (Cont'd)".

What's required:

- Level D—not required
- Level C—statement coverage (every line of code is executed)
- Level B—decision coverage (path coverage of the software)
- Level A—modified condition decision coverage (combinations and Boolean coverage)

Slide 68

FAA National Software Conference

Software Verification

What is Structural Coverage?

Requirements Based Test Case 1 Requirements Based Test Case 2

Resulting Structural Coverage Resulting Structural Coverage

Slide 69

Level A Modified Condition Decision Coverage

If (A or B) and C

All combinations for 3 variables = 8

MCDC—minimum of n+1 test cases

- Make A or B both true and false
- Simultaneously make the expression true and false

	A	B	C	
	0	0	0	F
	0	0	1	F
	0	1	0	F
	0	1	1	T
	1	0	0	F
	1	0	1	T
	1	1	0	F
	1	1	1	T

Slide 70

FAA National Software Conference

Software Verification



 *Digital Flight*

Level B--
Decision Coverage

Decision Coverage

```
if (WOW and wheel speed < 80 knots)
  on-ground
else
  air
endif
```

Need to hit both logic paths

- Assures all logic paths are taken for the right reason

Slide 71



 *Digital Flight*

Level C --
Statement Coverage

Statement Coverage

Every line of code must be executed!

- Finds dead code
- Assures that all code has run on target at least once

Slide 72

FAA National Software Conference

Software Verification

 *Digital Flight* What is Structural Coverage Analysis?

Determining if the Test Coverage of the Software Structure is Achieved!

Resulting Structural Coverage



100
200
300
400
500
600

Slide 73

 *Digital Flight* Potential Causes For Missing Structural Coverage

Resulting Structural Coverage



100
200
300
400
500
600

- Dead code
- Missing requirements
- Unintended functionality
- Poor testing

Slide 74

FAA National Software Conference

Software Verification

 *Digital Flight* Dead Code

Code that can not be executed due to the way it is coded.

Example:

```
if (A and B) and not C
  compute height
  if C
    compute weight
  end if
end if
```

Slide 75

 *Digital Flight* Test Coverage of the Data and Control Coupling

Assure test process has tested the data and control aspects of the software

Data Coupling	Control Coupling
<ul style="list-style-type: none">• Interfaces• Typing• Global usage	<ul style="list-style-type: none">• Calling tree• Timing of events• Scheduler/transition• Interrupts

Slide 76

FAA National Software Conference

Software Verification



Digital Flight

MCDC Analysis

- **Single Term IF Statements**
 - MCDC ensures both branches of the IF executed
- **Multiple Term IF Statements**
 - MCDC Analysis required if -- IF statement has 2 or more logic terms

Note: There are other logic statements besides IF in many languages, e.g. DO WHILE

Slide 77



Digital Flight

IF Statement Coverage

- **Does the following require 1 or 2 test cases for full coverage?**

```
If ( A == 0 )  
    x = 1;  
x = x + 1;
```

Slide 78

FAA National Software Conference

Software Verification

 **Digital Flight** **IF Statement Coverage**

- It can require 2, depending on how the compiler implements IF statements, e.g.

	CASE 1 (A=0)	CASE 2 (A=1)
<code>If (A == 0)</code>	<code>LD A,0</code>	<code>LD A,0</code>
	<code>JEQ A,L1</code>	<code>JEQ A,L1</code>
	<code>JMP L2</code>	<code>JMP L2</code>
<code> x = 1;</code>	<code>L1 ST X,1</code>	<code>L1 ST X,1</code>
<code> x = x + 1;</code>	<code>L2 ADD X,1</code>	<code>L2 ADD X,1</code>

- To be safe, always test both sides of the IF

Slide 79

 **Digital Flight** **MCDC and Logic Terms**

- **Logic Term Rules**
 - any term with two operands and an operator which evaluates to TRUE or FALSE
 - second operand can be implicit (e.g. !A is equivalent to A == 0)
 - logic terms are typically combined together via ANDs, ORed, or other logical operators
 - a variable which is ANDed or ORed with a constant is not a logic term, it is really a variable (it evaluates to a value, not T or F)

Slide 80

FAA National Software Conference

Software Verification

 *Digital Flight* MCDC Example

How many Logic Terms?
Is MCDC Coverage Analysis Required?

```
if ( alarm_hist [ k ] > 0 )  
  
if ( ( state [ i ] & MASK ) != 0xA )  
  
if ( temp & BIT12 + BIT15 )  
  
if ( ( status [ i ] == HIGH) && ( ( states & 4 ) == 0 ) )  
  
if ( !G10 && !wow1 && !wow2 )  
  
if ( ( !G10 && ( !wow1 && !wow2 ) ) || ( !wow1 || !wow2 ) )
```

Slide 81

 *Digital Flight* Summary

- Provided an understanding of how the annexes work together 
- Verification is an important part of DO-178B not found in many other standards 
- DO-178B is no silver bullet--verification is only as good as the verifier no matter which standard you use 

Slide 82