

Static Analysis (Coverity) Architecture Analysis

Synopsys Static Analysis Architecture Analysis provides the visibility to define a well-structured model and control code complexity

Product overview

Synopsys Static Analysis (Coverity) Architecture Analysis provides an architecture development environment that helps development teams organize the codebase files into a well-structured containment model. A well-structured model provides a modular hierarchy, keeps dependencies low and controlled, and can be enforced with automatic detection of rule violations to control code complexity. Maintaining a well-understood and well-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 that 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.

Common usage scenarios

Architect: Discover, define, publish, evolve

Synopsys Static Analysis Architecture Analysis will discover the architecture and map it to the actual implementation. Then the levelized structure map (LSM) enables the creation of well-structured containment models to control dependencies. Architects can easily expand and collapse the map to any level and manipulate the contents,

Improve code quality and maintainability

- Generate a comprehensive visual map of complex architectures.
- Create a leveled 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 the class, folder, or design level.
- Find unused code clusters at any level.

moving items from one container to another while viewing the impact on 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. Static Analysis Architecture Analysis will surface architectural violations within Coverity Connect for easy triage and remediation. It also integrates directly into the IDE, where developers can view architectural diagrams and address violations and refactoring tasks.

Manager: Monitor, enforce, assign

Static Analysis 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 Static Analysis 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 and 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 Synopsys Static Analysis platform have been addressed.

Platform support	Language support	IDE support
<ul style="list-style-type: none">• Linux• Windows	<ul style="list-style-type: none">• C, C++• C#• Java	<ul style="list-style-type: none">• Eclipse• Visual Studio

The Synopsys difference

Synopsys helps development teams build secure, high-quality software, minimizing risks while maximizing speed and productivity. Synopsys, a recognized leader in application security, provides static analysis, software composition analysis, and dynamic analysis solutions that enable teams to quickly find and fix vulnerabilities and defects in proprietary code, open source components, and application behavior. With a combination of industry-leading tools, services, and expertise, only Synopsys helps organizations optimize security and quality in DevSecOps and throughout the software development life cycle.

For more information, go to www.synopsys.com/software.

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

U.S. Sales: 800.873.8193
International Sales: +1 415.321.5237
Email: sig-info@synopsys.com