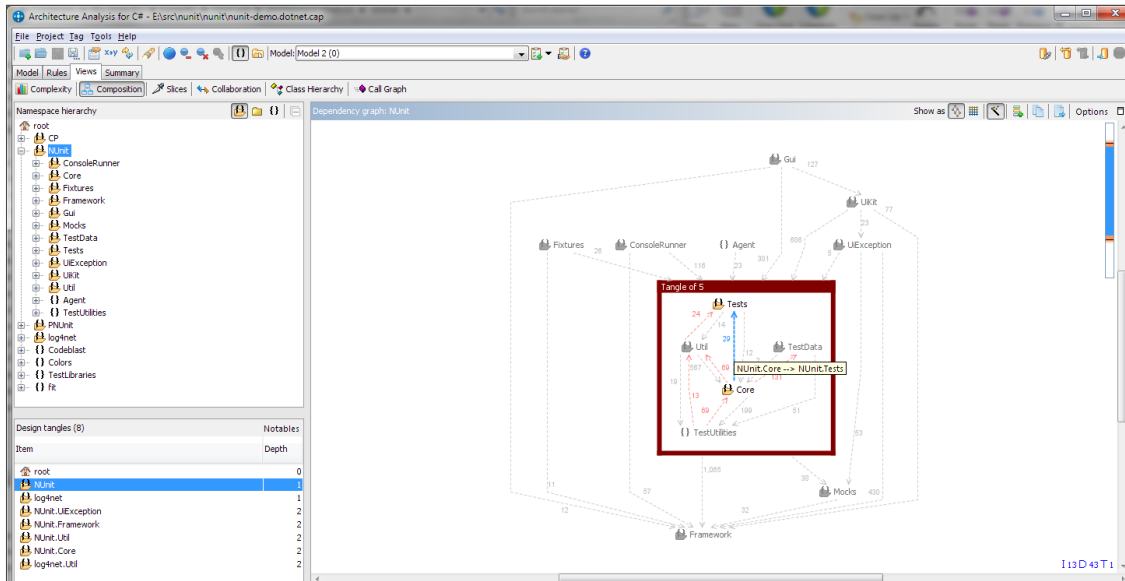


# Coverity Architecture Analysis

Coverity® Architecture Analysis discovers and maps code architecture to the actual implementation, providing the visibility to define a well-structured model that can be enforced with automatic detection of rule violations to control code complexity.



Discover, define and visualize code structure to detect rule violations and maintain architectural integrity.

## Product Overview

Coverity Architecture Analysis is part of the Coverity Development Testing Platform and provides an architecture development environment (ADE) that helps development teams organize the codebase files into a well-structured containment model that provides a modular hierarchy and keeps dependencies low and controlled. Maintaining a well-understood and enforced architecture lets developers deliver new features in a fraction of the time, at a lower cost and with fewer defects.

## Key Features

### Manage Code Complexity

Identify cyclic dependencies and other code complexities which should be simplified to reduce technical debt and assist in development and refactoring. Reducing code complexity lowers the cost of ownership by increasing maintainability and reducing defects.

### Improve Code Change Management

Improve code change management by establishing and enforcing policies for defining and controlling software architecture changes. Determine the impact of changes before committing them.

### Discover and Visualize Architecture

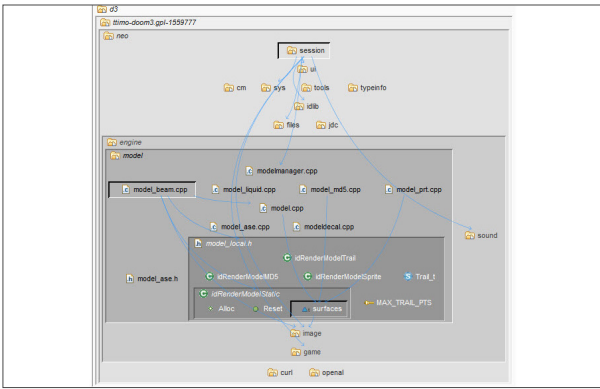
Discover existing structure from implementation and/or physical organization, and visualize all relationships and dependencies of the software components.

### Define and Enforce Architectural Integrity

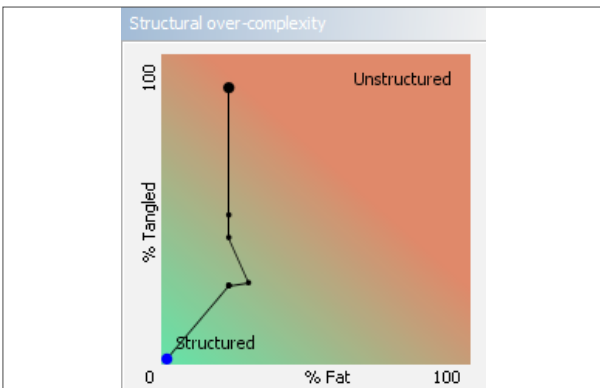
Define an ongoing architecture that maps directly to the evolving codebase. Detect and remediate architectural flaws that create quality exposures so developers can be sure they are building on a secure foundation. Enforce architectural standards and track trends with structural visibility that helps architects and developers ensure that design specifications do not degrade over time.

## IMPROVE CODE QUALITY AND MAINTAINABILITY

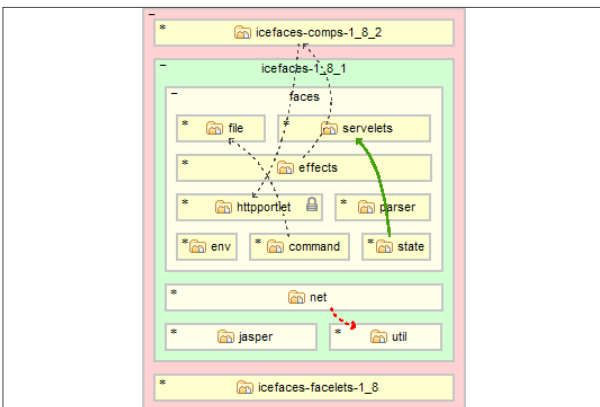
- Generate a comprehensive visual map of complex architectures
- Create a Levelized Structure Map (LSM) to track and minimize dependencies
- Automatically detect design rule violations and surface issues within a unified management console
- Identify excess complexity including cyclic dependencies
- View dependency graphs as diagrams or matrices
- Use "auto-partitioning" to find the hidden structure in large graphs
- Define well-structured containment models to control dependencies
- Analyze slices through the codebase at class, folder or design levels
- Find unused code clusters at any level



Create well-structured containment models with the LSM.



Monitor progress as changes are made with the structural over-complexity graph.



Communicate and enforce design rules with architecture diagrams.

## Common Usage Scenarios

### Architect: Discover, Define, Publish, Evolve

Coverity Architecture Analysis will discover and map the architecture to the actual implementation. Then the Levelized Structure Map (LSM) enables the creation of well-structured containment models to control dependencies. Developers can easily expand and collapse to any level and manipulate the contents, moving items from one container to another while viewing the impact on the dependencies. As they make changes, they are guided by the over-complexity graph which measures tangles, (cyclic dependencies) against fat (too much code in one place.)

To ensure ongoing architectural integrity, they must also define the architectural structure and dependency rules. Once the structure is defined, they can publish the architecture diagrams to the whole development team so that code structure evolves in a controlled manner.

### Developer: Understand, Write, Improve

It is important for developers to understand the architecture so that they can write code that follows the design rules. Coverity Architecture Analysis will surface architectural violations within Coverity Connect for easy triage and remediation. It also integrates directly into the IDE where they can view architectural diagrams and address violations and refactoring tasks.

### Manager: Monitor, Enforce, Assign

Coverity Architecture Analysis collects key metrics that allow managers to monitor complexity, track trends over time, enforce design rules and allocate resources for refactoring and other tasks. Architecture violations are visible in Coverity Connect along with all issues surfaced by Coverity development testing solutions for resolution within a unified workflow. In addition, Coverity Policy Manager provides a visual representation of the areas of risk across projects and teams. Development Managers or Application Owners can view a hierarchical heat map that is tailored specifically to the needs of their organization. They can establish a stage gate to ensure that the product is not promoted to the next phase of the life cycle until all critical issues surfaced by the Coverity platform have been addressed.

Platform Support	Language Support	IDE Support
<ul style="list-style-type: none"> <li>Windows</li> <li>Linux</li> </ul>	<ul style="list-style-type: none"> <li>C / C++</li> <li>C#</li> <li>Java</li> </ul>	<ul style="list-style-type: none"> <li>Eclipse</li> <li>Visual Studio</li> </ul>



For More Information  
[www.coverity.com](http://www.coverity.com)  
 Email: [info@coverity.com](mailto:info@coverity.com)

Coverity Inc. Headquarters  
 185 Berry Street, Suite 6500  
 San Francisco, CA 94107 USA

U.S. Sales: (800) 873-8193  
 International Sales: +1 (415) 321-5237  
 Email: [sales@coverity.com](mailto:sales@coverity.com)