DO-332 Compliance Documentation

Overview

This document maps STUNIR DO-332 tool capabilities to DO-332 objectives.

DO-332 Objectives Coverage

OO.1 - Inheritance Analysis

Requirement: Verify class hierarchy structure is correct.

Implementation:
- `inheritance_analyzer.Calculate_Depth` - Compute DIT
- `inheritance_analyzer.Detect_Diamond_Pattern` - Find diamonds
- `inheritance_analyzer.Has_Circular_Inheritance` - Detect errors
- `inheritance_analyzer.Verify_Override` - Check override correctness

Evidence: `inheritance_report.json`

OO.2 - Polymorphism Verification

Requirement: Verify type substitution correctness.

Implementation:
- `polymorphism_verifier.Scan_Virtual_Methods` - ID virtuals
- `polymorphism_verifier.Count_Possible_Types` - Type counting
- `substitutability.Check_LSP` - LSP verification
- `substitutability.Check_Covariance` - Return type check

Evidence: `polymorphism_report.json`

OO.3 - Dynamic Dispatch Analysis

Requirement: Verify dynamic binding is deterministic.

Implementation:
- `dispatch_analyzer.Resolve_Targets` - Target enumeration
- `dispatch_analyzer.Analyze_Site` - Boundedness proof
- `vtable_builder.Build_VTable` - VTable construction
- `dispatch_analyzer.Can_Devirtualize` - Optimization

Evidence: `dispatch_report.json`

OO.4 - Object Coupling Analysis

Requirement: Verify object interactions.

Implementation:
- `coupling_analyzer.Build_Dependency_Graph` - Dependencies
- `coupling_metrics.Calculate_CBO` - CBO metric
- `coupling_metrics.Calculate_RFC` - RFC metric
- `coupling_analyzer.Detect_Circular_Dependencies` - Cycles

Evidence: `coupling_report.json`

OO.5 - Exception Handling (Basic)

Implementation: Exception flow tracking in methods.

OO.6 - Constructor/Destructor (Basic)

Implementation: Lifecycle analysis in test generation.

Verification Activities per DAL

Activity	DAL A	DAL B	DAL C	DAL D	DAL E
OO.1 Inheritance	✓	✓	✓	✓	-
OO.2 Polymorphism	✓	✓	✓	✓	-
OO.2 LSP	✓	✓	-	-	-
OO.3 Dispatch	✓	✓	-	-	-
OO.3 Timing	✓	-	-	-	-
OO.4 Coupling	✓	✓	✓	-	-

Tool Qualification

Classification: TQL-5 (verification tool, cannot insert errors)

Rationale: Tool verifies existing designs; errors would cause increased verification, not mask errors in output.

Certification Credit

Output reports can be used as certification evidence for:
- Software Design Standards compliance
- Code review automation
- Structural coverage analysis
- Test case derivation