

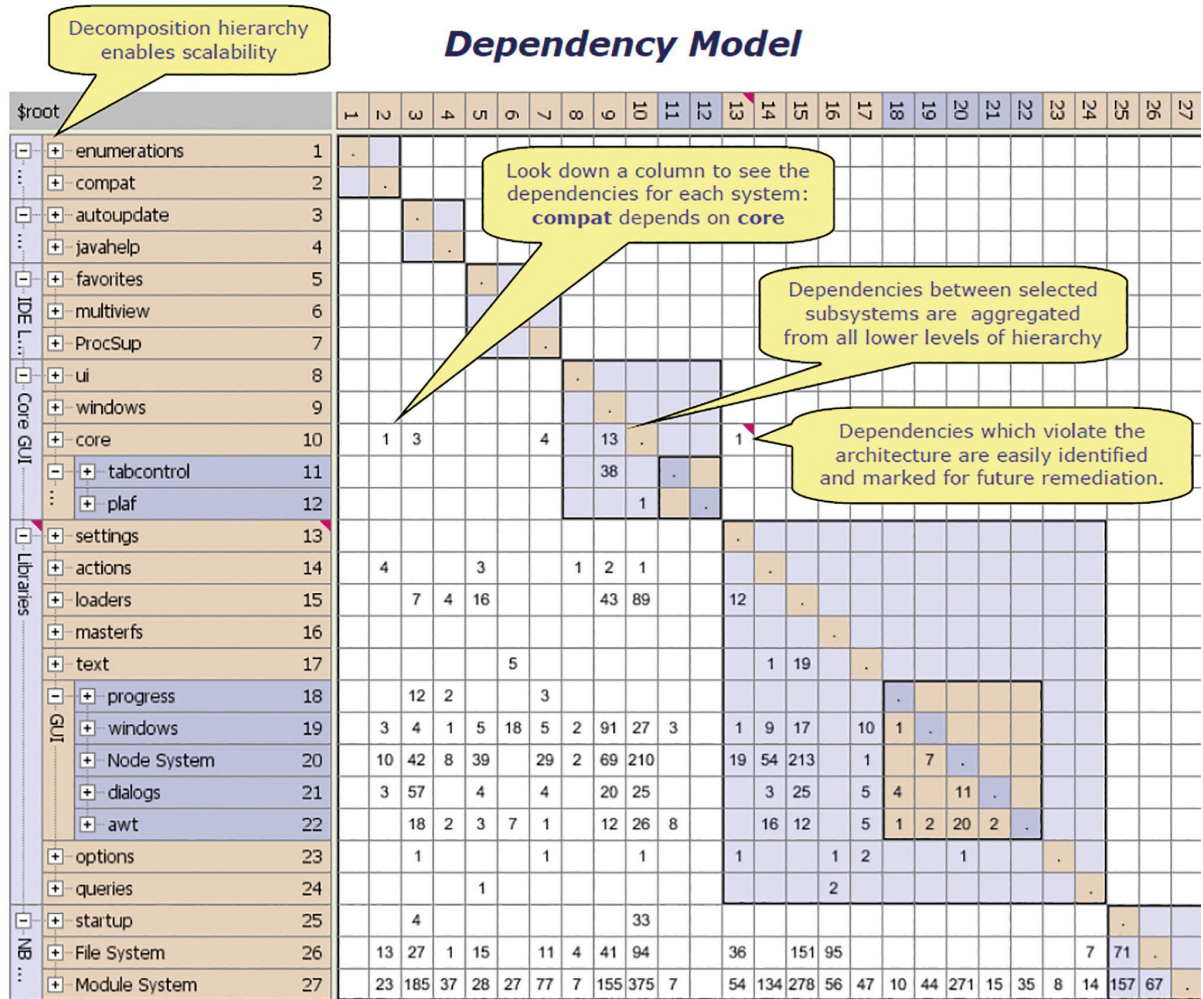
产品概述

Cantata Test Architect 帮助用户快速识别和修复软件架构问题，使用依赖结构矩阵（DSM，Dependency Structure Matrix）为开发经理和软件架构师提供一个快速可视化的方法展示应用程序的架构。在 DevOps 的开发、构建和测试的概念上，Cantata Test Architect 具有突出的能力：

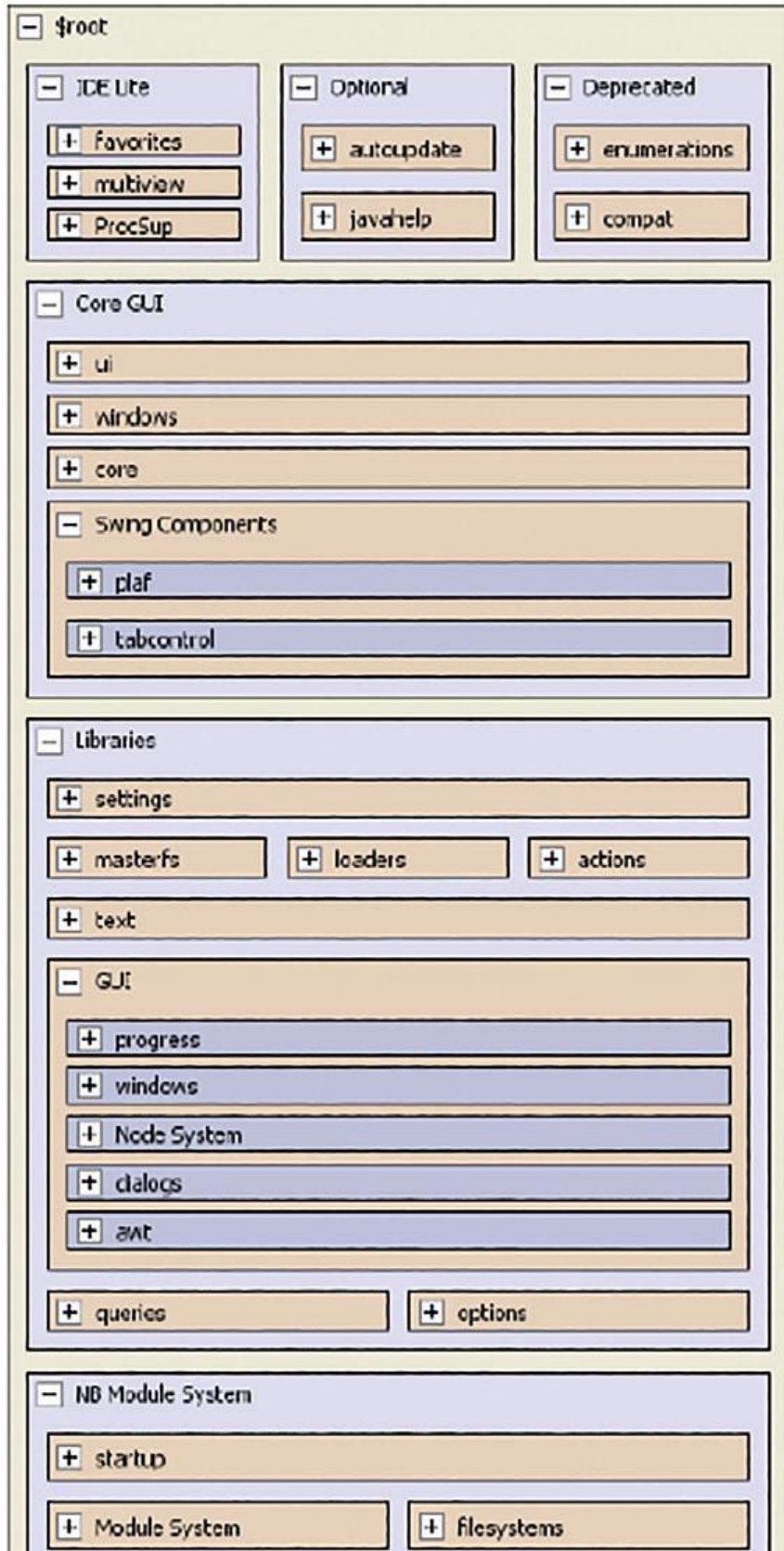
- 理解大型、复杂的软件架构
- 定义或重构架构
- 控制，指定和实施架构
- 持续架构分析

定义或重构架构

Cantata Test Architect 为软件长时间的架构侵蚀提供了强大的重构支持，通过灵活的架构编辑器、持续模型工具和代码重构算法等功能增强软件的模块化。使用架构编辑器，用户无需修改代码就能改变架构，而且所有改变都自动记忆在工作表（worklist）中。持续模型工具允许用户随着代码的改变持续更新模型，而 Test Architect 提供的算法帮助用户轻松地重构代码。



Conceptual Architecture

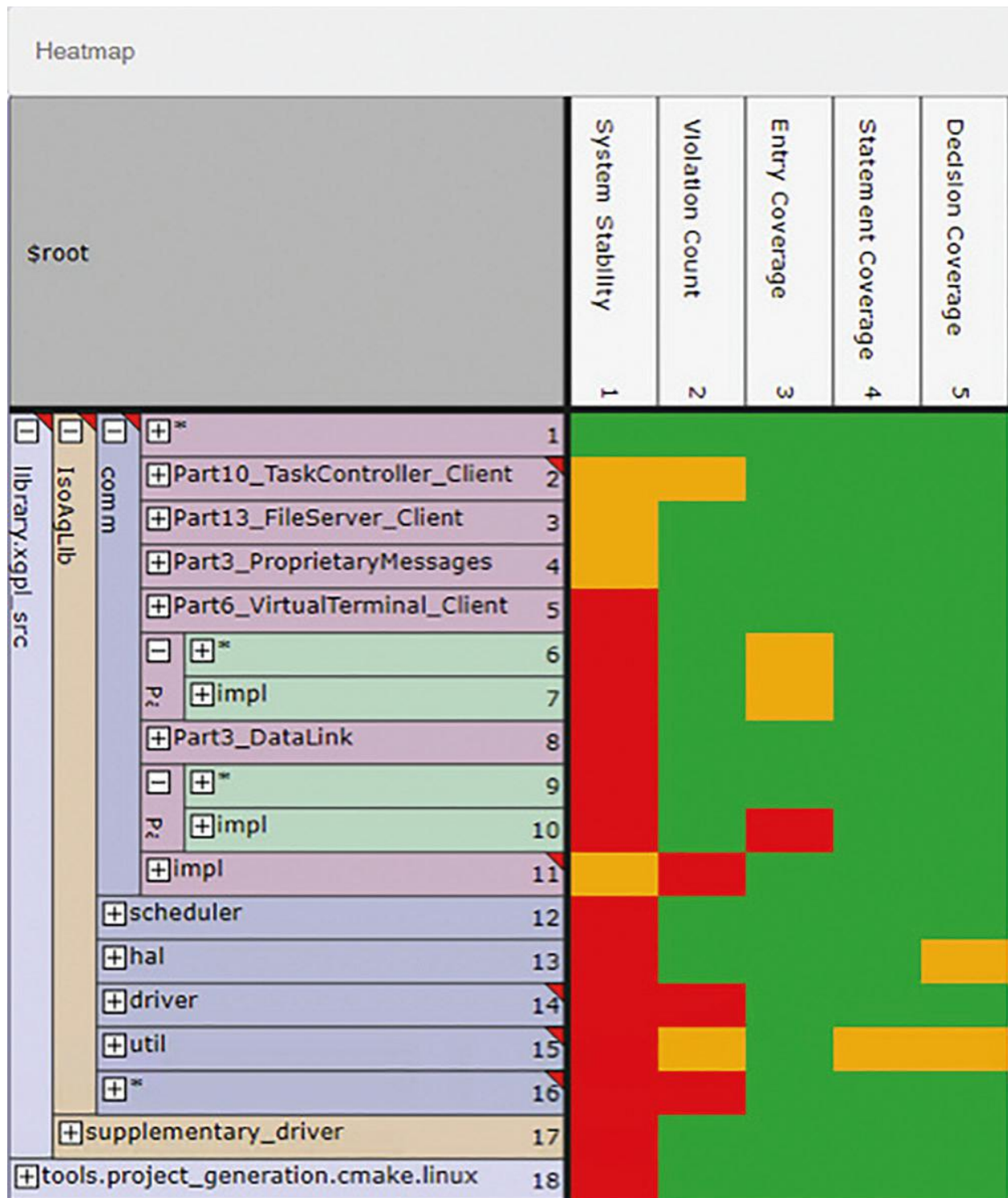


关键度量的热图显示

热图的引进是为了能够在代码架构的上下文中查看关键的度量信息。Cantata Test Architect 建立了架构的层次化视图，有层，有组件，位于 DSM 中，使用户能够从顶层视图深入挖掘到任何问题的源头。热图在显示这个结构时还带有所选度量的栅格，形成组合的视图，据此轻松了解问题出现在架构的什么地方。热图可以从桌面、Web 和 SonarQube 插件访问。

用户定义度量的阈值和对应颜色，评价和排序问题对架构的潜在影响。在架构层次上越深，问题的潜在影响越大。Cantata Test Architect 能够为那些可能被修复重构影响的架构部分建立测试。

通过与其他工具的集成使用，能够将架构分析结果结合静态分析结果（缺陷和规则违背）、测试结果（代码覆盖率和测试失败）、开源代码安全性（授权问题和漏洞）等，构成在各个开发阶段的代码质量的综合评估。



控制，指定和实施架构

Cantata Test Architect 使用规则来指定可被接受的多个子系统之间的关系，设计规则允许指定和讨论系统架构，并在整个团队中实施它。

- 创建内部和外部依赖的规则
- 指定分层规则
- 指定独立部件的规则
- 根据依赖的类型建立规则
- 根据源/目标子系统的类型建立规则

持续架构分析

LDC (Lattix 命令行) 应用可被用于自动化检查和更新依赖模型，它是我们 DevOps 和持续集成支持的核心内容。下面是 LDC 的一般应用：

- 在维护历史依赖时建立或更新依赖模型
- 产生多种格式的报告内容
- 在工程上执行定制化脚本
- 向依赖库发布工程或从依赖库抽取工程

LDC 自动分析每一次构建，让开发人员在产品发布之前发现和修复架构问题。对于大型复杂系统，分析其架构完整度对保证系统质量来说始终是至关重要的。

产品亮点

- 可视化历史应用的所有依赖关系
- 在架构的上下文中查看关键度量热图
- 自动发现过度复杂/架构缺陷
- 分区算法发现大型代码的结构
- 揭示所有使用的和被使用的元素
- 根据依赖类型进行过滤和理解
- 识别任何层次上的循环
- 发现不可达代码
- 深入部件、模块、类和函数