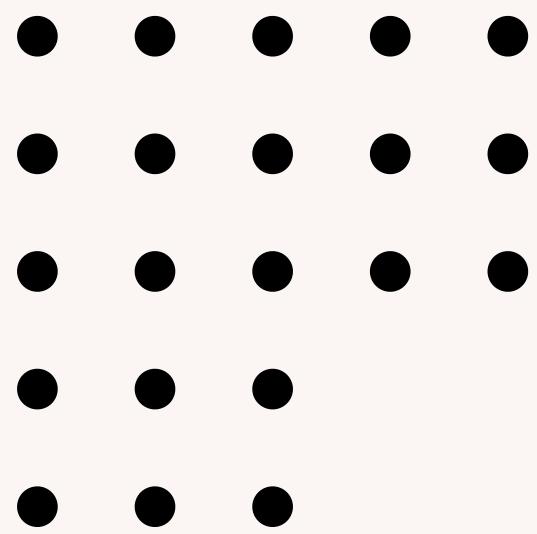


PROGRAMAÇÃO ORIENTADA A OBJETOS AVANÇADA: DESIGNITE



Henrique Coutinho
Felipe Kenji
Pedro Sakai
Vitoria Hilgert



FONTES

- **Documentação oficial do Designite:** <https://www.designite-tools.com/docs/designitejava.html>
- Fowler, Martin. "Code Smell." Martin Fowler's Bliki, 9 Fevereiro 2006, Disponível em: <https://martinfowler.com/bliki/CodeSmell.html>.
- Meszaros, Gerard. "Test Smells." xUnit Test Patterns: Refactoring Test Code, Addison-Wesley Professional, 2007.
- T. Sharma, P. Mishra, and R. Tiwari, "Designite - A Software Design Quality Assessment Tool," in 2016 IEEE/ACM 1st International Workshop on Bringing Architectural Design Thinking Into Developers' Daily Activities (BRIDGE), 2016, pp. 1-4.
- Link do GitHub para o Alibaba: <https://github.com/alibaba/spring-ai-alibaba>
- "Understanding Testability and Test Smells." Designite Tools Blog, <https://www.designite-tools.com/blog/understanding-testability-test-smells>.
- Aulas de métricas em POOA

INTRODUÇÃO

POR QUE UTILIZAR FERRAMENTAS PARA ANALISAR O CÓDIGO?

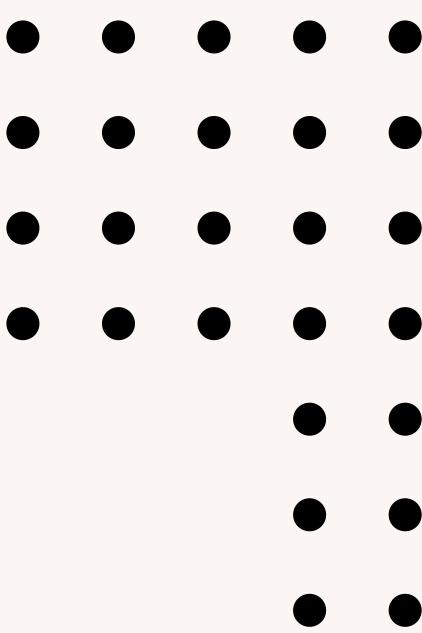
Códigos e arquiteturas limpas são essenciais para a **manutenibilidade e qualidade** de um projeto de software

- Ferramentas de análise tornam esse processo mais **objetivo e imparcial**.
- Ajudam a revelar problemas que os próprios desenvolvedores, por **viés**, podem não perceber.
- Tornam mais clara a necessidade de **melhoria contínua** no código.

SMELLS

DEFINIÇÃO

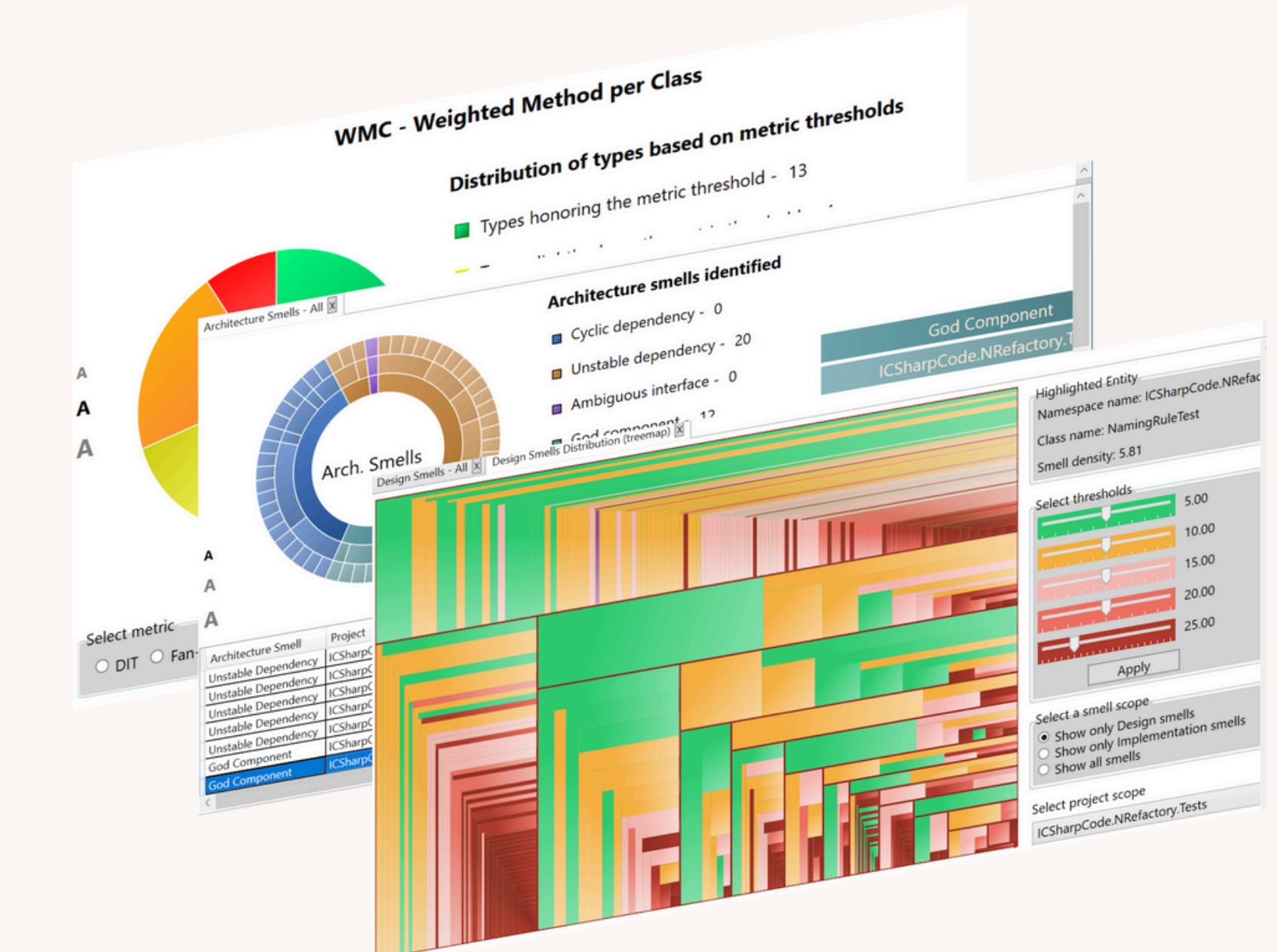
- São **sinais** que tem a **possibilidade de algo de errado com um trecho de código**, do ponto de vista estrutural ou de implementação, por exemplo.
- São diferentes de bugs, porque **a aplicação ainda funciona**.
- Indicam **potenciais problemas** com a manutenibilidade, legibilidade ou testes.



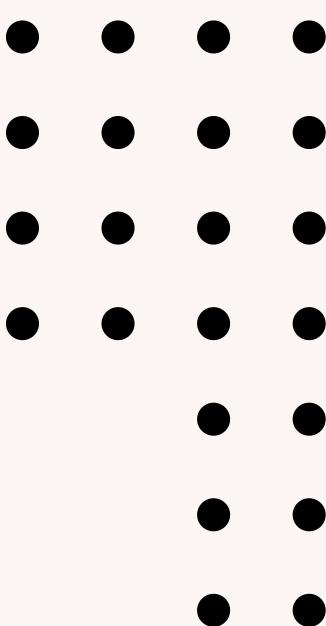
VISÃO GERAL

DESIGNITE

O que é: É uma ferramenta que oferece um **conjunto de análises estatísticas** com objetivo de facilitar a **identificação de possíveis problemas no código** e, assim, promover melhoria na qualidade das aplicações.



TIPOS DE SMELLS DETECTADOS



O Designite divide suas detecções em **cinco categorias de smells**:

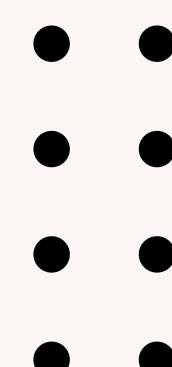
IMPLEMENTATION SMELLS

São smells relacionados à problemas na implementação que dificultam a legibilidade e a manutenção.

Exemplos: Declaração Longa, Condicional Complexa, Lista de Parâmetros Longa

O que o Designite informa sobre os Implementation Smells:

- Tipo do smell
- Nome do pacote, da classe e do **método**
- Descrição



TIPOS DE SMELLS DETECTADOS

DESIGN SMELLS

São smells relacionados à problemas estruturais que violam princípios de design/projetos orientados a objetos. São sub-categorizados em: smells de abstração, encapsulamento, modularização e hierarquização.

Exemplos: Abstração não Utilizada, Encapsulamento Deficiente, Falta de Modularização, Hierarquia Cíclica

O que o Designite informa sobre os Design Smells:

- Tipo do smell
 - Nome do pacote e da **classe**
 - Descrição
-

TIPOS DE SMELLS DETECTADOS

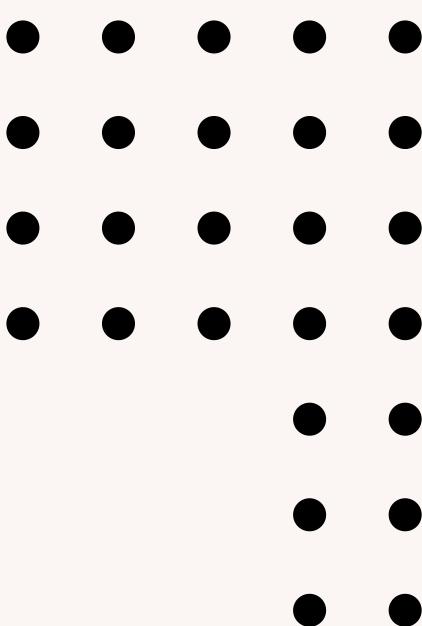
ARCHITECTURE SMELLS

São smells relacionados à problemas na **estruturação arquitetural** do projeto.

Exemplos: Dependência Instável, Dependência Cíclica, Componente Deus

O que o Designite informa sobre os Architecture Smells:

- Tipo do smell
- Nome do pacote
- Descrição



TIPOS DE SMELLS DETECTADOS

TEST SMELLS

São smells relacionados à uso de más práticas de programação que indiquem potenciais problemas de design no **código de teste**.

Exemplos: Roleta de Verificação, Tratamento de Exceção

O que o Designite informa sobre os Test Smells:

- Tipo do smell
- Nome do pacote, da classe e do **método**
- Descrição

TIPOS DE SMELLS DETECTADOS

TESTABILITY SMELLS

São smells relacionados a identificar escolhas do projeto que podem dificultar o dese de casos de teste:

Exemplos: Estado Global, Excesso de Dependências

O que o Designite informa sobre os Testability Smells:

- Tipo do smell
- Nome do pacote e da classe
- Descrição

TIPOS DE SMELLS DETECTADOS

VISÃO GERAL

Tipos de smells do Designite e o nível de cada um:

- **Implementation Smells:** nível de método
- **Design Smells:** nível de classe
- **Architecture Smells:** nível de pacote/módulo
- • **Test Smells:** nível de método
- • **Testability Smells:** nível de classe
- •
- • • •
- • • •
- • • •
- • • •

MÉTRICAS DETECTADAS

Além da detecção de smells em diferentes níveis, o Designite também oferece a detecção de **12 métricas de projeto orientado a objetos**:

SIGLA	NOME COMPLETO	DEFINIÇÃO
LOC	Lines of Code	Quantidade de linhas de código em um método ou classe
CC	Complexidade Ciclomática	Quantidade de caminhos de execução diferentes do método
PC	Contagem de Parâmetros	Quantidade de parâmetros do método

MÉTRICAS DETECTADAS

SIGLA	NOME COMPLETO	DEFINIÇÃO
NOF	Number of Fields	Quantidade de variáveis de uma classe
NOPF	Number of Public Fields	Quantidade de variáveis públicas de uma classe
NOM	Number of Methods	Quantidade de métodos de uma classe
NOPM	Number of Public Methods	Quantidade de métodos públicos de uma classe

MÉTRICAS DETECTADAS

SIGLA	NOME COMPLETO	DEFINIÇÃO
WMC	Weighted Methods per Class	A soma de todas as complexidades ciclomáticas dos métodos da classe
NC	Number of Children	Número de subclasses
DIT	Depth of Inheritance Tree	Caminho máximo de herança para a classe raiz

MÉTRICAS DETECTADAS

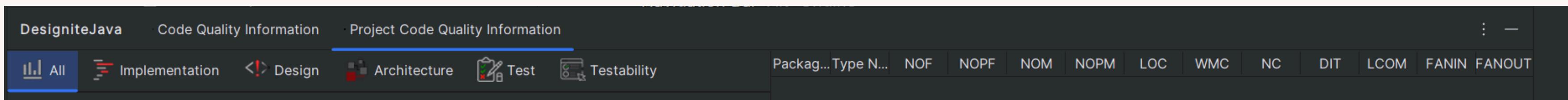
SIGLA	NOME COMPLETO	DEFINIÇÃO
LCOM	Lack of Cohesion in Methods	Falta de Coesão entre os Métodos. Correlação entre métodos e as variáveis que estes modificam.
FANIN	Fan-In	Acoplamentos aferentes
FANOUT	Fan-Out	Acoplamentos eferentes

FUNCIONAMENTO NO DESIGNITE

COMO O DESIGNITE ORGANIZA AS INFORMAÇÕES

Para o Java, o Designite é um plugin que pode ser instalado e adicionado no IntelliJIDEA.

O Designite aparece como um ícone no canto esquerdo inferior da tela e, ao ser clicado, abre uma aba com as visualizações.



FUNCIONAMENTO NO DESIGNITE

EXPORTAÇÕES

O Designite também fornece exportações em .csv, gerando uma tabela para as métricas e uma para cada tipo de smell.

Project Name	Package Name	Type Name	Method Name	Implementation Smell	Cause of the Smell
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	RetrievalRerankAdvisorTests	setUp	Long Statement	The length of the statement "testRequest=ChatC
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	RetrievalRerankAdvisorTests	testAroundCallWithDocuments	Long Statement	The length of the statement "ChatClientRespons
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	RetrievalRerankAdvisorTests	testAroundCallWithDocuments	Long Statement	The length of the statement "assertThat(respons
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	RetrievalRerankAdvisorTests	testAroundCallWithDocuments	Magic Number	The method contains a magic number: 0.8
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	RetrievalRerankAdvisorTests	testAroundCallWithEmptyDocuments	Long Statement	The length of the statement "ChatClientRespons
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	RetrievalRerankAdvisorTests	testAroundCallWithEmptyDocuments	Long Statement	The length of the statement "assertThat(respons
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	RetrievalRerankAdvisorTests	testAroundStream	Magic Number	The method contains a magic number: 0.8
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	RetrievalRerankAdvisorTests	testFilterExpression	Long Statement	The length of the statement "ChatClientRequest
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	RetrievalRerankAdvisorTests	testFilterExpression	Long Statement	The length of the statement "ChatClientRespons
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	RetrievalRerankAdvisorTests	testScoreFiltering	Long Statement	The length of the statement "ChatClientRespons
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	RetrievalRerankAdvisorTests	testScoreFiltering	Long Statement	The length of the statement "assertThat(respons
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	RetrievalRerankAdvisorTests	testScoreFiltering	Magic Number	The method contains a magic number: 0.9
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	RetrievalRerankAdvisorTests	testScoreFiltering	Magic Number	The method contains a magic number: 0.8
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	DocumentRetrievalAdvisorTests	setUp	Long Statement	The length of the statement "testRequest=ChatC
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	DocumentRetrievalAdvisorTests	testAroundCallWithDocuments	Long Statement	The length of the statement "ChatClientRespons
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	DocumentRetrievalAdvisorTests	testAroundCallWithEmptyDocuments	Long Statement	The length of the statement "ChatClientRespons
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	DocumentRetrievalAdvisorTests	testAroundStream	Long Statement	The length of the statement "ChatClientRespons
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	DocumentRetrievalAdvisorTests	testMetadataHandling	Long Statement	The length of the statement "ChatClientRespons
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	RetrievalRerankAdvisor	RetrievalRerankAdvisor	Long Parameter List	The method has 5 parameters.
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	RetrievalRerankAdvisor	RetrievalRerankAdvisor	Long Parameter List	The method has 6 parameters.
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	RetrievalRerankAdvisor	doRerank	Long Statement	The length of the statement "return response.get
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	RetrievalRerankAdvisor	before	Long Statement	The length of the statement "var searchRequestI
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	RetrievalRerankAdvisor	before	Long Statement	The length of the statement "String augmentedU
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	RetrievalRerankAdvisor	after	Long Statement	The length of the statement "return ChatClientRe
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	DocumentRetrievalAdvisor	before	Long Statement	The length of the statement "String augmentedU
spring-ai-alibaba-core	com.alibaba.cloud.ai.advisor	DocumentRetrievalAdvisor	after	Long Statement	The length of the statement "return ChatClientRe
spring-ai-alibaba-core	com.alibaba.cloud.ai.transformer	SentenceSplitterTests	testSplitLongText	Long Statement	The length of the statement "String longSentence
spring-ai-alibaba-core	com.alibaba.cloud.ai.transformer	SentenceSplitterTests	testSplitLongText	Magic Number	The method contains a magic number: 50
spring-ai-alibaba-core	com.alibaba.cloud.ai.document	DocumentWithScoreTests	testBuilderAndGetters	Long Statement	The length of the statement "DocumentWithScor
spring-ai-alibaba-core	com.alibaba.cloud.ai.document	DocumentWithScoreTests	testBuilderAndGetters	Magic Number	The method contains a magic number: 0.95

ANÁLISE DESIGNITE

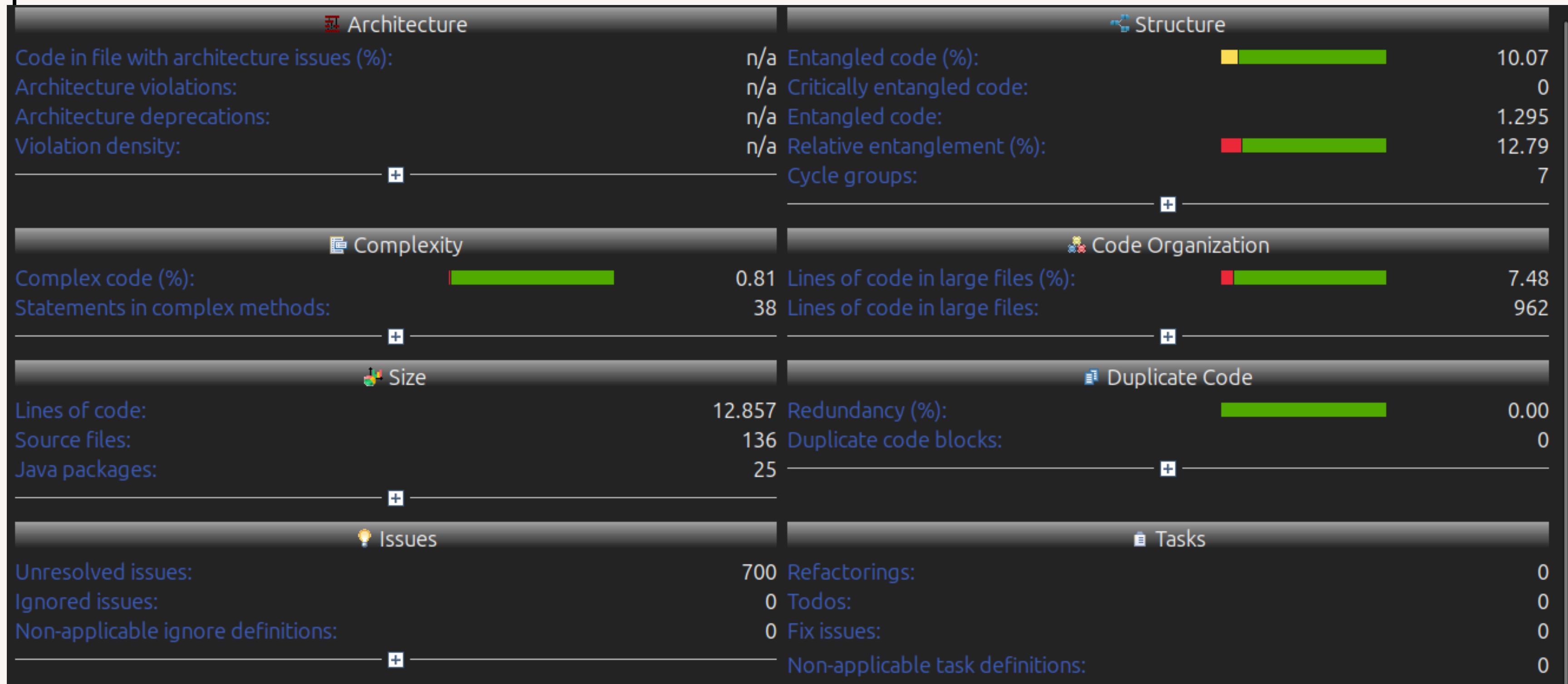
Projeto utilizado: [Spring AI Alibaba Core](#) (modo de teste não suporta projetos grandes)

The screenshot shows the GitHub repository page for `spring-ai-alibaba`. The repository is public and has 57 watchers, 749 forks, and 4k stars. It contains 9 branches and 9 tags. The main branch is selected. The repository is described as an "Agentic AI Framework for Java Developers" maintained by `java2ai.com`. It uses Java, artificial-intelligence, and spring-ai technologies. Key repository details include the Apache-2.0 license, code of conduct, security policy, activity, custom properties, and reporting. The repository has 2,186 commits, with the most recent being a fix for deepresearch by GTyingzi. Other commits include fixes for nl2sql, style improvements, and feature additions for MCP and Nacos.

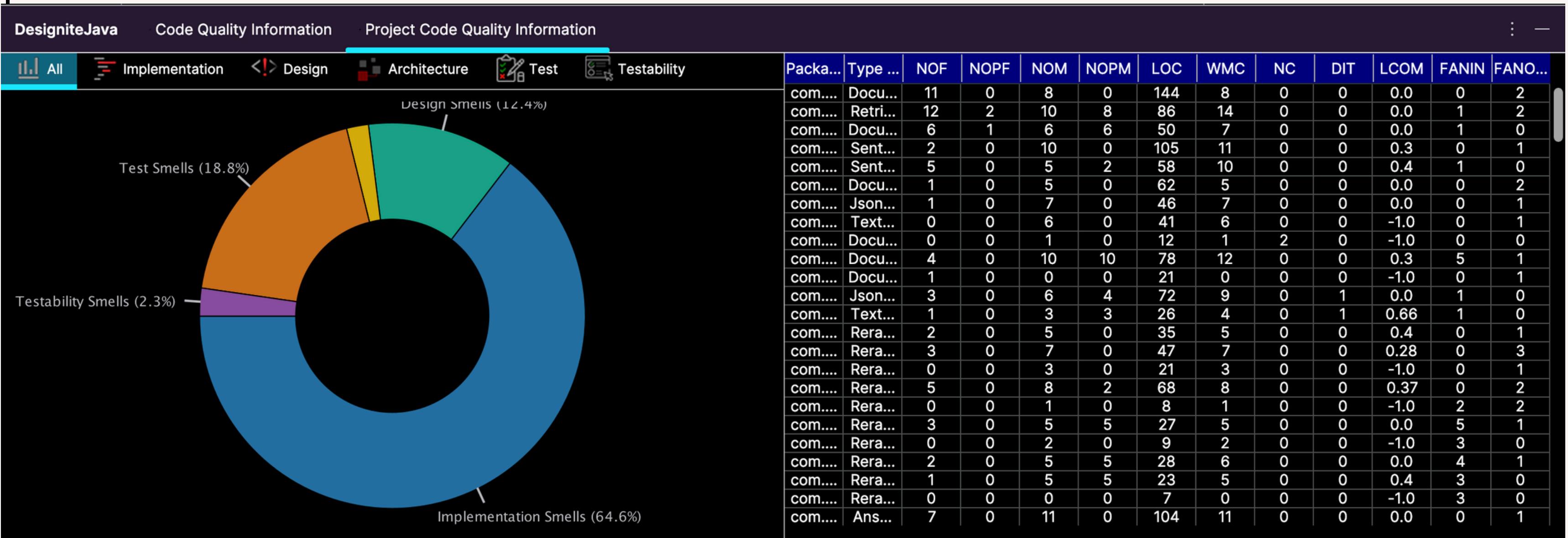
Commit	Message	Time Ago
GTYingzi fix(deepresearch): 修复research、coder的一些问题 (#1161)	1b903ff · 7 minutes ago	2,186 Commits
.github	fix(nl2sql): add nl2sql ci (#1126)	2 days ago
.mvn/wrapper	style: add newline check makefil target	last month
auto-configurations	feat(mcp): adjust mcp-nacos3 (#1092)	5 days ago
community	fix(deepsearch) : Add the retry logic for the temporarily u...	5 hours ago
docs/imgs	docs(core): update readme (#1114)	4 days ago
spring-ai-alibaba-bom	fix(mcp): fix mcp dep and config metadata (#1080)	last week
spring-ai-alibaba-core	refactor: Refactored the uploadFile method (#1066)	4 days ago
spring-ai-alibaba-deepresearch	fix(deepresearch): 修复research、coder的一些问题 (#1161)	7 minutes ago
spring-ai-alibaba-graph	feat(graph): add variableAggregatorNode and unit test (...)	20 hours ago
spring-ai-alibaba-jmanus	fix(jmanus): 步骤执行详情部分交互优化 (#1160)	30 minutes ago
spring-ai-alibaba-mcp	feat(mcp): adjust mcp-nacos3 (#1092)	5 days ago
spring-ai-alibaba-nl2sql	feat(nl2sql): add simpleVectorStore nl2sql (#1148)	7 hours ago

ANÁLISE SONARGRAPH

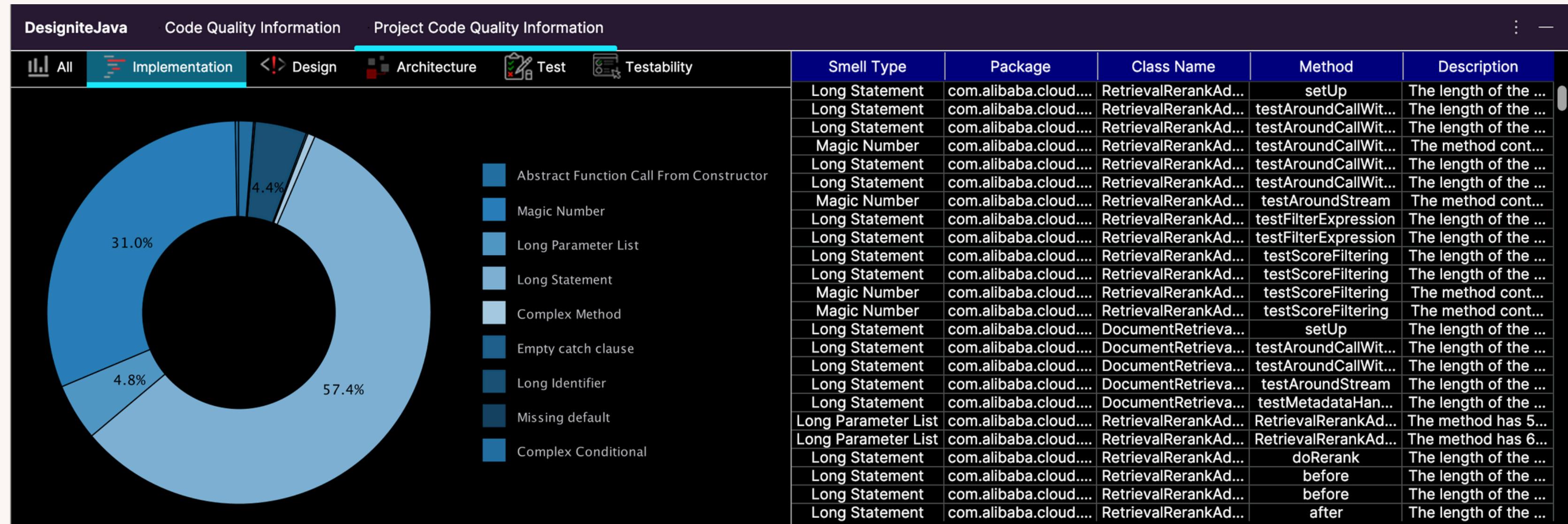
Métricas do sistema no SonarGraph Explorer



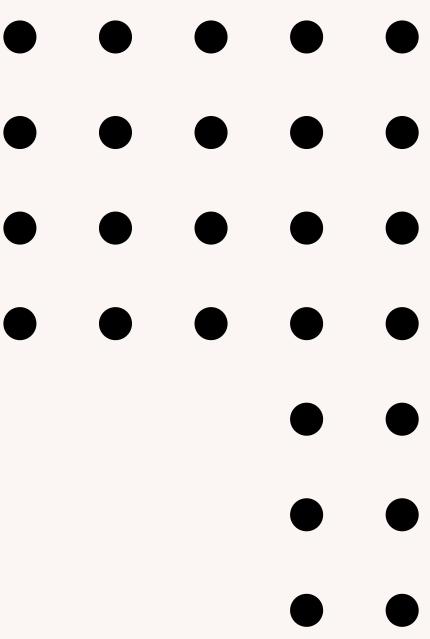
ANÁLISE COMPLETA



ANÁLISE IMPLEMENTAÇÃO



Quase 90% do código é Long Statement e Magic Number.
O que significa?



ANÁLISE IMPLEMENTAÇÃO

- **ABSTRACT FUNCTION CALL FROM CONSTRUCTOR**

- O QUE É: CHAMAR MÉTODO INSTÁVEL NO CONSTRUTOR.
- PROBLEMA: COMPORTAMENTO IMPREVISÍVEL, OBJETO MAL CONSTRUÍDO.
- SOLUÇÃO: USAR UM MÉTODO DE INICIALIZAÇÃO APÓS A CRIAÇÃO.

- **LONG STATEMENT (DECLARAÇÃO LONGA)**

- O QUE É: UMA ÚNICA LINHA DE CÓDIGO MUITO COMPRIDA.
- PROBLEMA: DIFÍCULTA A LEITURA E O ENTENDIMENTO DA LÓGICA.
- SOLUÇÃO: QUEBRAR EM VÁRIAS LINHAS; USAR VARIÁVEIS AUXILIARES.

- **MAGIC NUMBER (NÚMERO MÁGICO)**

- O QUE É: NÚMEROS SOLTOS NO CÓDIGO, SEM EXPLICAÇÃO (EX: STATUS == 2).
- PROBLEMA: CÓDIGO SEM SENTIDO, MAIS DIFÍCIL DE MANTER.
- SOLUÇÃO: SUBSTITUIR POR CONSTANTES NOMEADAS (EX: STATUS_ATIVO).

ANÁLISE IMPLEMENTAÇÃO

- **LONG PARAMETER LIST (LISTA DE PARÂMETROS LONGA)**

- O QUE É: MÉTODO QUE RECEBE ARGUMENTOS DEMAIS.
- PROBLEMA: DIFÍCIL DE USAR; SINAL DE QUE O MÉTODO FAZ MUITO.
- SOLUÇÃO: AGRUPAR PARÂMETROS EM UM ÚNICO OBJETO.

- **COMPLEX METHOD (MÉTODO COMPLEXO)**

- O QUE É: MÉTODO COM EXCESSO DE IF, ELSE, FOR.
- PROBLEMA: DIFÍCIL DE TESTAR E ENTENDER; ALTA CHANCE DE BUGS.
- SOLUÇÃO: QUEBRAR EM MÉTODOS MENORES E COM UMA SÓ RESPONSABILIDADE.

- **EMPTY CATCH CLAUSE (CATCH VAZIO)**

- O QUE É: UM BLOCO CATCH QUE NÃO FAZ NADA COM O ERRO.
- PROBLEMA: ESCONDE FALHAS CRÍTICAS DO SISTEMA.
- SOLUÇÃO: SEMPRE REGISTRAR (LOGAR) OU TRATAR A EXCEÇÃO.

ANÁLISE IMPLEMENTAÇÃO

• LONG IDENTIFIER (IDENTIFICADOR LONGO)

- O QUE É: NOMES DE VARIÁVEIS OU MÉTODOS LONGOS DEMAIS.
- PROBLEMA: POLUI O CÓDIGO E PREJUDICA A LEGIBILIDADE.
- SOLUÇÃO: USAR NOMES MAIS CONCISOS, MAS AINDA CLAROS.

• MISSING DEFAULT (DEFAULT FALTANTE)

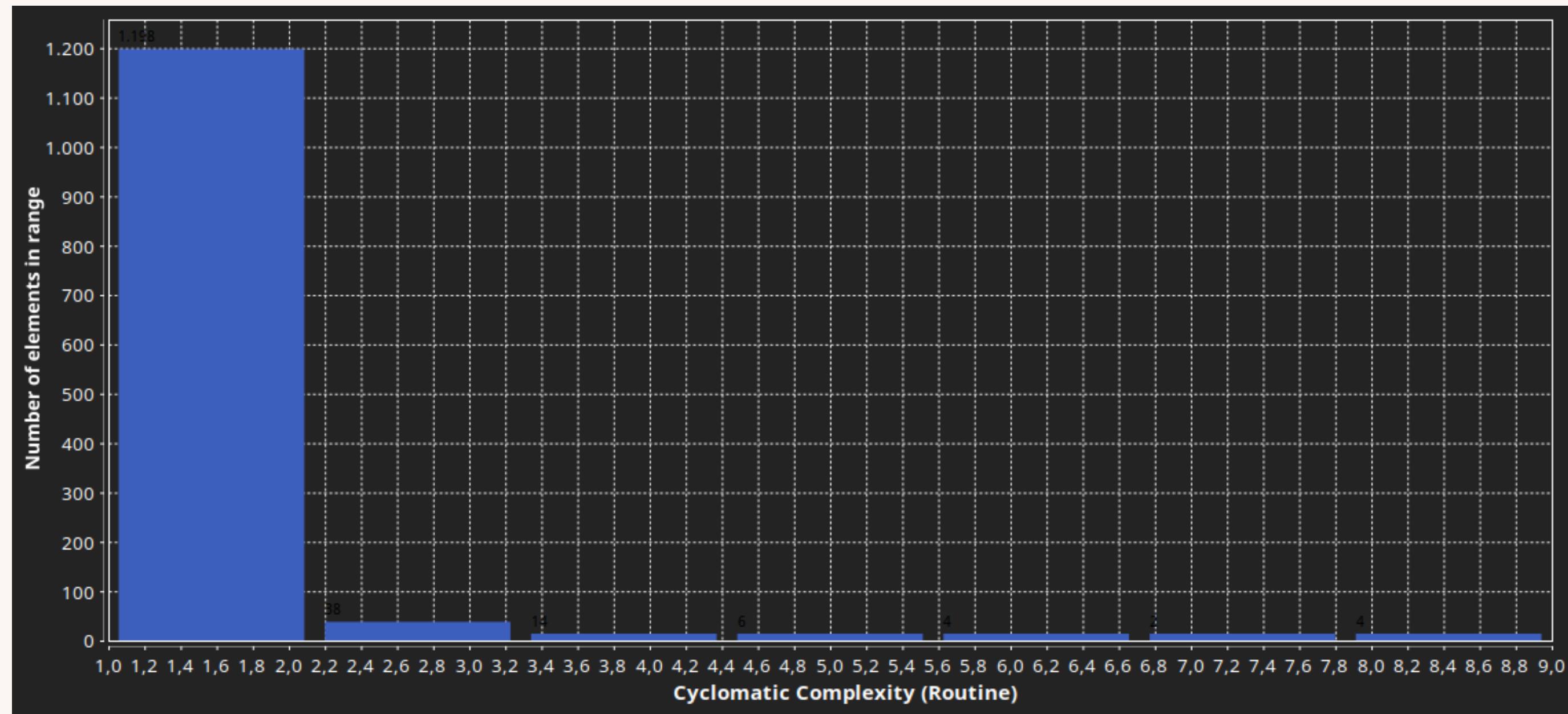
- O QUE É: UM SWITCH SEM A CLÁUSULA DEFAULT.
- PROBLEMA: FALHA SILENCIOSA QUANDO SURGEM NOVOS CASOS NÃO TRATADOS.
- SOLUÇÃO: SEMPRE ADICIONAR UM DEFAULT PARA TRATAR VALORES INESPERADOS.

• COMPLEX CONDITIONAL (CONDICIONAL COMPLEXA)

- O QUE É: UM IF COM MUITAS CONDIÇÕES (&&, ||).
- PROBLEMA: DIFÍCIL DE LER E ENTENDER A REGRA DE NEGÓCIO.
- SOLUÇÃO: MOVER A LÓGICA PARA UM MÉTODO COM NOME DESCRIPTIVO.

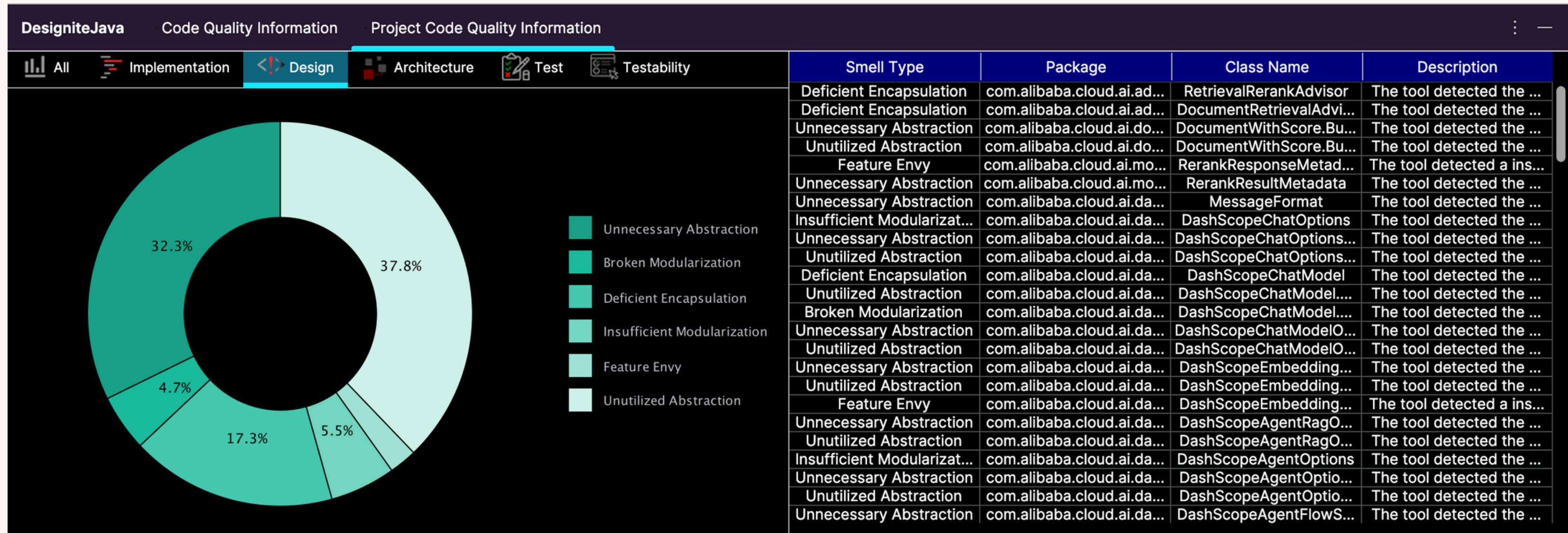
ANÁLISE IMPLEMENTAÇÃO

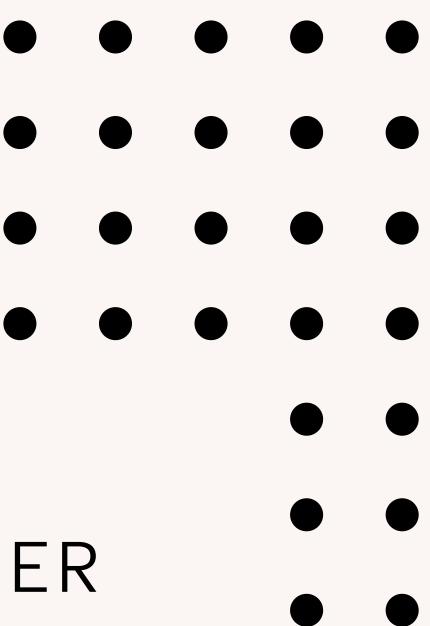
A métrica de **complexidade ciclomática** se relaciona com o **Complex Method**



A **complexidade ciclomática** mede a complexidade de todos os métodos, enquanto que o **Complex Method**, os métodos cuja medição ultrapassou um limite de complexidade aceitável

ANÁLISE DESIGN





ANÁLISE DESIGN

• UNNECESSARY ABSTRACTION (ABSTRAÇÃO DESNECESSÁRIA)

- O QUE É: INTERFACE OU CLASSE ABSTRATA QUE COMPLICA SEM TRAZER BENEFÍCIO.
- PROBLEMA: AUMENTA A COMPLEXIDADE E DIFICULTA O ENTENDIMENTO.
- SOLUÇÃO: SIMPLIFICAR A HIERARQUIA, REMOVENDO ABSTRAÇÕES INÚTEIS.

• UNUTILIZED ABSTRACTION (ABSTRAÇÃO NÃO UTILIZADA)

- O QUE É: UMA INTERFACE OU CLASSE ABSTRATA QUE NINGUÉM USA.
- PROBLEMA: CÓDIGO MORTO QUE POLUI E CONFUNDE.
- SOLUÇÃO: REMOVER O CÓDIGO NÃO UTILIZADO.

• FEATURE ENVY (INVEJA DE FUNCIONALIDADE)

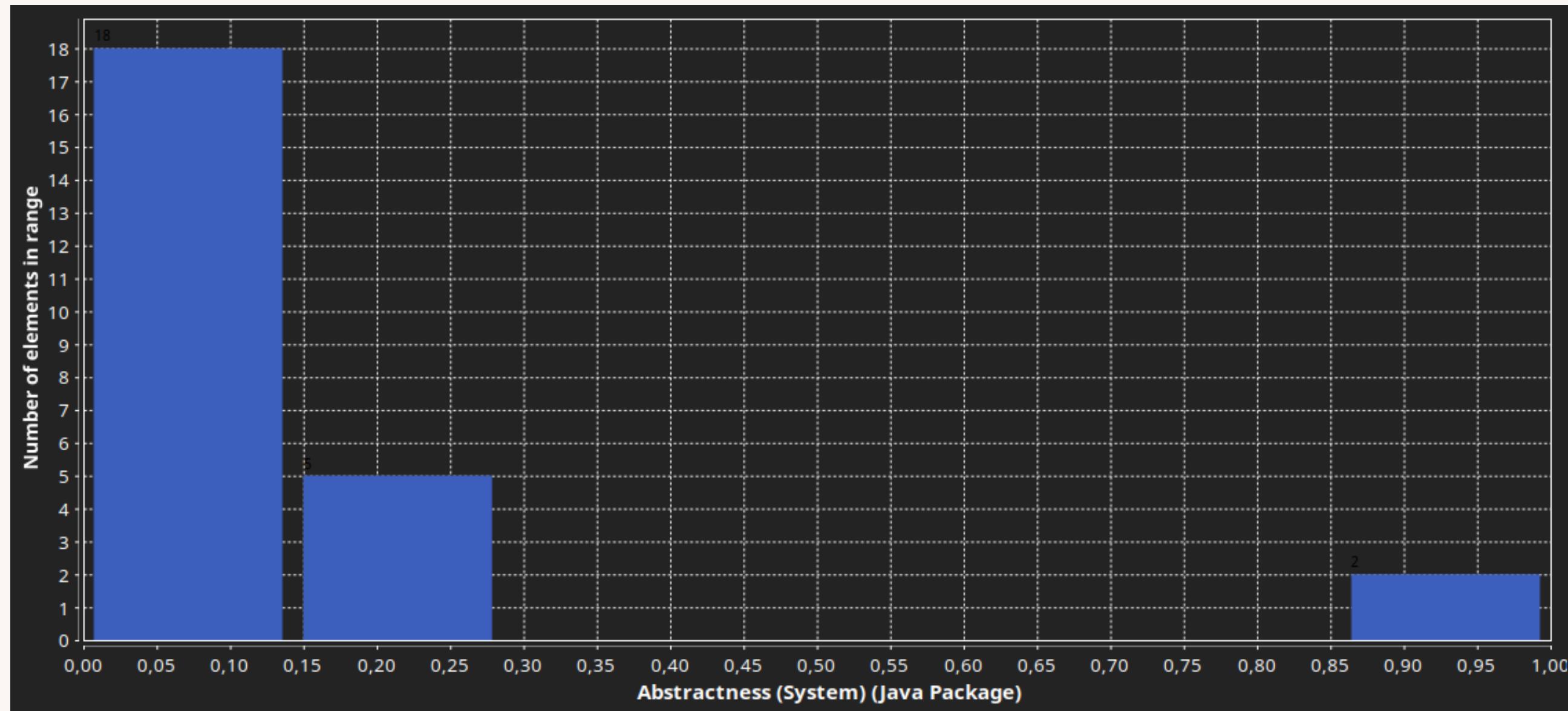
- O QUE É: MÉTODO QUE USA MAIS DADOS DE OUTRA CLASSE DO QUE DA SUA PRÓPRIA.
- PROBLEMA: RESPONSABILIDADE NO LUGAR ERRADO; ALTO ACOPLAGEMTO.
- SOLUÇÃO: MOVER O MÉTODO PARA A CLASSE QUE ELE "INVEJA".

ANÁLISE DESIGN

- **INSUFFICIENT MODULARIZATION (MODULARIZAÇÃO INSUFICIENTE)**
 - O QUE É: CLASSE COM MUITAS RESPONSABILIDADES NÃO RELACIONADAS.
 - PROBLEMA: PRECURSORA DA "GOD CLASS"; DIFÍCIL DE MANTER E TESTAR.
 - SOLUÇÃO: DIVIDIR A CLASSE EM COMPONENTES MENORES E MAIS COESOS.
- **DEFICIENT ENCAPSULATION (ENCAPSULAMENTO DEFICIENTE)**
 - O QUE É: EXPOR DADOS INTERNOS DA CLASSE (EX: CAMPOS PÚBLICOS).
 - PROBLEMA: QUEBRA A PROTEÇÃO DOS DADOS; MUDANÇAS SE TORNAM ARRISCADAS.
 - SOLUÇÃO: MANTER CAMPOS PRIVADOS E CONTROLAR O ACESSO COM MÉTODOS.
- **BROKEN MODULARIZATION (MODULARIZAÇÃO QUEBRADA)**
 - O QUE É: MÓDULOS QUE SÃO MUITO DEPENDENTES UNS DOS OUTROS.
 - PROBLEMA: UMA MUDANÇA EM UM LUGAR CAUSA UM EFEITO CASCATA EM OUTROS.
 - SOLUÇÃO: REDUZIR O ACOPLAMENTO E REDEFINIR AS FRONTEIRAS ENTRE OS MÓDULOS.

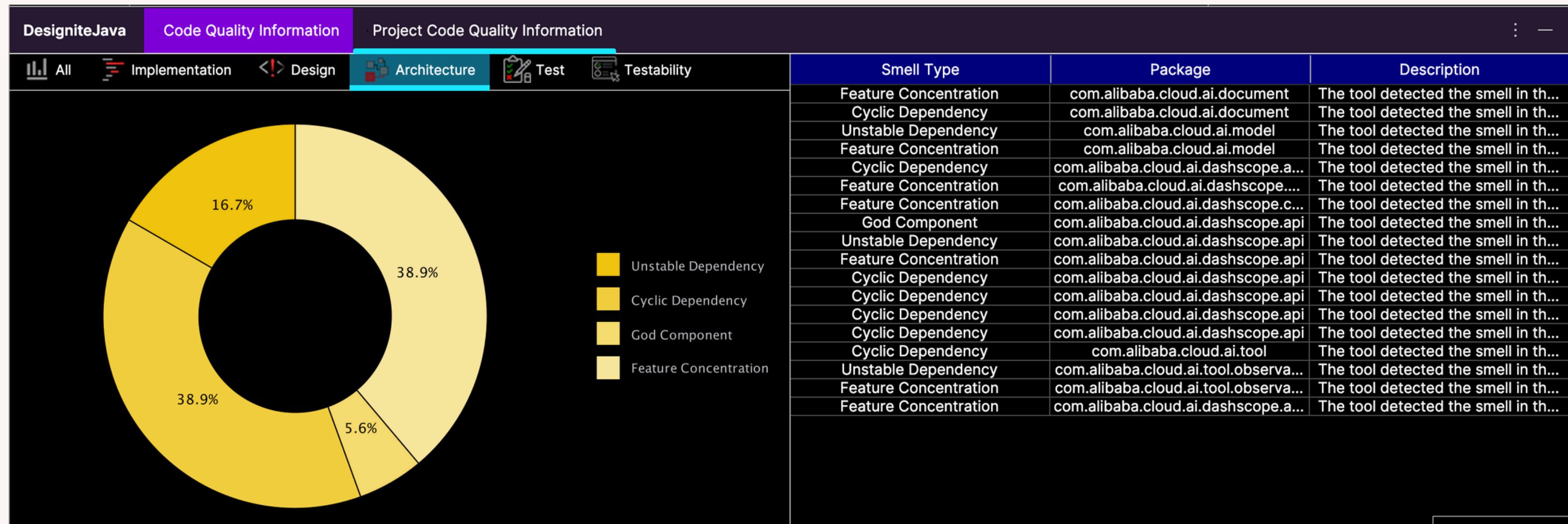
ANÁLISE DESIGN

A métrica de **abstração** se relaciona com o **Unnecessary Abstraction**

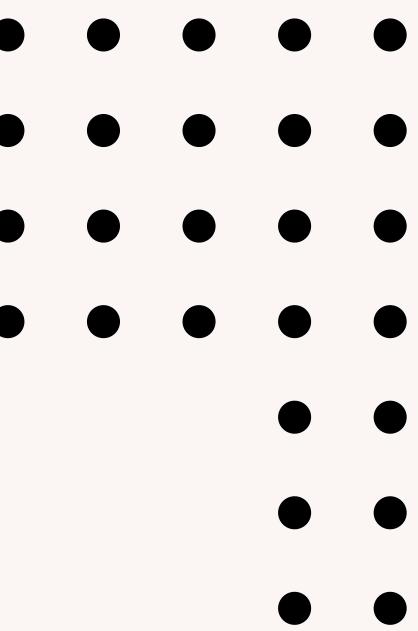


A **Abstração** mede o quanto abstrato é um pacote, enquanto que o **Unnecessary Abstraction**, as interfaces ou classes abstratas que foram criadas sem uma necessidade real, como ter apenas uma única implementação.

ANÁLISE ARQUITETURA



Quase 80% dos problemas se dividem entre Cyclic Dependency e Feature Concentration.
O sistema está emaranhado!!!



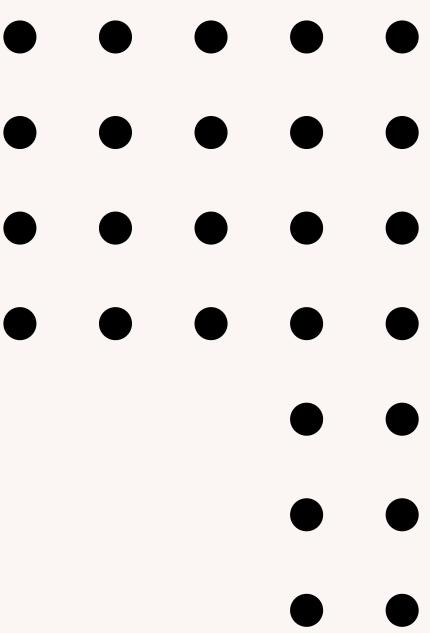
ANÁLISE ARQUITETURA

- **CYCLIC DEPENDENCY (DEPENDÊNCIA CÍCLICA)**

- O QUE É: PACOTES QUE DEPENDEM UNOS DOUTROS EM UM CICLO ($A \leftrightarrow B$).
- PROBLEMA: IMPOSSÍVEL ISOLAR, TESTAR OU MANTER OS PACOTES; SISTEMA "AMARRADO".
- SOLUÇÃO: USAR INTERFACES PARA INVERTER E QUEBRAR A DEPENDÊNCIA CIRCULAR.

- **FEATURE CONCENTRATION (CONCENTRAÇÃO DE FUNCIONALIDADE)**

- O QUE É: A MAIOR PARTE DA LÓGICA DO SISTEMA ESTÁ CONCENTRADA EM UM SÓ PACOTE.
- PROBLEMA: PACOTE VIRA UM GARGALO DE DESENVOLVIMENTO; BAIXA COESÃO.
- SOLUÇÃO: REDISTRIBUIR RESPONSABILIDADES PARA OUTROS PACOTES.



ANÁLISE ARQUITETURA

- **UNSTABLE DEPENDENCY (DEPENDÊNCIA INSTÁVEL)**

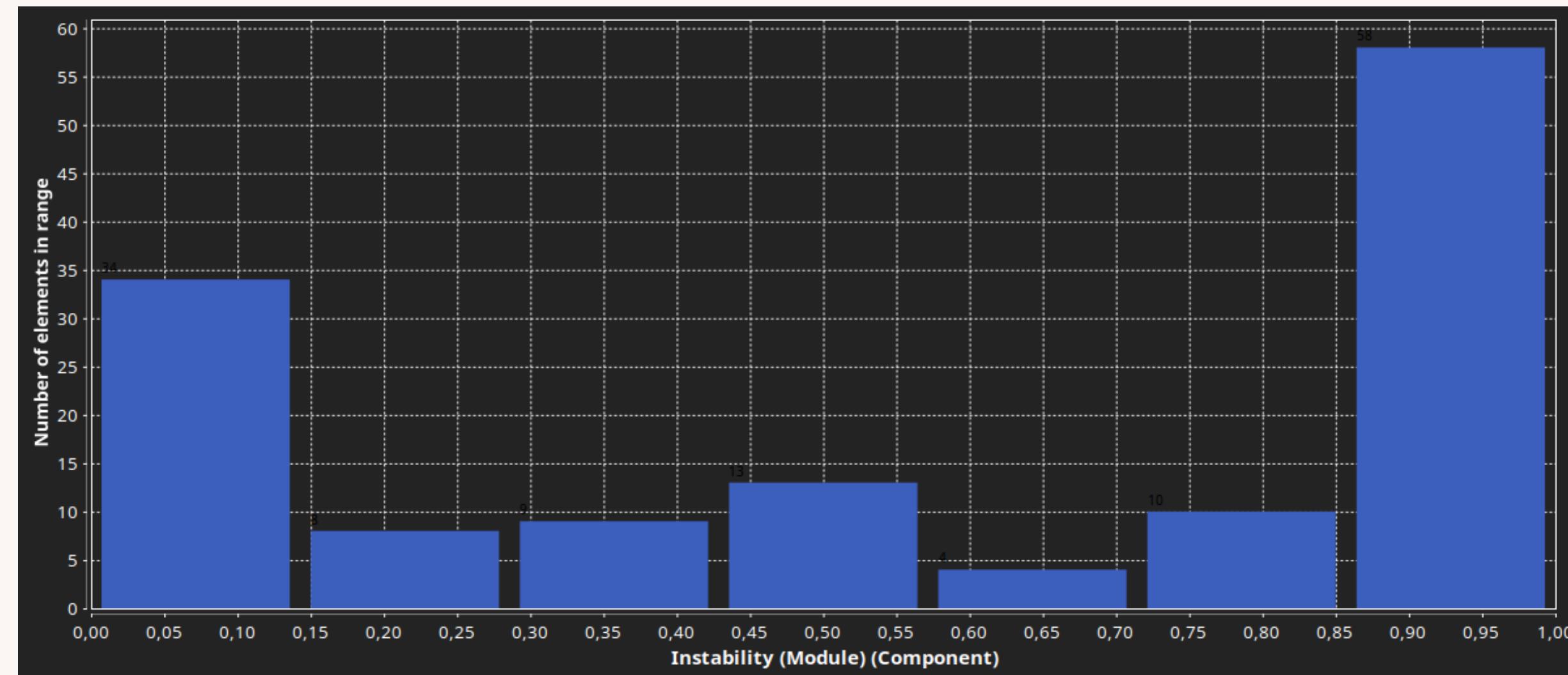
- O QUE É: UM PACOTE ESTÁVEL (QUE NÃO DEVERIA MUDAR) DEPENDE DE UM INSTÁVEL (QUE MUDA MUITO).
- PROBLEMA: DEIXA A ARQUITETURA FRÁGIL E PROPENSA A QUEBRAS EM CASCATA.
- SOLUÇÃO: INVERTER A DEPENDÊNCIA COM UMA ABSTRAÇÃO (INTERFACE).

- **GOD COMPONENT (COMPONENTE DEUS)**

- O QUE É: UM ÚNICO PACOTE QUE SABE E FAZ DE TUDO NO SISTEMA.
- PROBLEMA: PONTO ÚNICO DE FALHA; IMPOSSÍVEL DE MANTER.
- SOLUÇÃO: QUEBRAR O COMPONENTE EM PARTES MENORES, COESAS E INDEPENDENTES.

ANÁLISE ARQUITETURA

A métrica de **instabilidade** se relaciona com o **Unstable Dependency**



Um número alto de componentes nos extremos do histograma (muito estáveis e muito instáveis) aumenta a probabilidade de ocorrerem dependências incorretas entre eles, o que pode levar a um percentual significativo de smells de Unstable Dependency no relatório final

ANÁLISE DE TESTE

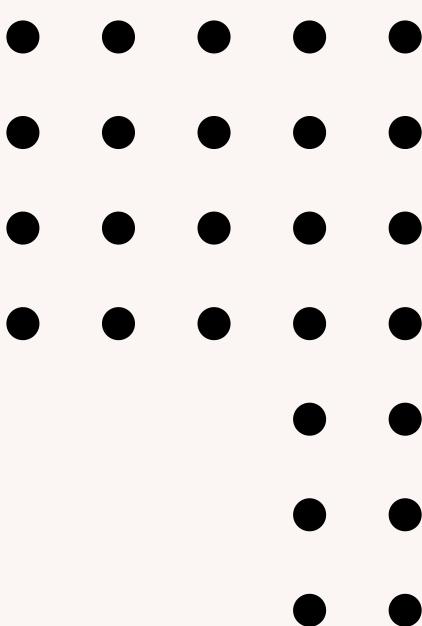


Quase 95% dos smells de testes são Assertion Roulette.
Dificuldade em identificar a causa do problema.

ANÁLISE TESTE

- **ASSERTION ROULETTE (ROLETA DE VERIFICAÇÃO)**
 - O QUE É: MÚLTIPLAS VERIFICAÇÕES (ASSERTS) EM UM ÚNICO MÉTODO DE TESTE.
 - PROBLEMA: QUANDO O TESTE FALHA, NÃO SE SABE QUAL VERIFICAÇÃO FOI A CULPADA.
 - SOLUÇÃO: UM TESTE DEVE FOCAR EM VERIFICAR UMA ÚNICA COISA. DIVIDIR EM TESTES MENORES.
- **CONSTRUCTOR INITIALIZATION (INICIALIZAÇÃO NO CONSTRUTOR)**
 - O QUE É: PREPARAR O AMBIENTE DO TESTE DENTRO DO CONSTRUTOR DA CLASSE.
 - PROBLEMA: DIFÍCULTA A LEITURA E O GERENCIAMENTO DO ESTADO INICIAL DOS TESTES.
 - SOLUÇÃO: USAR MÉTODOS DE "SETUP" DEDICADOS (EX: @BEFOREEACH).

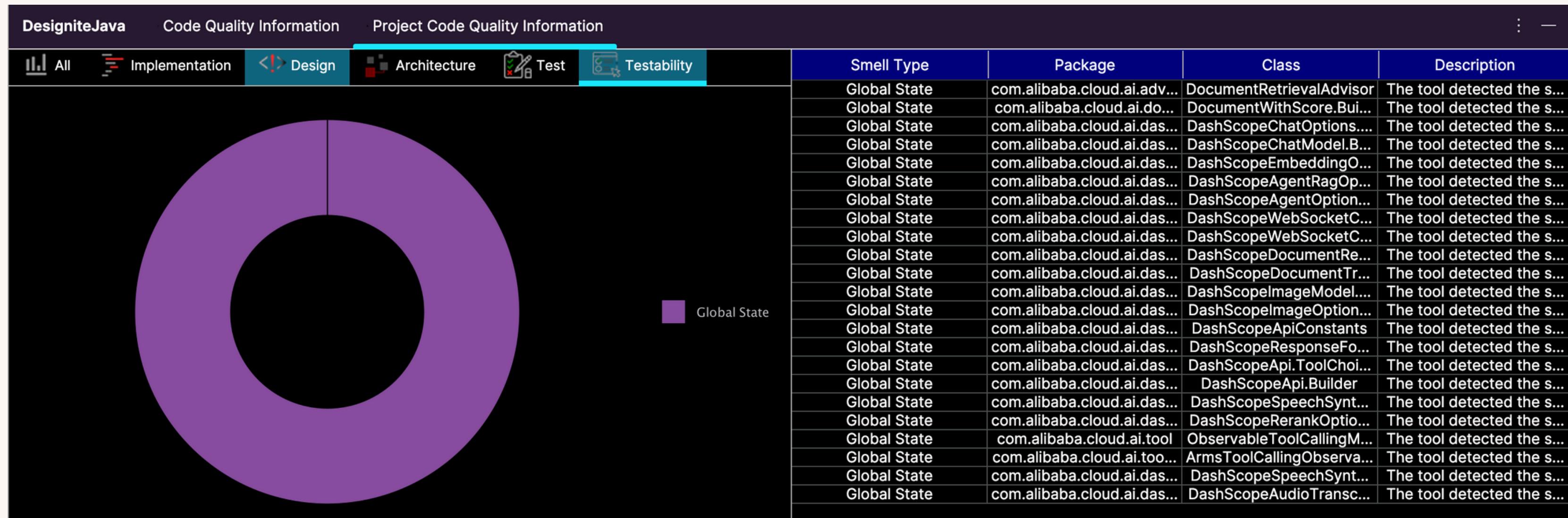
ANÁLISE TESTE



• MISSING ASSERTION (VERIFICAÇÃO FALTANTE)

- O QUE É: UM MÉTODO DE TESTE QUE NÃO VERIFICA NENHUM RESULTADO.
- PROBLEMA: TESTE INÚTIL QUE SEMPRE PASSA, GERANDO FALSA SEGURANÇA.
- SOLUÇÃO: ADICIONAR UMA VERIFICAÇÃO (ASSERT) OU REMOVER O TESTE.

ANÁLISE TESTABILIDADE



ANÁLISE TESTABILIDADE

- **GLOBAL STATE (ESTADO GLOBAL)**

- O QUE É: USO DE SINGLETONS OU VARIÁVEIS GLOBAIS/ESTÁTICAS QUE SÃO COMPARTILHADAS POR TODO O SISTEMA.
- PROBLEMA: IMPEDE O TESTE ISOLADO DAS CLASSES; UM TESTE PODE AFETAR O RESULTADO DE OUTRO, TORNANDO A SUÍTE NÃO CONFIÁVEL.
- SOLUÇÃO: ADOTAR INJEÇÃO DE DEPENDÊNCIA. EM VEZ DE A CLASSE BUSCAR A DEPENDÊNCIA, ELA DEVE RECEBÊ-LA DE FORA (EX: PELO CONSTRUTOR).

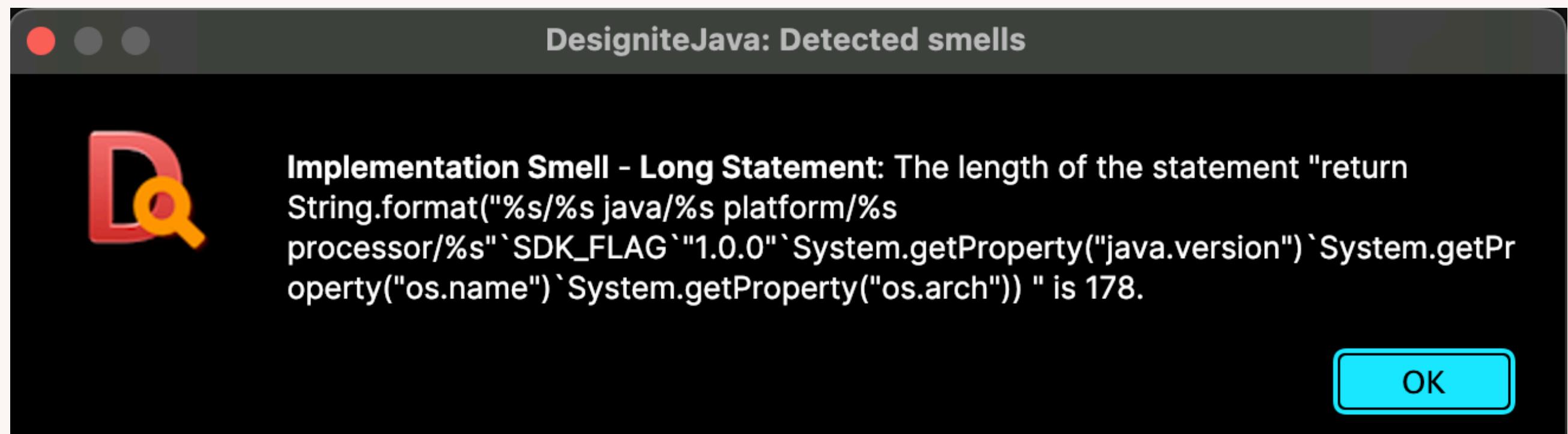
ANÁLISE

A métrica de **manutenibilidade** se relaciona com todos os smells, pois ela é afetada por eles. Code smells são os "sintomas" de uma doença chamada baixa manutenibilidade.

Element [1]	Maintainability Level
↳ SistemaAIAlibabaCore	71.16

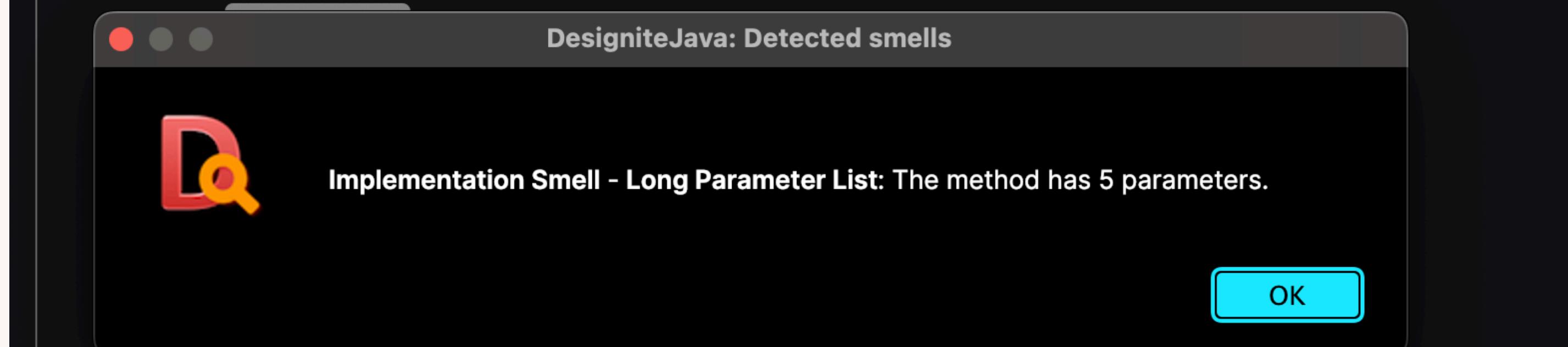
IMPLEMENTATION SMELL

```
1 usage  
① D  
private static String userAgent() {  
    return String.format("%s/%s; java/%s; platform/%s; processor/%s", SDK_FLAG, "1.0.0",  
        System.getProperty("java.version"), System.getProperty("os.name"), System.getProperty("os.arch"));  
}
```



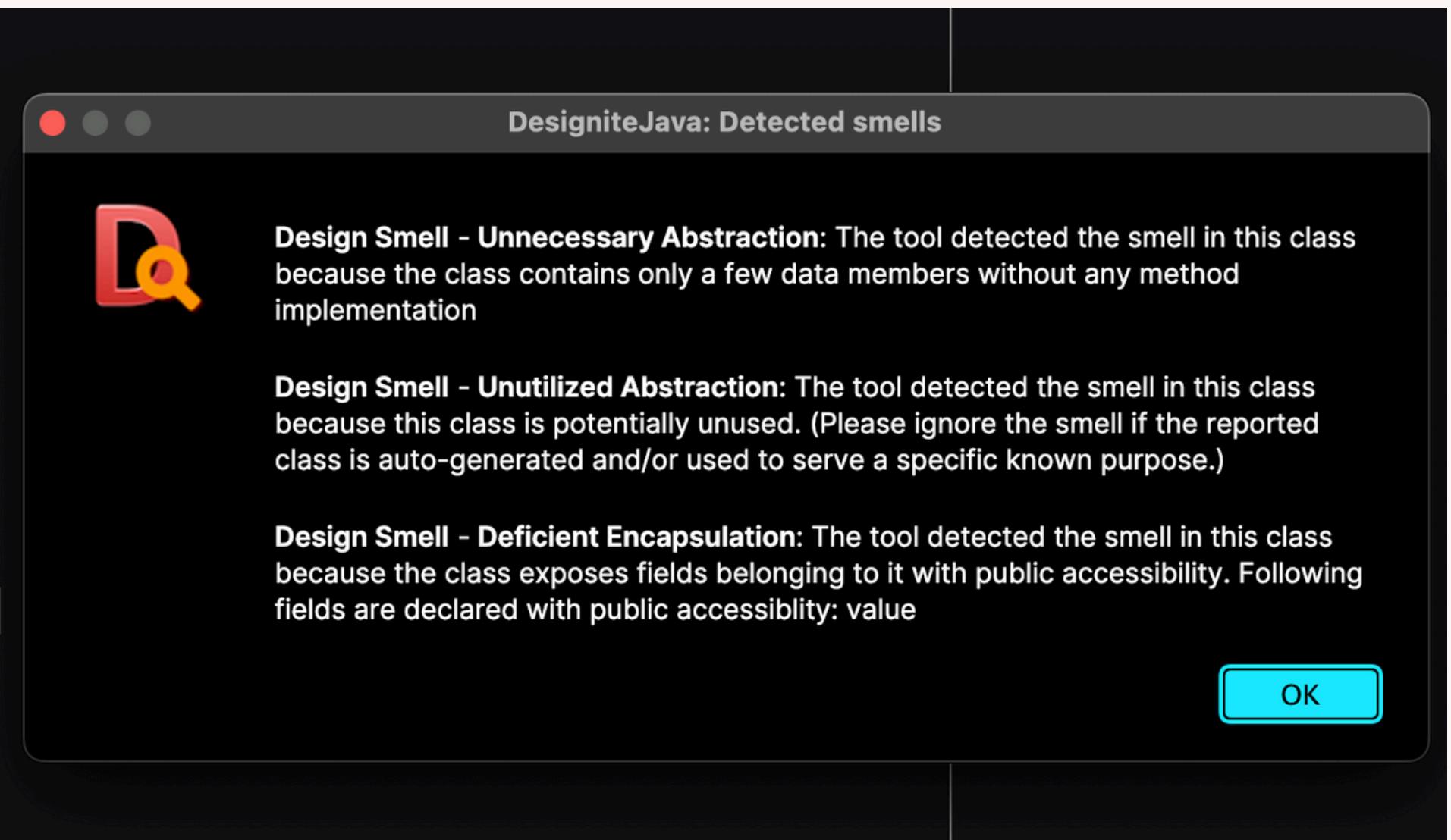
IMPLEMENTATION SMELL

```
3 usages
public static Consumer<HttpHeaders> getAudioTranscriptionHeaders(String apiKey, String workspace,
    Boolean isAsyncTask, Boolean isSecurityCheck, Boolean isSSE) {
```

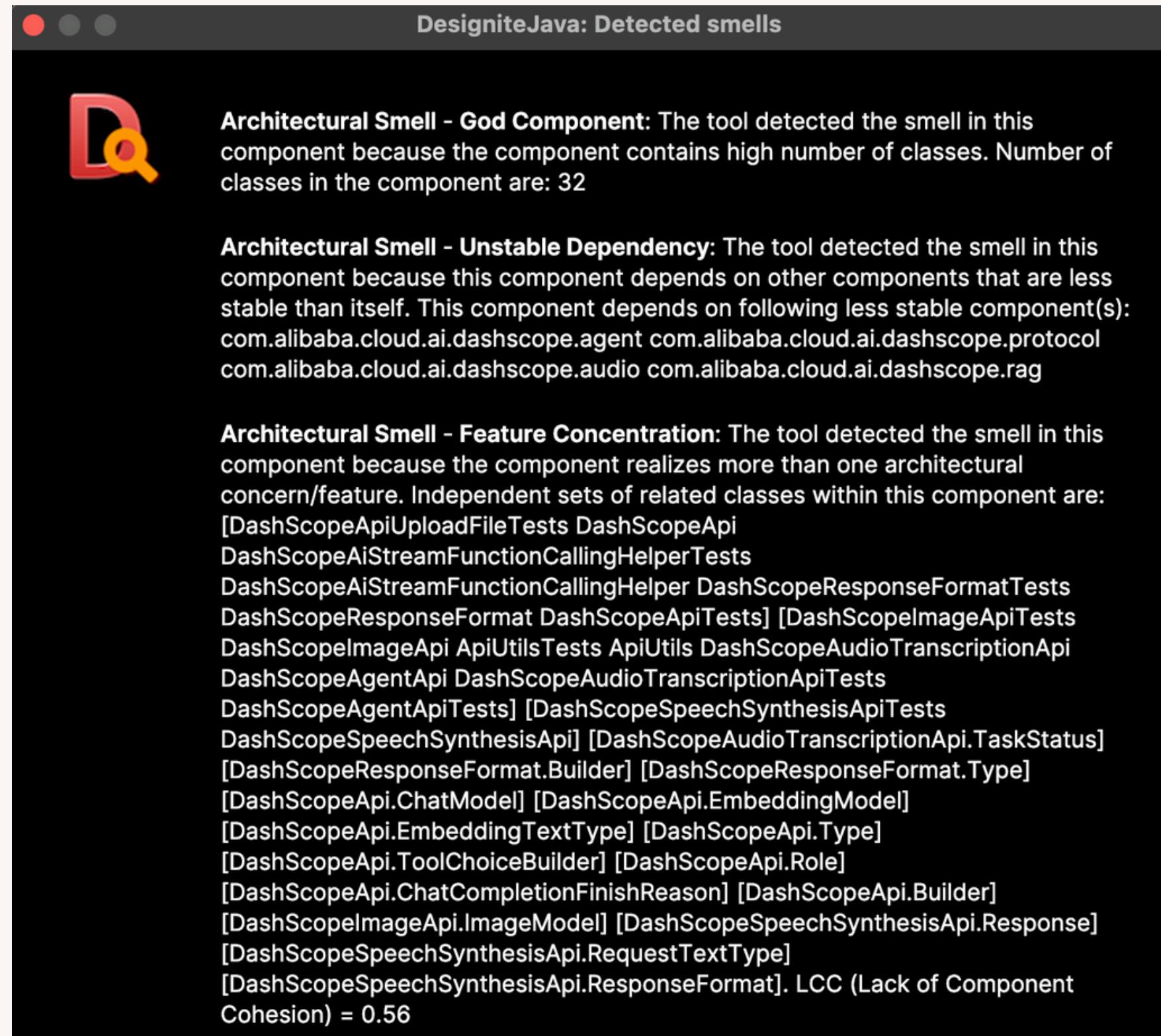


IMPLEMENTATION SMELL

```
4 usages
public enum EmbeddingTextType {
    1 usage
    QUERY(value: "query"),
    2 usages
    DOCUMENT(value: "document");
    public final String value;
    4 usages
    EmbeddingTextType(String value) {this.value = value;}
    public String getValue() {return value;}
}
```



ARCHITECTURAL SMELL



ARCHITECTURAL SMELL

Architectural Smell - Cyclic Dependency: The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.alibaba.cloud.ai.dashscope.api
com.alibaba.cloud.ai.dashscope.protocol com.alibaba.cloud.ai.dashscope.api

Architectural Smell - Cyclic Dependency: The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.alibaba.cloud.ai.dashscope.api
com.alibaba.cloud.ai.dashscope.audio com.alibaba.cloud.ai.dashscope.api

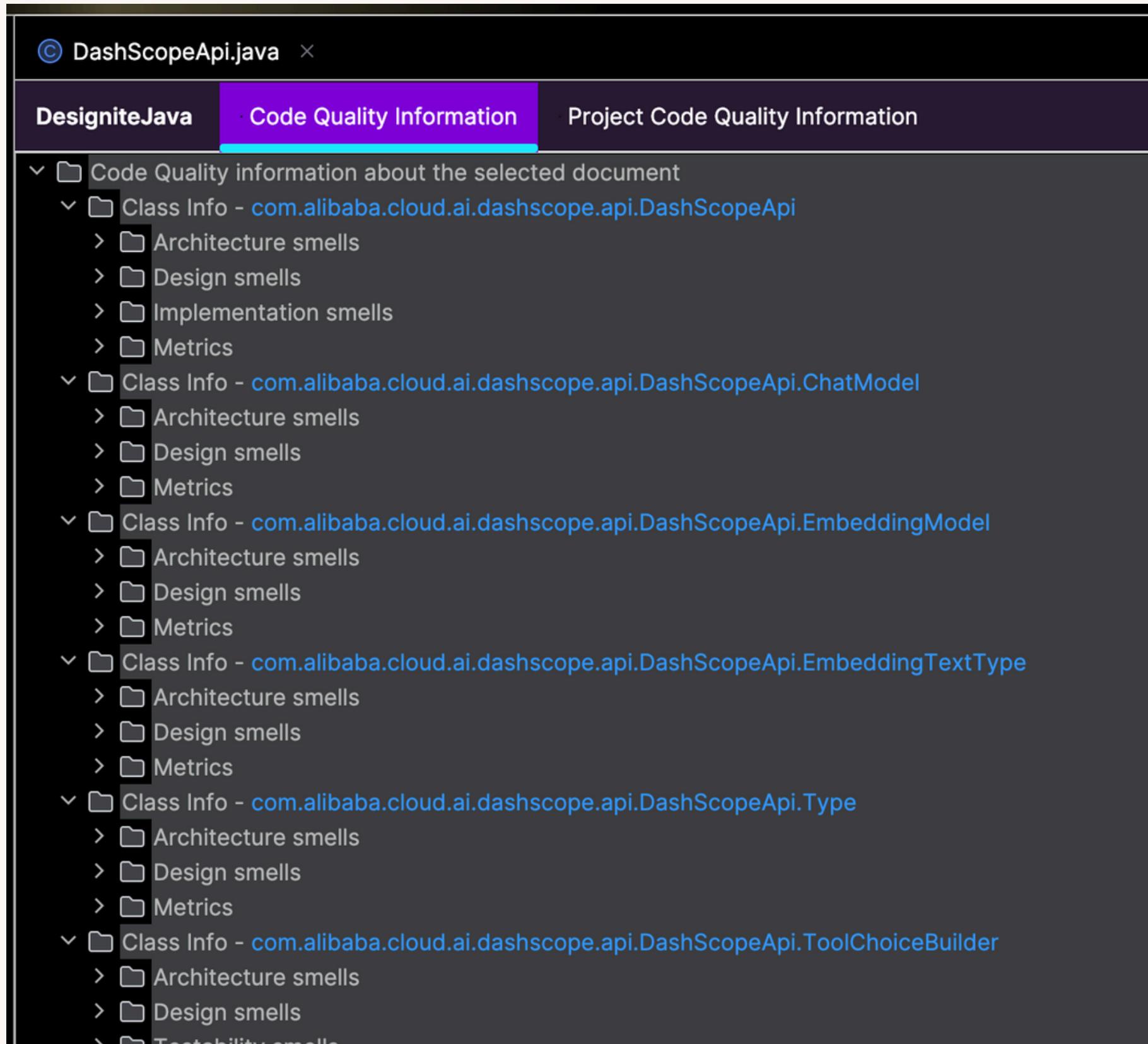
Architectural Smell - Cyclic Dependency: The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components in the cycle are: com.alibaba.cloud.ai.dashscope.api
com.alibaba.cloud.ai.dashscope.rag com.alibaba.cloud.ai.dashscope.api

CODE QUALITY INFORMATION

The screenshot shows a software interface for analyzing code quality. The title bar indicates the file is `DashScopeApi.java`. The top navigation bar has three tabs: `DesigniteJava`, `Code Quality Information` (which is highlighted with a purple background), and `Project Code Quality Information`. The main content area is a tree view under the heading "Code Quality information about the selected document". The tree structure is as follows:

- Class Info - `com.alibaba.cloud.ai.dashscope.api.DashScopeApi`
 - Architecture smells
 - Design smells
 - Implementation smells
 - Metrics
- Class Info - `com.alibaba.cloud.ai.dashscope.api.DashScopeApi.ChatModel`
 - Architecture smells
 - Design smells
 - Metrics
- Class Info - `com.alibaba.cloud.ai.dashscope.api.DashScopeApi.EmbeddingModel`
 - Architecture smells
 - Design smells
 - Metrics
- Class Info - `com.alibaba.cloud.ai.dashscope.api.DashScopeApi.EmbeddingTextType`
 - Architecture smells
 - Design smells
 - Metrics
- Class Info - `com.alibaba.cloud.ai.dashscope.api.DashScopeApi.Type`
 - Architecture smells
 - Design smells
 - Metrics
- Class Info - `com.alibaba.cloud.ai.dashscope.api.DashScopeApi.ToolChoiceBuilder`
 - Architecture smells
 - Design smells
 - Testability smells

CODE QUALITY INFORMATION



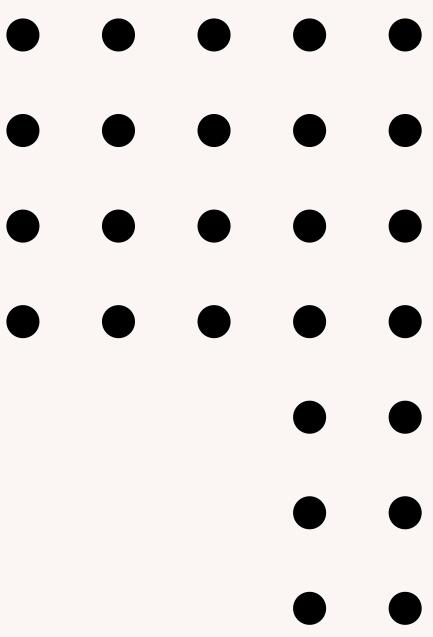
CODE QUALITY INFORMATION

Code Quality information about the selected document

- Class Info - [com.alibaba.cloud.ai.dashscope.api.DashScopeApi](#)
 - Architecture smells
 - God Component [in package: com.alibaba.cloud.ai.dashscope.api]: The tool detected the smell in this component because the component contains high number of classes. Number of classes in the component: 105
 - Unstable Dependency [in package: com.alibaba.cloud.ai.dashscope.api]: The tool detected the smell in this component because this component depends on other components that are less stable than itself.
 - Feature Concentration [in package: com.alibaba.cloud.ai.dashscope.api]: The tool detected the smell in this component because the component realizes more than one architectural concern/feature. Independence between concerns: 0.62
 - Cyclic Dependency [in package: com.alibaba.cloud.ai.dashscope.api]: The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components: 105
 - Cyclic Dependency [in package: com.alibaba.cloud.ai.dashscope.api]: The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components: 105
 - Cyclic Dependency [in package: com.alibaba.cloud.ai.dashscope.api]: The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components: 105
 - Cyclic Dependency [in package: com.alibaba.cloud.ai.dashscope.api]: The tool detected the smell in this component because this component participates in a cyclic dependency. The participating components: 105
 - Design smells
 - Deficient Encapsulation: The tool detected the smell in this class because the class exposes fields belonging to it with public accessibility. Following fields are declared with public accessibility: SSE_DONE_P
 - Insufficient Modularization: The tool detected the smell in this class because the class has bloated interface (large number of public methods). Total public methods in the class: 105 public methods
 - Implementation smells
 - Metrics
 - Lines of code: 888
 - Lack of cohesion of methods: 0.62
 - Weighted methods per class: 143
 - Number of fields: 26
 - Number of public fields: 8
 - Number of methods: 115
 - Number of public methods: 105
 - Fan-in: 24
 - Fan-out: 6
 - Depth of inheritance tree: 0
 - Number of children (subtypes): 0

CODE QUALITY INFORMATION

Implementation smells	
✓	Long Parameter List [in method: DashScopeApi and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The method has 7 parameters.
✓	Long Statement [in method: DashScopeApi and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The length of the statement "this.restClient=restClientBuilder.clone().baseUrl(baseUrl).defaultHea
✓	Long Parameter List [in method: EmbeddingList and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The method has 5 parameters.
✓	Long Statement [in method: EmbeddingRequest and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The length of the statement "this(DEFAULT_EMBEDDING_MODEL`new EmbeddingRequestInp
✓	Long Statement [in method: EmbeddingRequest and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The length of the statement "this(DEFAULT_EMBEDDING_MODEL`new EmbeddingRequestInp
✓	Long Statement [in method: embeddings and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The length of the statement "return this.restClient.post().uri("/api/v1/services/embeddings/text-emb
✓	Magic Number [in method: embeddings and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The method contains a magic number: 25
✓	Long Parameter List [in method: QueryFileResponseData and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The method has 7 parameters.
✓	Long Parameter List [in method: QueryFileParseResultData and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The method has 5 parameters.
✓	Long Parameter List [in method: DocumentSplitRequest and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The method has 6 parameters.
✓	Long Parameter List [in method: DocumentChunk and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The method has 6 parameters.
✓	Long Statement [in method: upload and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The length of the statement "throw new DashScopeException("ApplyLease Failed`code:%s`message:%s".
✓	Long Statement [in method: queryFileInfo and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The length of the statement "return this.restClient.post().uri("/api/v1/datacenter/category/{category
✓	Long Statement [in method: getFileParseResult and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The length of the statement "ResponseEntity> fileParseResponse=this.restClient.post().uri("/a
✓	Long Statement [in method: getFileParseResult and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The length of the statement "ResponseEntity response=restTemplate.exchange(new URI(com
✓	Long Statement [in method: addFile and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The length of the statement "ResponseEntity> response=this.restClient.post().uri("/api/v1/datacenter/cat
✓	Long Statement [in method: uploadFile and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The length of the statement "OkHttpClient client=new OkHttpClient.Builder().connectTimeout(60`Time
✓	Magic Number [in method: uploadFile and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The method contains a magic number: 60
✓	Magic Number [in method: uploadFile and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The method contains a magic number: 60
✓	Magic Number [in method: uploadFile and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The method contains a magic number: 60
✓	Long Statement [in method: uploadLease and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The length of the statement "return this.restClient.post().uri("/api/v1/datacenter/category/{categoryl
✓	Long Statement [in method: documentSplit and class: com.alibaba.cloud.ai.dashscope.api.DashScopeApi]: The length of the statement "DocumentSplitRequest request=new DocumentSplitRequest(docume



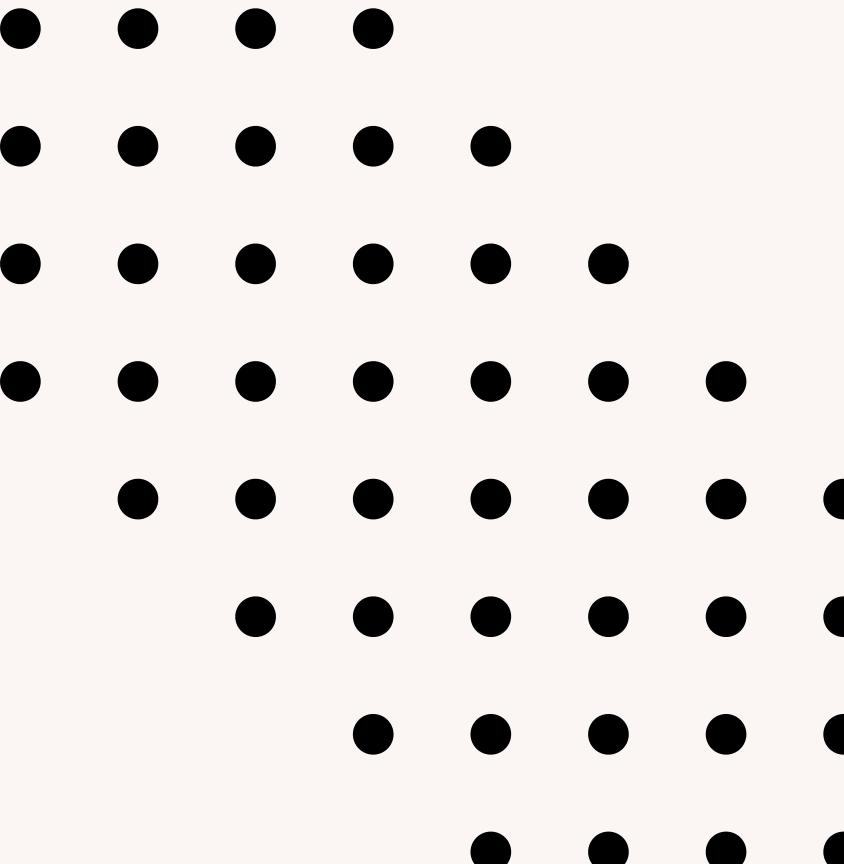
VANTAGENS X DESVANTAGENS

Pontos positivos:

- Caso a classe ou método tenha smells, mostra direto do lado do código
- Análise direta na IDE, não precisa de programa externo, então mais facilidade de rapidez para analisar
- Gerar relatórios como CSV ajuda em análises e documentações, além de ser possível desses dados serem usados por outras ferramentas

Pontos negativos:

- Difícil de configurar, não fica claro quais erros ocorrem ao tentar rodar (é necessário entrar nos logs da IDE para entender)
- UI desatualizada e nada amigável, alguns dados ficam cortados e não é possível visualizar tudo sem entrar nos logs output na pasta raiz
- A ferramenta pode, às vezes, detectar comentários como smells. Por exemplo, comentários longos são detectados como código duplicado ou Long Statement
- Pouco utilizado, sem comunidade consolidada ou suporte, difícil achar tutoriais ou muitas informações online



Agradecemos a atenção!

