

目录

1. 启动应用.....	- 3 -
2. 新建项目，选择源码及文档上传.....	- 3 -
3. 点击“提交”	- 6 -
4. 删除项目.....	- 6 -
5. 点击项目，进入项目页	- 6 -
6. “信息提取及可视化”栏	- 7 -
6.1 项目信息提取	- 8 -
6.2 代码解析及信息提取	- 8 -
6.3 设计恢复	- 10 -
6.3.1 控制流图	- 10 -
6.3.2 过程依赖图.....	- 10 -
6.3.3 调用图	- 11 -
6.3.4 系统依赖图.....	- 12 -
6.3.5 架构依赖图.....	- 13 -
6.3.6 管道过滤器风格架构图.....	- 13 -
6.3.7 调用返回风格架构图.....	- 14 -
6.3.8 分层风格架构图.....	- 14 -
6.3.9 类图.....	- 15 -
7. “设计度量”栏.....	- 15 -
7.1 缺陷率计算.....	- 16 -
7.2 坏味率计算.....	- 17 -
7.3 软件质量评估	- 20 -
7.3.1 指标选择	- 20 -
7.3.2 权重设置	- 21 -
7.3.3 结果展示	- 22 -
7.4 设计质量评估	- 25 -
7.4.1 指标选择	- 25 -
7.4.2 结果展示	- 26 -
8. “变更信息检测”栏.....	- 26 -
9. “设计重构建议”栏.....	- 29 -

1. 启动应用

进入工具首页。首页项目中，项目右上角显示项目类型，分别有“C 项目”，“C&C++混合项目”以及“C++项目”三种类型。右上角为用户手册按钮，点击该按钮即可查看用户手册。

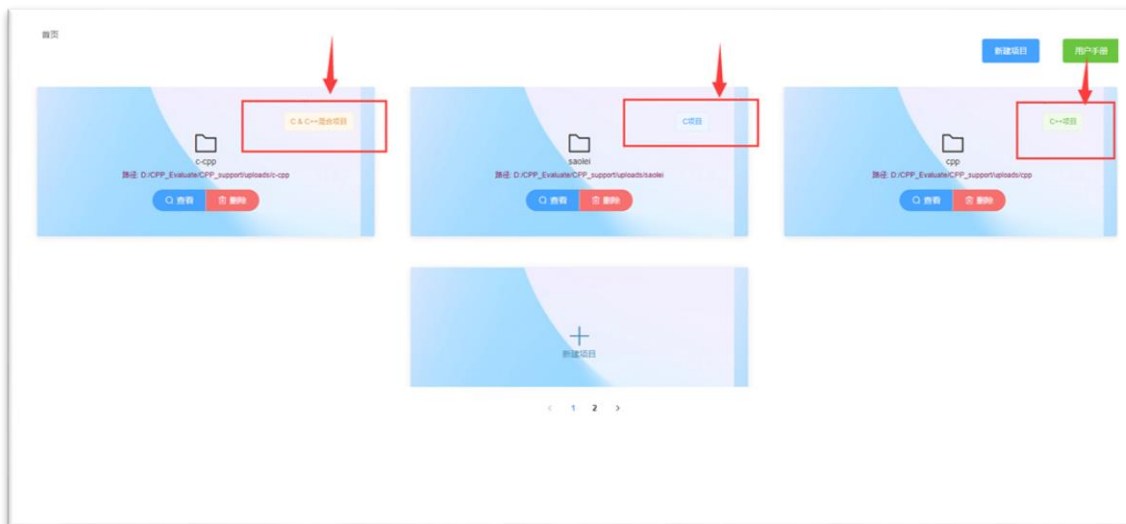


图 1 工具首页



图 2 用户手册页面

2. 新建项目，选择源码及文档上传

点击首页“新建项目”，弹出创建操作提示窗口。关闭窗口后弹出新建项目名称窗口，填写项目名称后点击“创建”按钮，进入选择源码和上传文档部分。

源码上传后，点击“上传”按钮，进入下一步选择需要解析的文件夹；选择完毕后点击“确定”进入下一步选择需要上传的文件；上传需要的文件，此处支持“Word 文档”、“禅道文档”、“TMMI 文档”以及“PingCode 文档”以上 4 种模板。



图 3 创建操作提示窗口

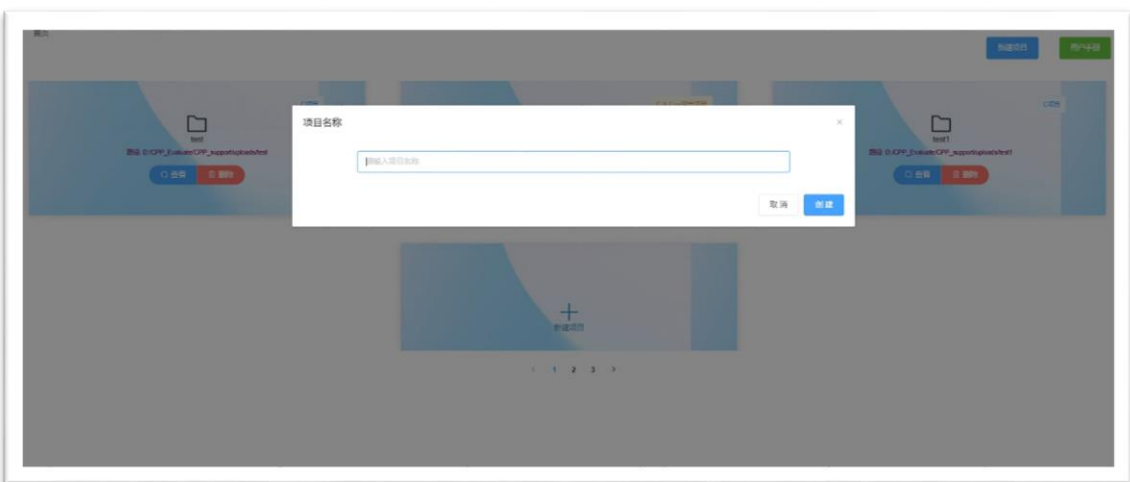


图 4 新建项目名称窗口

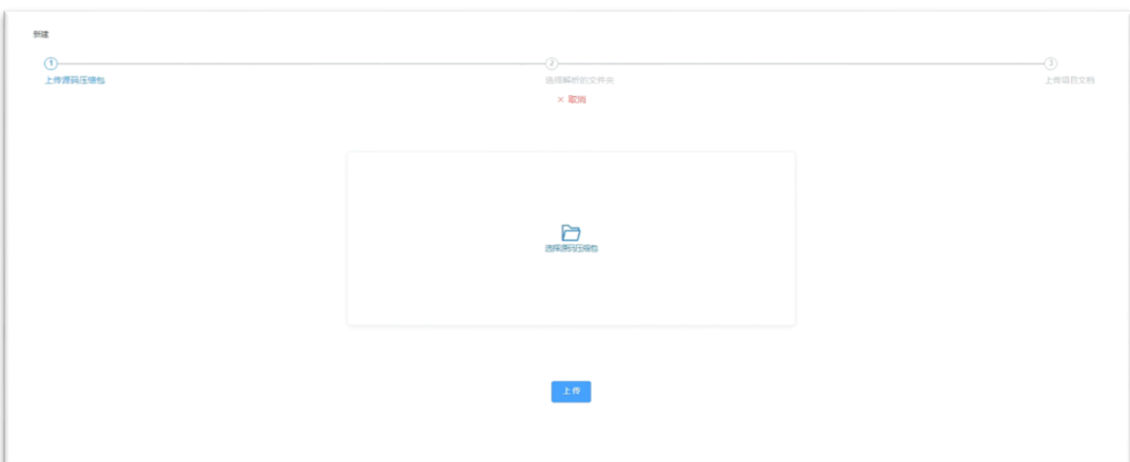


图 5 上传源码

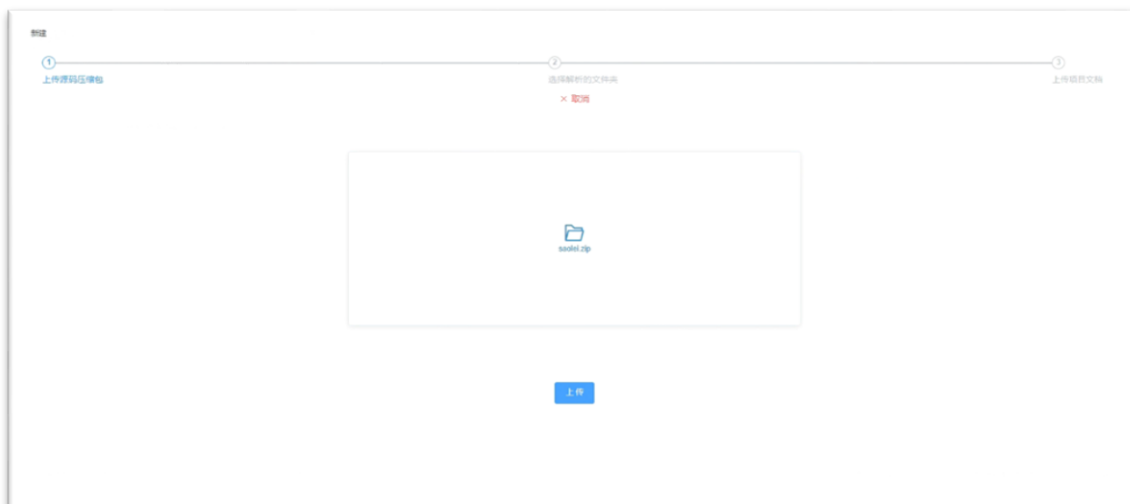


图 6 源码上传后

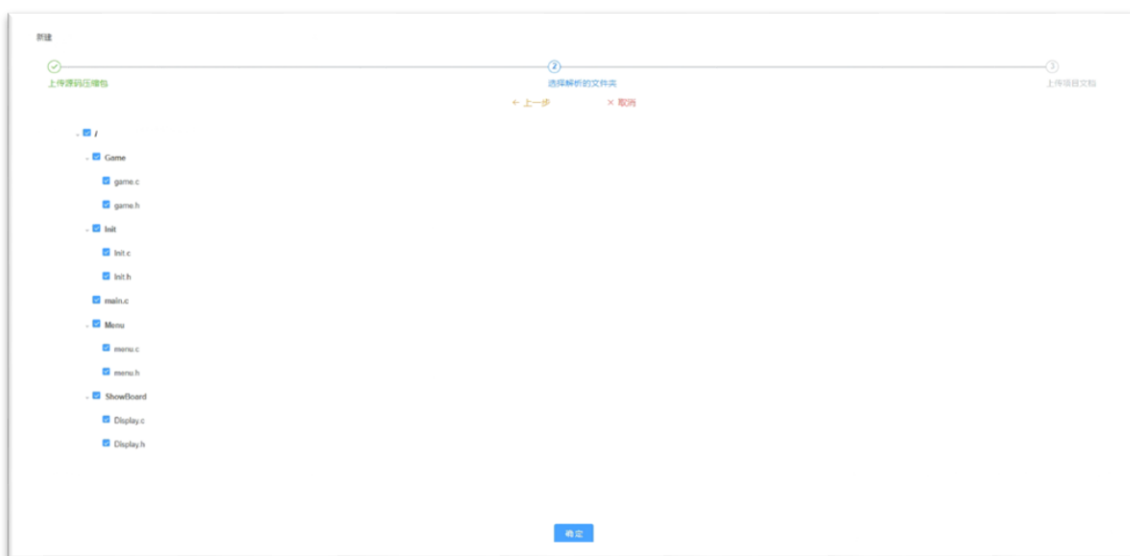


图 7 选择解析的文件夹

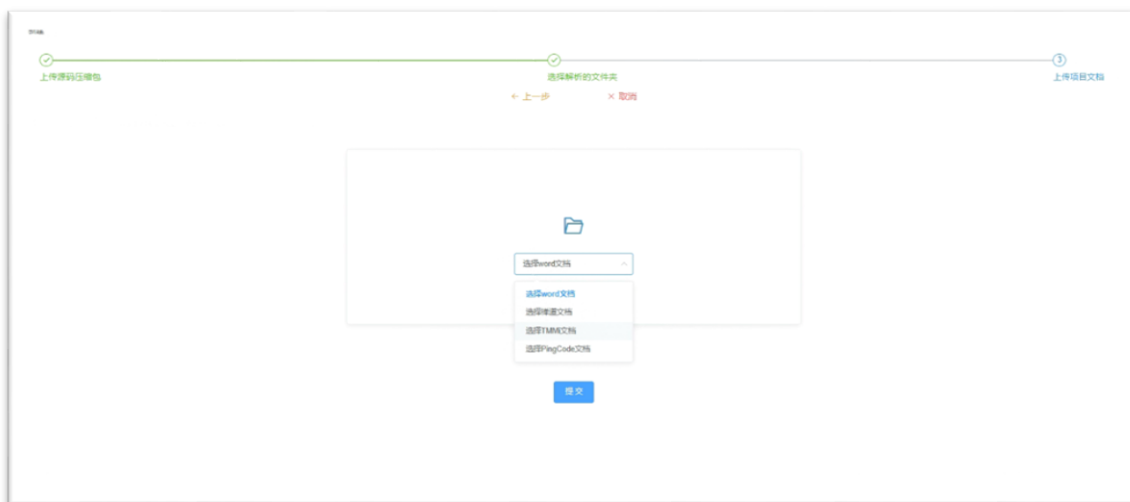


图 8 上传项目文档

3. 点击“提交”

如果只上传了源码，点击“提交”按钮后弹出“是否仅源码分析”提示框，点击“确定”后仍可创建。创建成功后，回到首页。首页中，项目右上角显示项目语言类型。

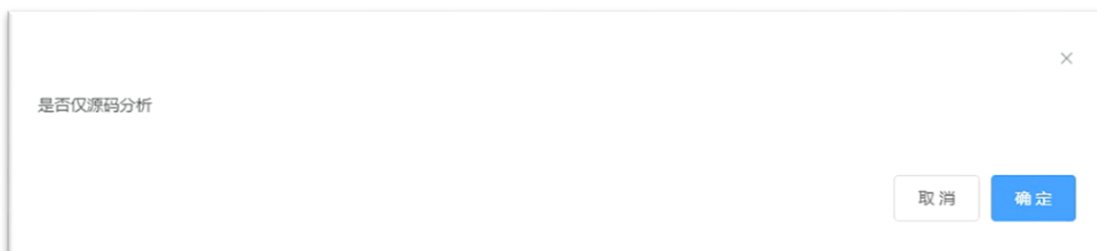


图 9 “是否仅源码分析”提示框

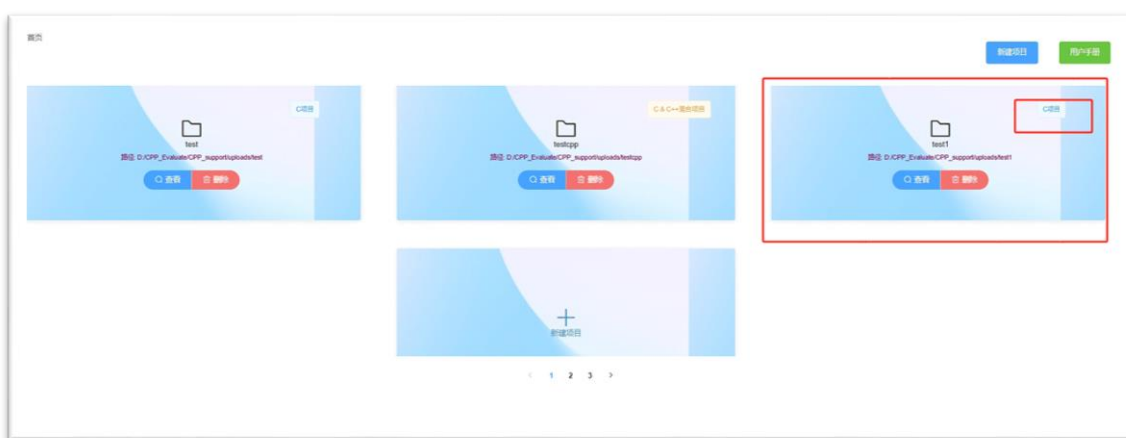


图 10 回到首页，首页包含新创建的项目

4. 删除项目

点击首页上的删除按钮，弹出提示框提示是否删除。

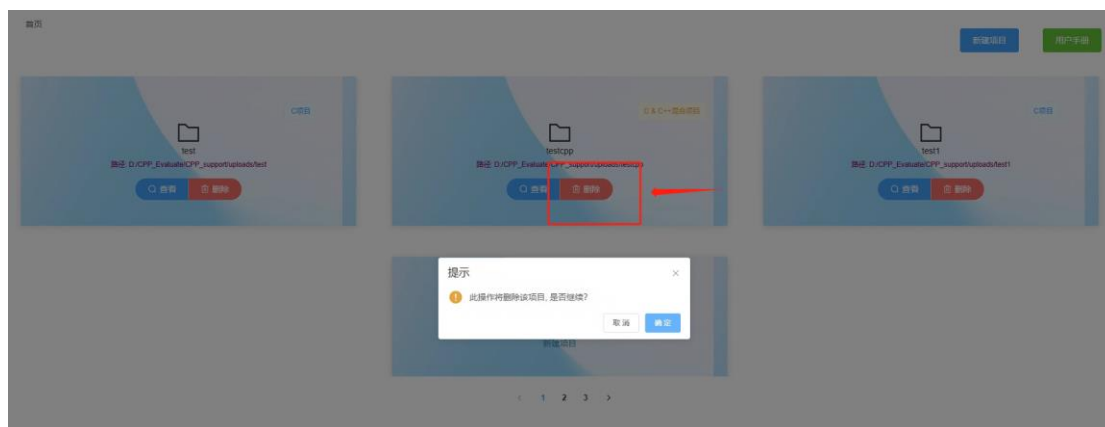


图 11 删除项目

5. 点击项目，进入项目页

点击项目，如果项目解析未完成，点击项目后弹出提示框“项目尚未加载完成，请稍后”；如果项目解析失败，点击项目后弹出提示框“项目解析失败”

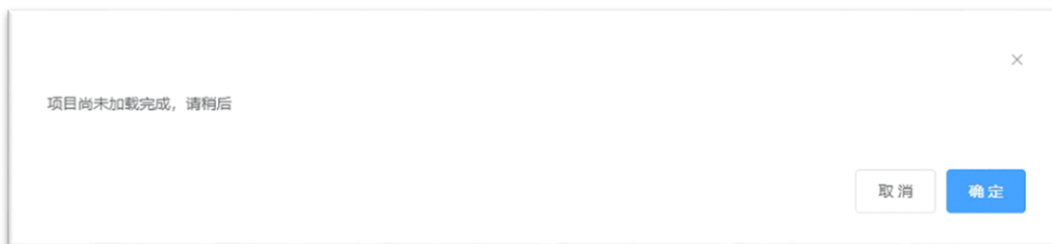


图 12 “项目尚未加载完成，请稍后”提示框

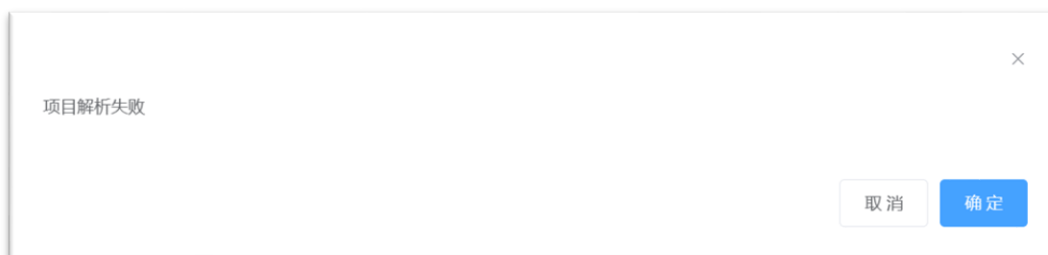


图 13 “项目解析失败”提示框

项目加载完成后点击项目，进入项目页。进入项目后默认首先加载项目级源码信息。项目页的导航栏包括“信息提取及可视化”、“设计度量”、“变更信息检测”和“设计重构建议”。

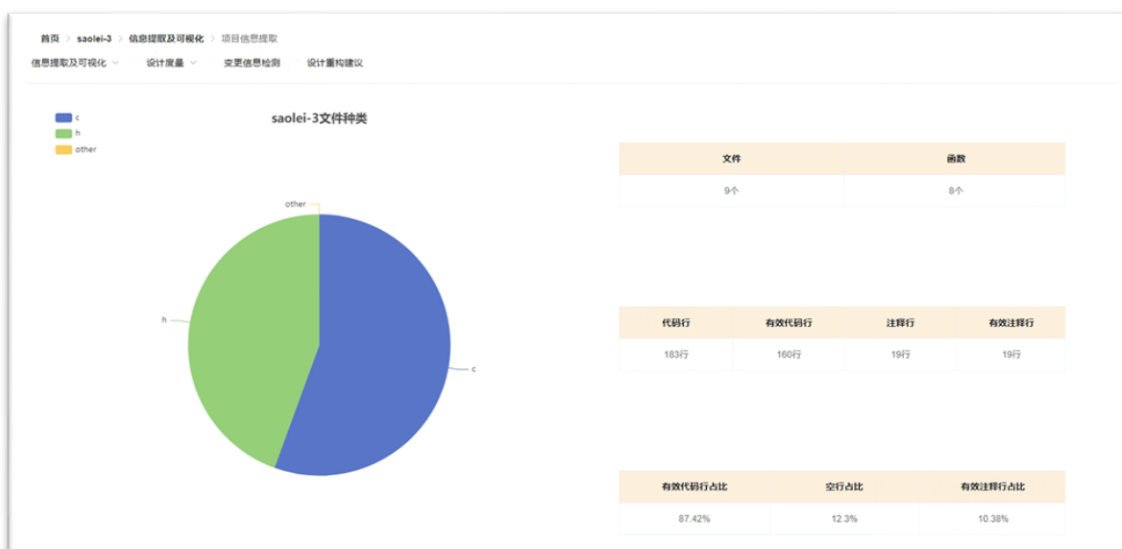


图 14 进入项目解析结果页

6. “信息提取及可视化”栏

“信息提取及可视化”栏包括项目信息提取、代码解析及信息提取、设计恢复。

C 项目设计恢复包括控制流程图、过程依赖图、调用图、系统依赖图、架构依赖图、管道过滤器风格架构图、调用返回风格架构图、分层风格架构图。C&C++ 混源项目和 C++项目的设计恢复只包括控制流程图、过程依赖图、调用图和类图。



图 15 C 项目-“信息提取及可视化”栏



图 16 C&C++混源项目和 C++项目-“信息提取及可视化栏”

6.1 项目信息提取

在“项目信息提取”页面，左侧饼状图显示项目的文件种类及其占比，右侧显示文件、函数以及代码的具体信息。

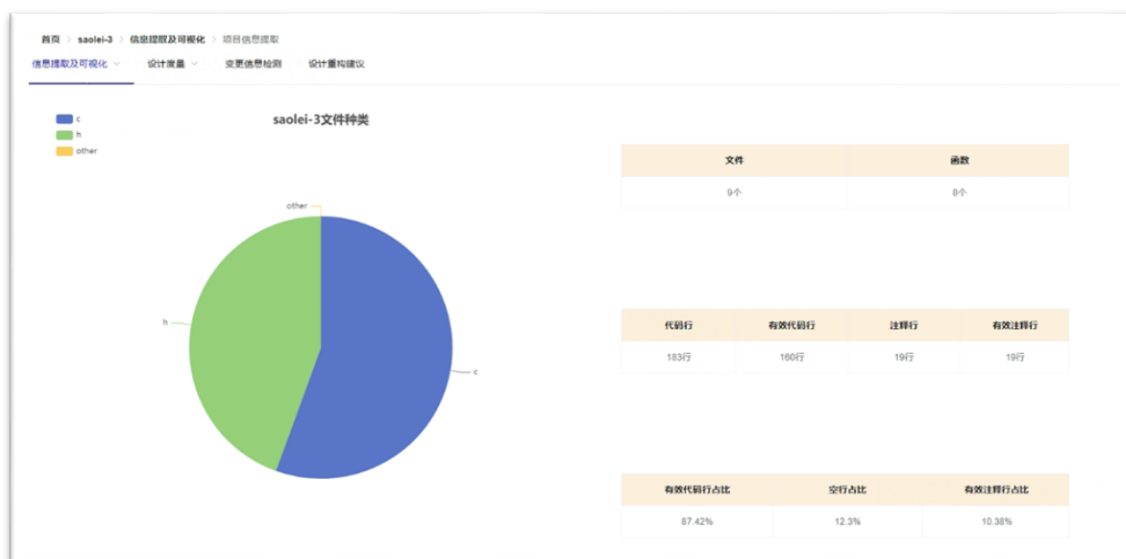


图 17 “项目信息提取”页面

6.2 代码解析及信息提取

在“代码解析及信息提取”界面，C 项目中，页面上方显示 c 语言文件占比、最大行数函数信息、最大行数文件信息、包含函数最多的文件信息，页面下方通过下拉框选择文件。选择文件后，展示该文件的函数列表以及函数的相关信息，包括返回值、参数、出度、入度、文件位置。

C 项目为 C 信息页面，C++项目为 C++信息页面，C&C++混源项目为 C 信息和 C++信息页面。

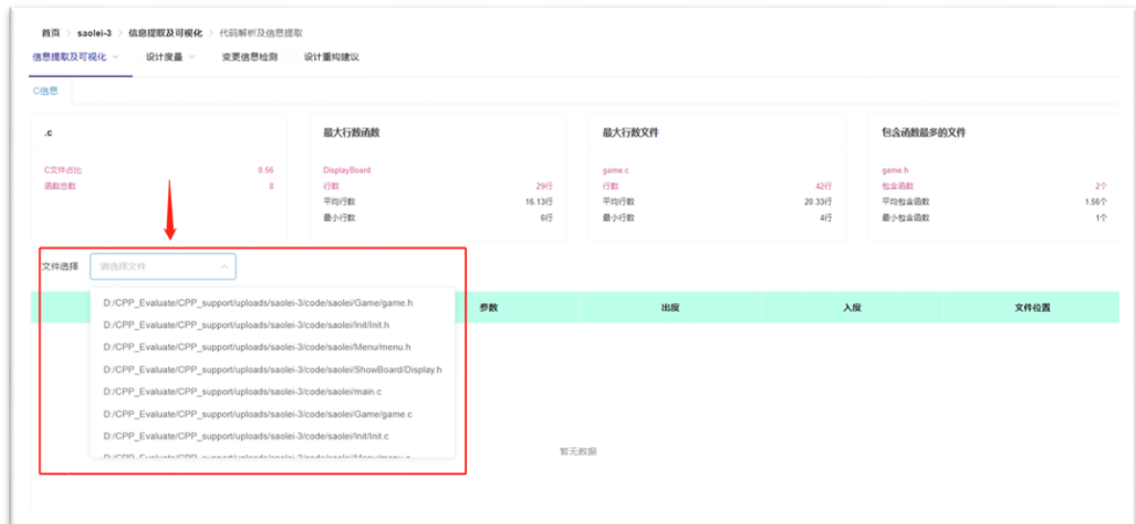


图 18 C 项目-“代码解析及可视化”页面-选择文件



图 19 C&C++混源项目-“代码解析及可视化”页面-文件信息详情



图 20 “代码解析及可视化”页面-文件信息详情

6.3 设计恢复

6.3.1 控制流图

控制流图(Control Flow Graph, CFG)是一个有向图 $G = (V, E)$ ，其中 V 是顶点的集合， E 是有向边的集合。在 CFG 中，每个顶点代表一个基本块，该基本块是具有一个入口点(执行的第一条指令)和一个出口点(执行最后一条指令)的程序指令的线性序列，有向边显示控制流路径。CFG 用图的形式表示一个过程内所有基本块执行的可能流向，能反映一个过程的实时执行过程。从 CFG 中可以提取过程中顺序结构、选择结构和循环结构，观察过程内各个基本块之间的关系，有助于帮助分析程序代码执行的流程和可能的控制流路径，可以作为静态分析工具的基础。

页面左侧显示文件层级，中间显示选中的文件中函数的控制流图，右侧显示图信息以及点击的当前的节点信息。

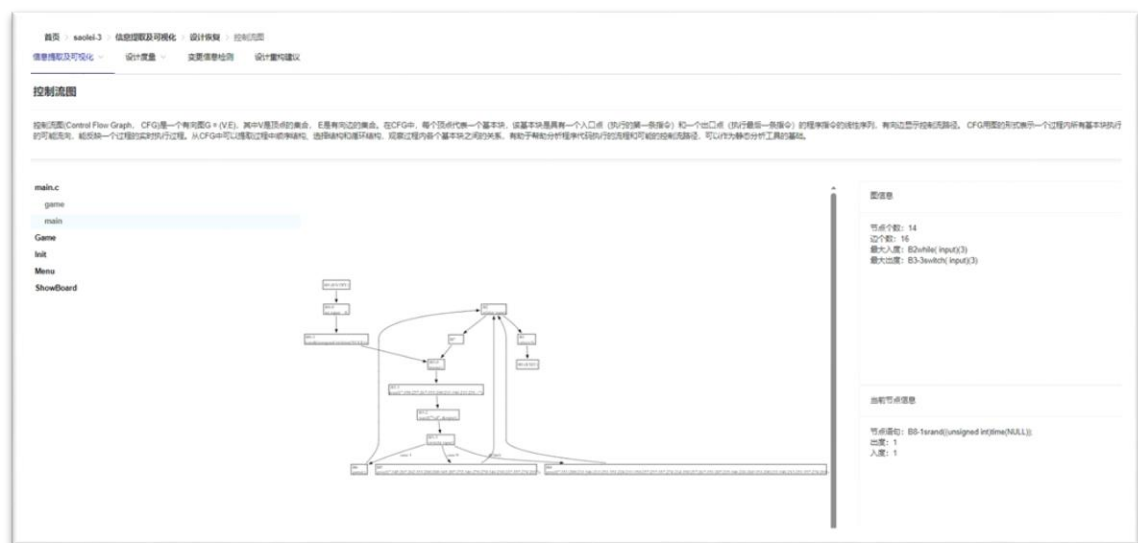


图 21 “设计恢复”-“控制流图”界面

6.3.2 过程依赖图

过程依赖图(Procedure Dependence Graph, PDG)是一个有向图 $G = (V, E)$ ， V 为节点集合，是语句和谓词表达式(或运算符和操作数)， E 为边集合表示依赖关系，依赖关系包括数据依赖和控制依赖。PDG 明确了程序中每个操作的数据和控制依赖关系，可以帮助分析程序中语句之间的依赖关系、可达性分析和数据流分析等。通过分析 PDG，可以帮助开发人员更好地理解代码的结构，提高代码的可读性和可维护性。

页面左侧显示文件层级，中间显示所选文件中函数的过程依赖图，右侧显示图信息以及点击的当前的节点信息。



图 24 “设计恢复” - “调用图” - “函数调用图” 界面，文件级

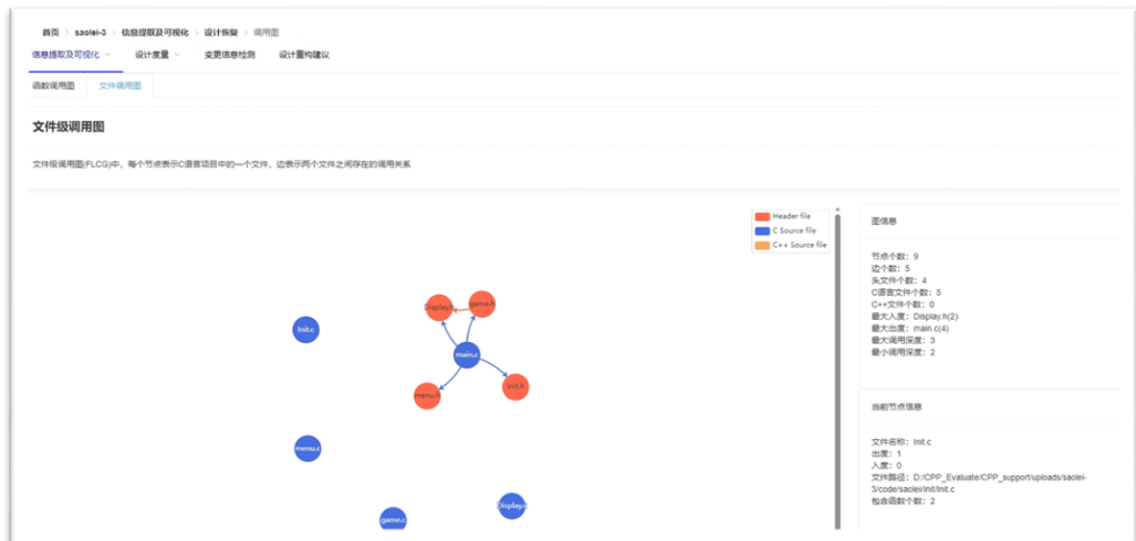


图 25 “设计恢复” - “调用图” - “文件调用图” 界面

6.3.4 系统依赖图

系统依赖图（System Dependence Graph, SDG）是对过程依赖图（PDG）的扩展，SDG 是一个有向图 $G=(N,E)$ ，其中，N 中的每个节点表示程序代码中的语句，而 E 中的每条边表示的式节点中的依赖关系。在 SDG 中，边分为两种类型：控制依赖边和数据依赖边。SDG 可以用于表示和分析系统中各组件之间依赖关系，包括数据依赖、控制依赖等，可以帮助开发人员更好地理解系统结构和优化系统性能，提高系统的可靠性、可维护性和可扩展性。

页面左侧显示系统依赖图，右侧显示图信息以及点击的当前的节点信息。

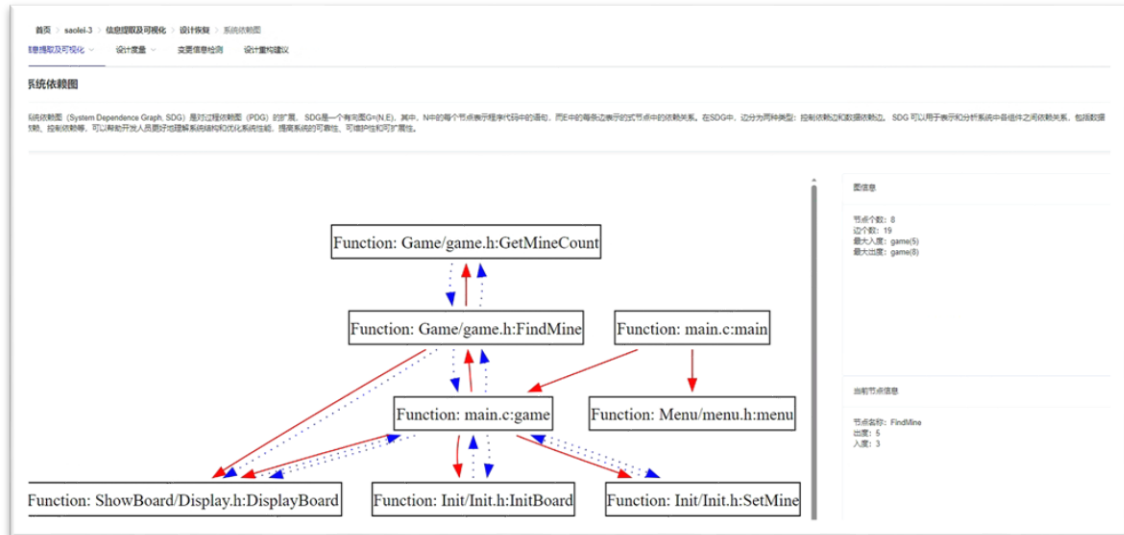


图 26 “设计恢复” - “系统依赖图”界面

6.3.5 架构依赖图

系统架构图是在系统依赖图的基础上恢复的，图中的节点是系统中的组件，组件是由一组紧密关联的函数组成，边表示组件之间的依赖关系。它提供了系统组件架构的简明描述，以协助组件与组件之间的关系和系统的运作。系统架构图是软件架构的一个可视化表示，可以清晰地表示系统中不同组件之间的关系，以便系统设计者和软件工程师更好地了解系统的架构。

页面左侧显示架构依赖图，右侧显示图信息以及点击的当前的节点信息。



图 27 “设计恢复” - “架构依赖图”界面

6.3.6 管道过滤器风格架构图

过滤器和管道体系结构风格为处理数据流的系统提供了一种结构。每个处理步骤封装在一个过滤器组件中。数据通过相邻过滤器之间的管道传输。每个过滤器有一组输入端和输出端。一个过滤器从输入端读取数据流，通过本地转换和渐增计算，向输出端输出数据流。管道充当数据流的通道，将一个过滤器的输出端连接到另一个过滤器的输入端。

首页 · [scool-3](#) · [信息提取及可视化](#) · [设计案例](#) · [普通过滤器风格架构图](#)

[信息提取及可视化](#) · [设计案例](#) · [普通过滤器风格架构图](#)

普通过滤器风格架构图

过滤器和管道系统为程序处理数据流提供了一种结构。每个处理步骤都在一个过滤器组件中，数据通过管道过滤器之间的管道传输。每个过滤器有一组输入和输出流。一个过滤器从输入流读取数据流，通过本地处理和计算，生成新的输出数据流。管道流由数据流的通道将一个过滤器的输出流连接到另一个过滤器的输入流。

```

graph LR
    Pump1((Pump)) --> F1[Filter: game.h.startMain]
    F1 --> F2[Filter: game.h.getMainCount]
    F2 --> F3[Filter: main.c.game]
    F3 --> F4[Filter: last.h.setMain]
    F3 --> F5[Filter: last.h.lastBoard]
    F4 --> F6[Filter: ShowBoard Display.h.Display Board]
    F5 --> F6
    F6 --> F7[Filter: info.h.print]
    F7 --> Sink1((Sink))
    F3 --> F7
    Pump2((Pump)) --> F8[Filter: info.h.scan]
    F8 --> F9[Filter: main.c.main]
    F9 --> F10[Filter: time.h.time]
    F9 --> F11[Filter: Menu menu.h.menu]
    F10 --> F8
    F11 --> F7
    F8 --> F12[Filter: info.h.setMain]
    F12 --> F1
  
```

图例

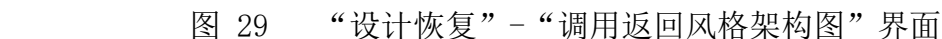
过滤器个数: 13
 管道个数: 23
 数据流入口个数: 2
 数据流出口个数: 2
 最大入度: 4
 最大出度: 7

节点节点信息

节点名称: game
 节点类型: 过滤器
 出度: 4
 入度: 1
 节点所在文件: main.c

6.3.7 调用返回风格架构图

页面左侧显示调用返回风格架构图,右侧显示图信息以及点击的当前的节点信息。



分层架构模式是一种分解系统结构的技术,其将整个系统分解为一组子任务,每组子任务基于特定的抽象层次,利用分层模式可将高等级的职责分解为较细粒度的职责。换句话说,分层模式设计的系统架构是一个有组织的分层结构,其中任一层次内软件实体负责向上层软件实体提供服务并使用下层实体的服务实现

自身功能。

页面左侧显示分层风格架构图，右侧显示图信息以及点击的当前的节点信息。

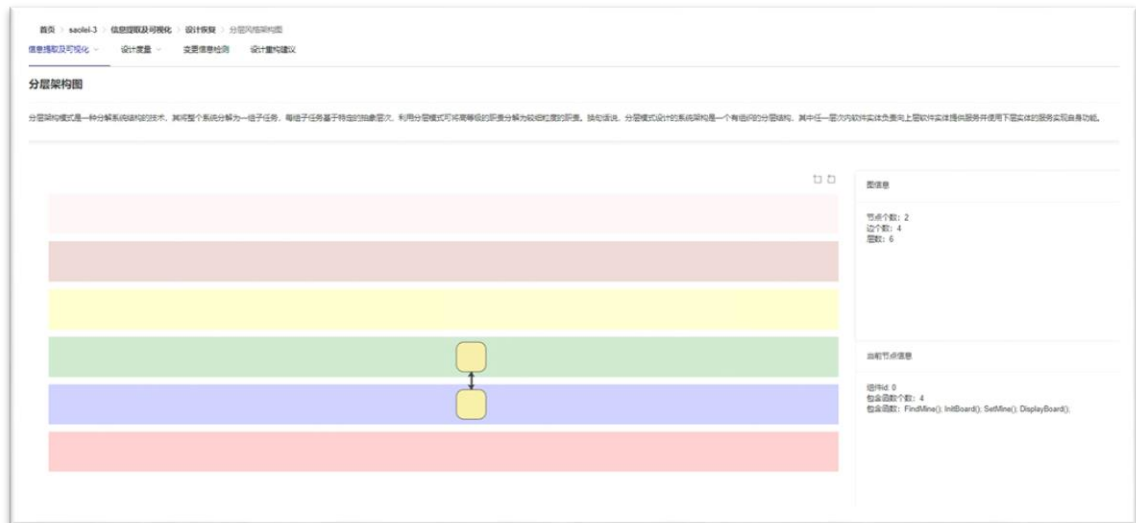


图 30 “设计恢复” - “分层风格架构图”界面

6.3.9 类图

类图(Class diagram)是显示了模型的静态结构，特别是模型中存在的类、类的内部结构以及它们与其他类的关系等。类图不显示暂时性的信息。类图是面向对象建模的主要组成部分。它既用于应用程序的系统分类的一般概念建模，也用于详细建模，将模型转换成编程代码。类图也可用于数据建模。

页面左侧显示类图，右侧显示图信息以及点击的当前的节点信息。

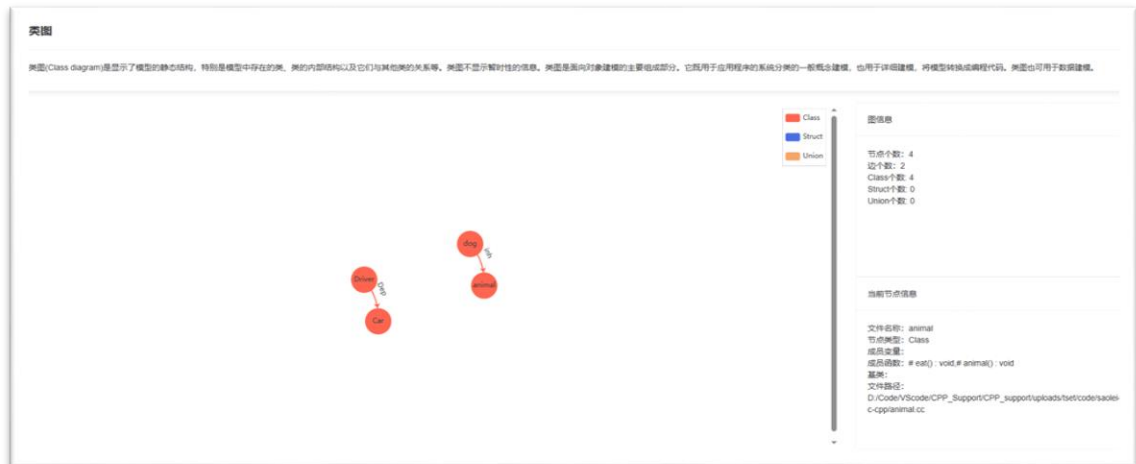


图 31 “设计恢复” - “类图”界面

7. “设计度量”栏

“设计度量”栏包括缺陷率计算、坏味率计算、软件质量评估（其中包括指标选择、圈中设置、结果展示）、设计质量评估（指标选择、结果展示）。



图 32 “设计度量”-“软件质量评估”栏



图 33 “设计度量”-“设计质量评估”栏

7.1 缺陷率计算

点击导航栏“缺陷率计算”进入页面。

在“总览信息”子页面中，通过选择文件路径（需是html文件），可以查看文件的千行缺陷数、缺陷密度等信息；在下方输入不同的权重可以更新饼图，查看建议类缺陷、强制类缺陷以及可选类缺陷的占比。

在“详细信息”子页面中，通过选择文件路径，可以查看强制类缺陷、可选类缺陷以及建议类缺陷的详细信息。

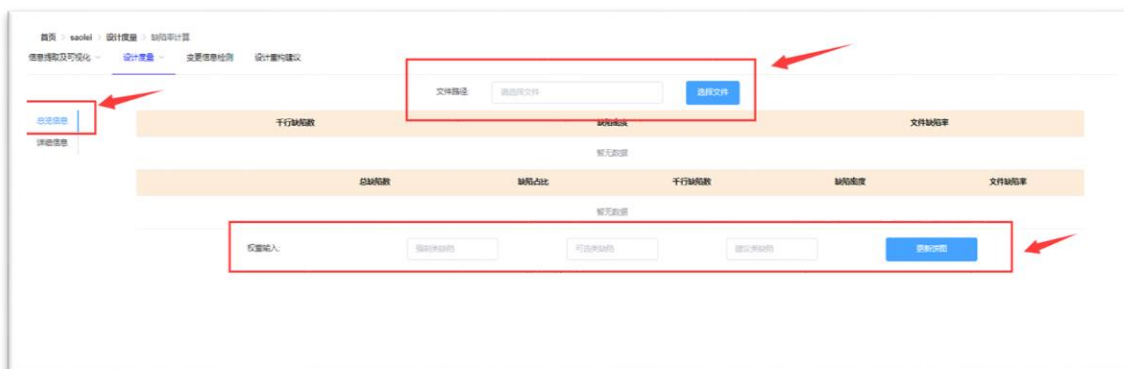


图 34 “缺陷率计算”-“总览信息”页面

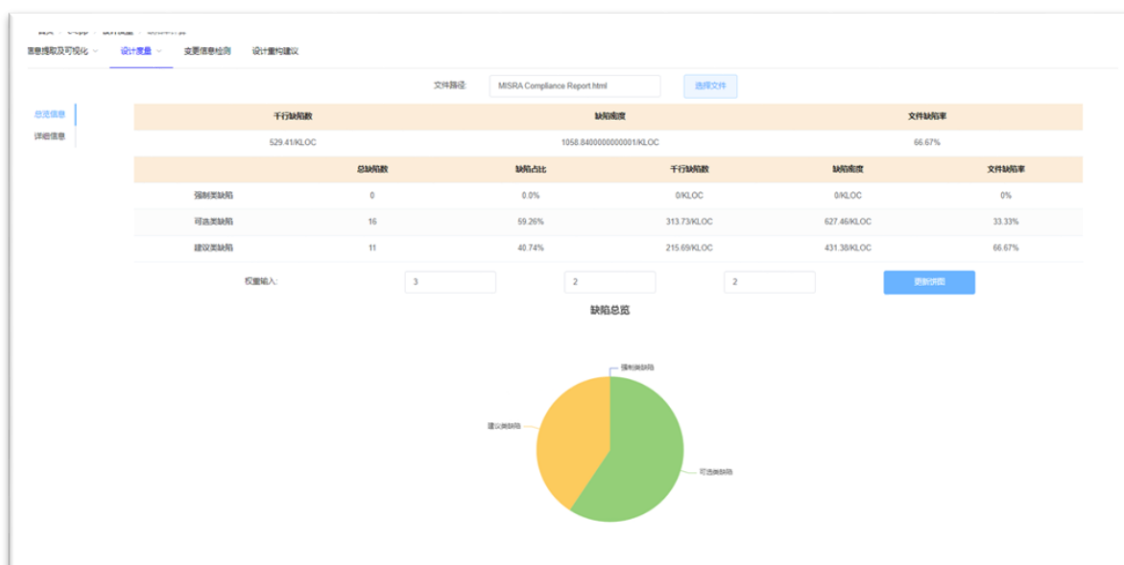


图 35 “缺陷率计算” - “总览信息” 页面

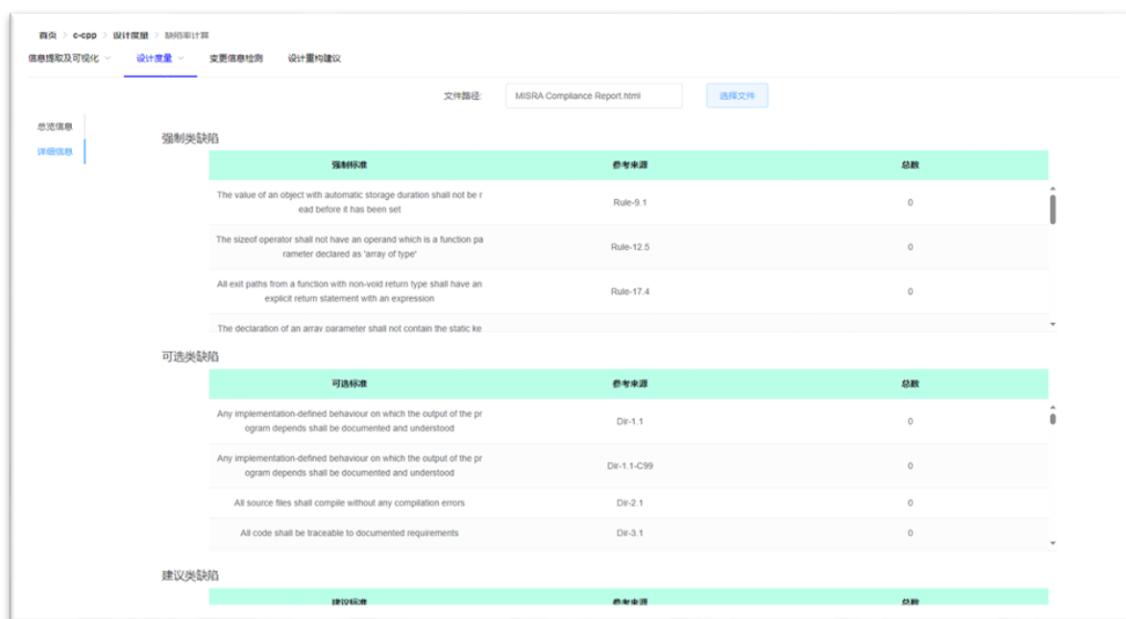


图 36 “缺陷率计算” - “详细信息” 页面

7.2 坏味率计算

进入“坏味率计算”页面，弹出弹框设置指标阈值，设置后的阈值支持导出数据，同时支持导入阈值模板。其中 C 项目、C&C++混源项目和 C++项目阈值选择部分包含的指标不同。

设置阈值后点击确认，进入页面。C 项目显示 C 信息页面，C++项目显示 C++信息页面，C&C++混源项目显示 C 信息和 C++信息页面。

“C 信息”子页面统计 C 文件信息，“统计信息”页面中包含代码坏味、DT、NPF、NCBFS、NCF、RNCFS、RNCFSBS 以及文件中代码坏味数量及坏味种类数量情况的图表；在“检测结果”中显示各个阈值对应的信息。该页面支持阈值设置。

“C++信息”子页面统计 C++文件信息，页面内容设置同“C 信息”页面。该页面支持阈值设置。

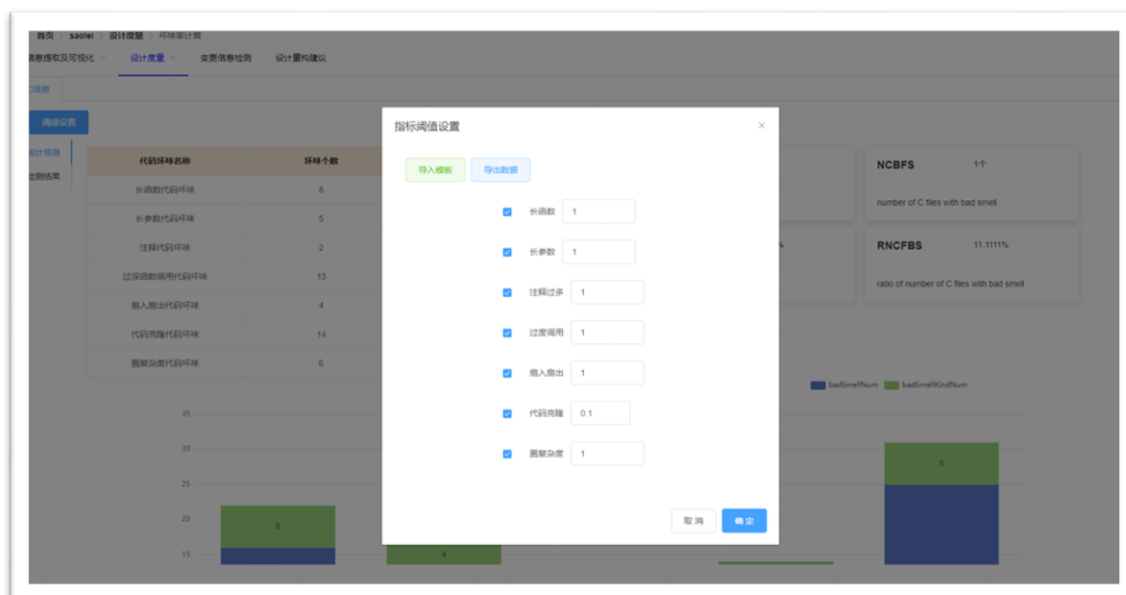


图 37 C 项目-“坏味率计算”-“阈值选择”页面

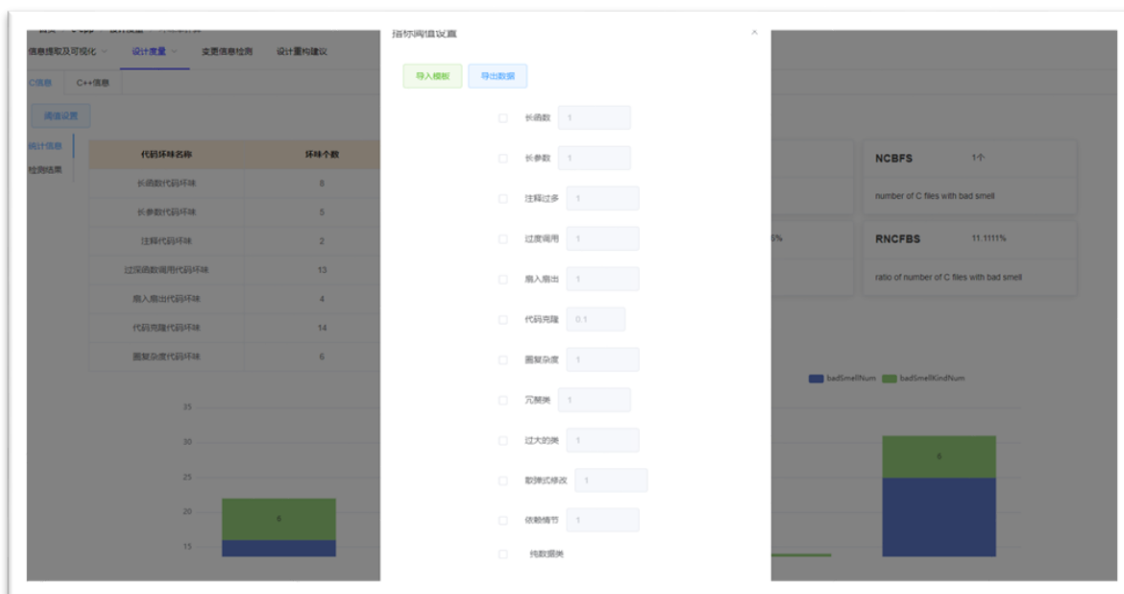


图 38 C&C++混源项目-“坏味率计算”-“阈值选择”页面

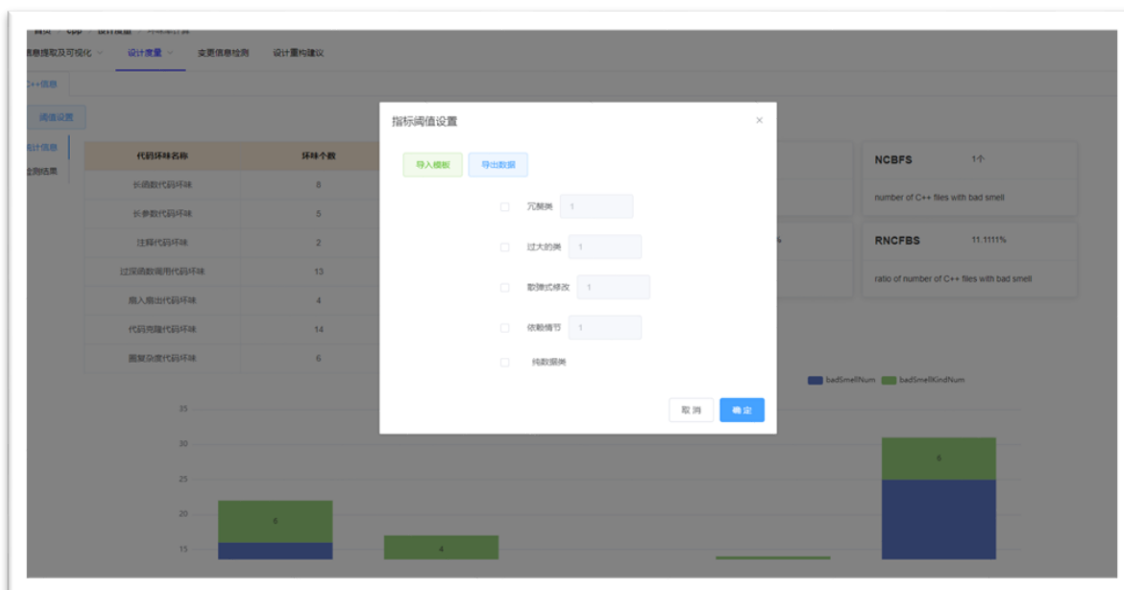


图 39 C++项目-“坏味率计算”-“阈值选择”页面

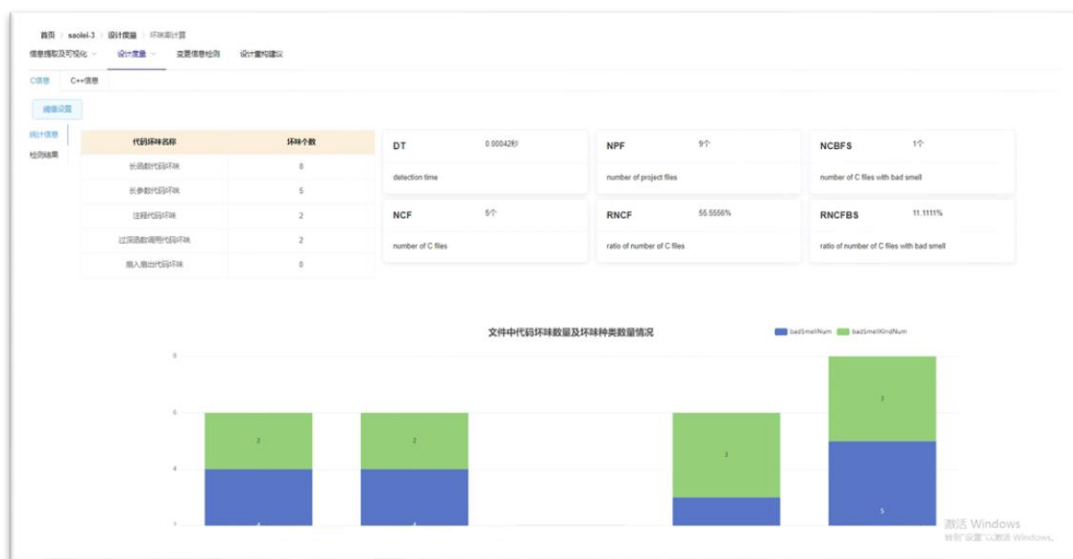


图 40 “坏味率计算” - “C 信息” - “统计信息” 页面

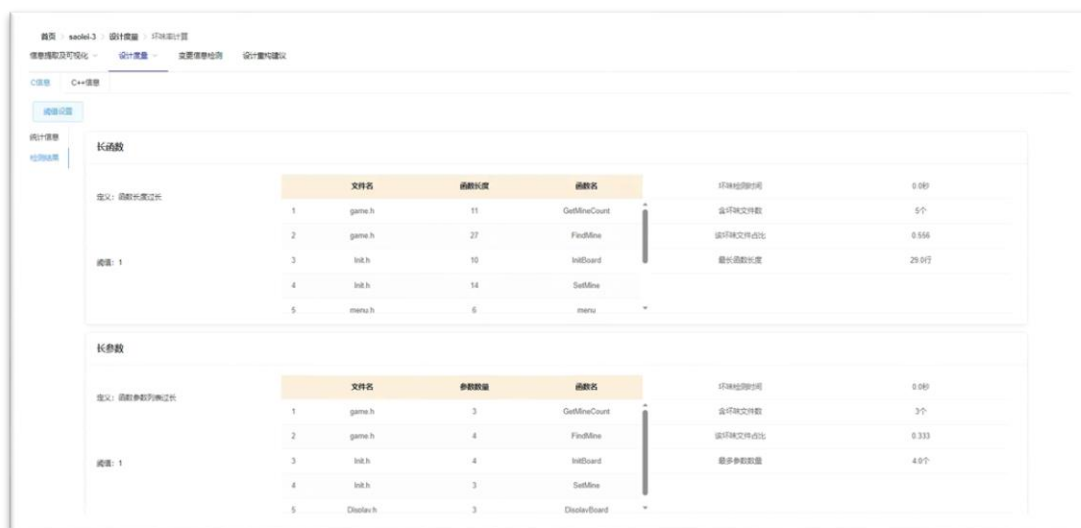


图 41 “坏味率计算” - “C 信息” - “检测结果” 页面

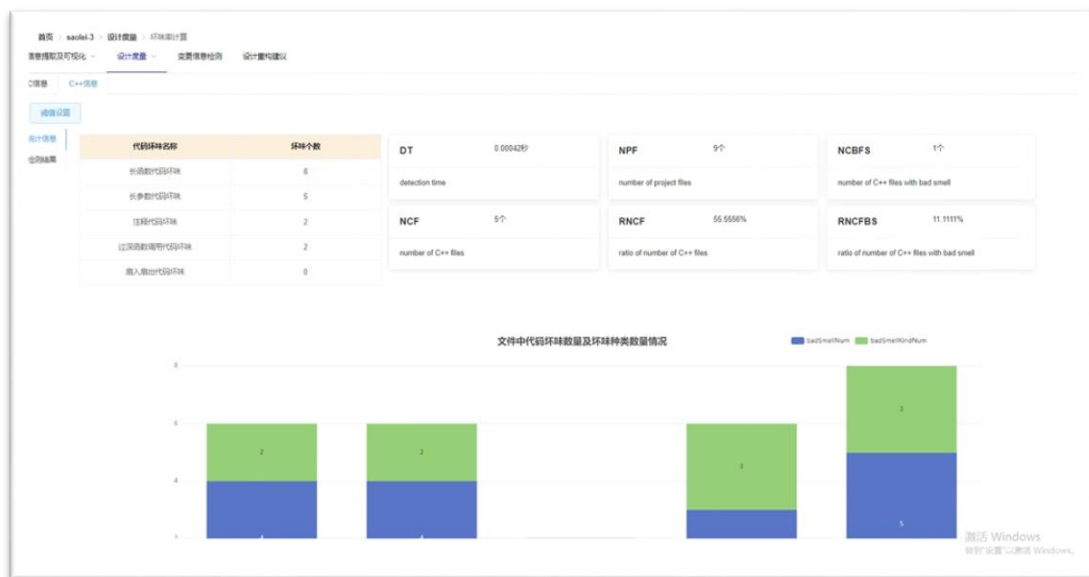


图 42 “坏味率计算” - “C++信息” - “统计信息” 页面

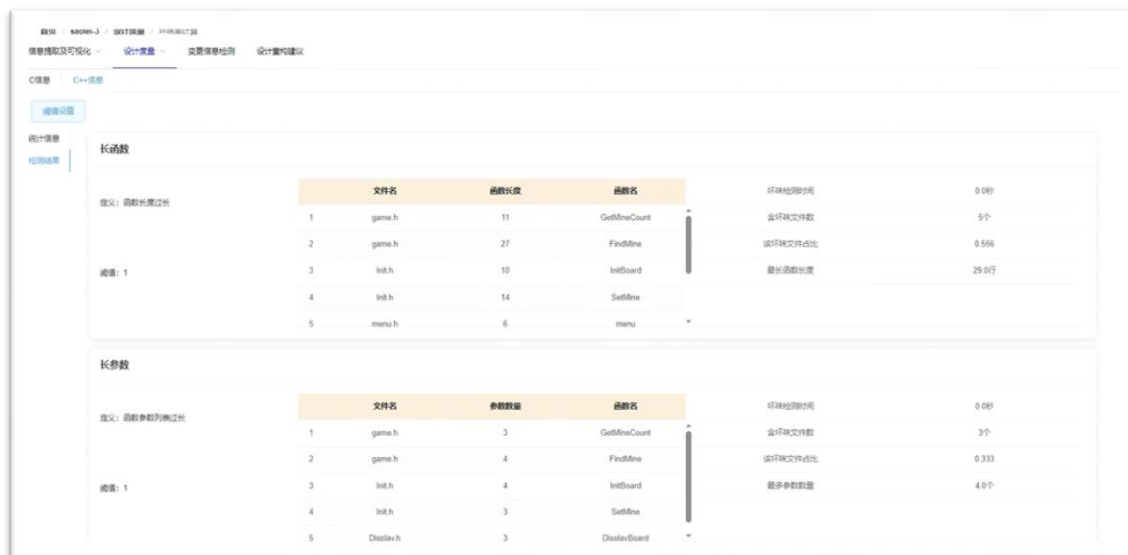


图 43 “坏味率计算” - “C++信息” - “检测结果” 页面

7.3 软件质量评估

7.3.1 指标选择

指标选择页面展示可以选择的指标，点击右侧“计算公式”可以显示该指标的公式。该页面支持导入模板以及导出数据，指标选择结束后点击“分配权重”，进入权重设置界面。

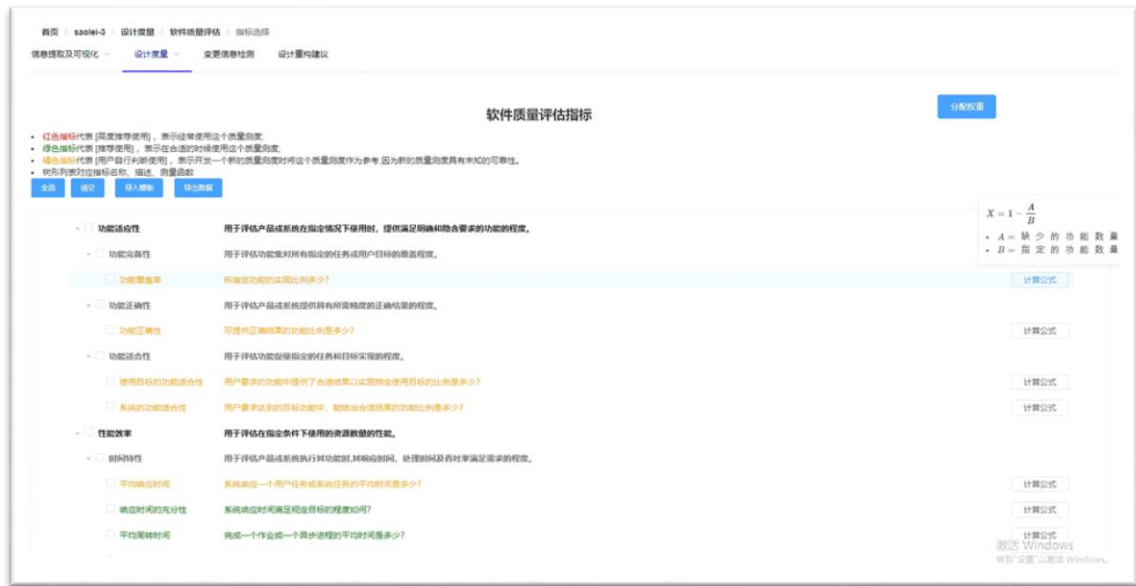


图 44 “软件质量评估” - “指标选择页”，展示计算公式

7.3.2 权重设置

在权重设置页面右上角点击“权重说明”，弹出权重说明提示。

本页面可以设置各个指标的权重，指标设置完成后，点击“校验”按钮，弹出“成功”提醒后，方可点击“下一步”按钮进入结果展示界面。如果校验失败，弹出“失败”提醒以及存在问题的指标权重，更改后需要再次校验直至成功。



图 45 “软件质量评估” - “权重说明”

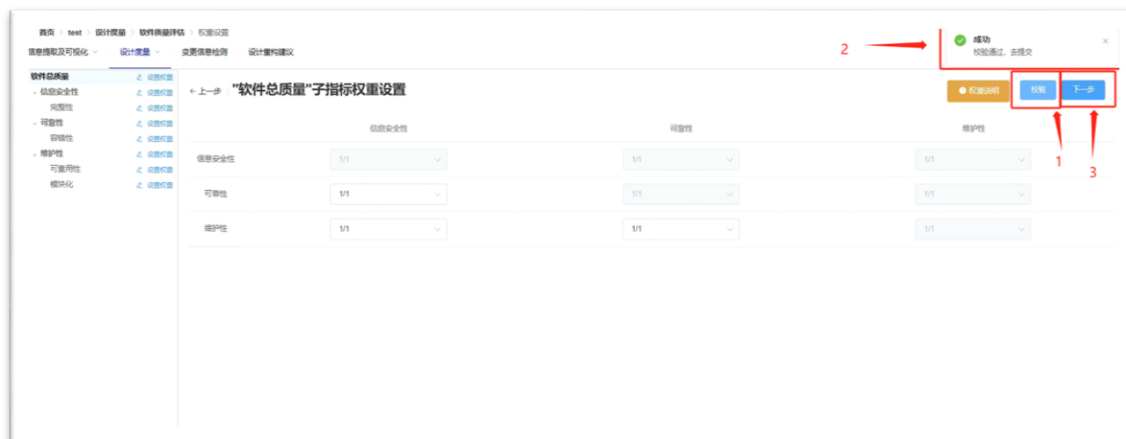


图 46 “软件质量评估” - “权重选择”，成功提示

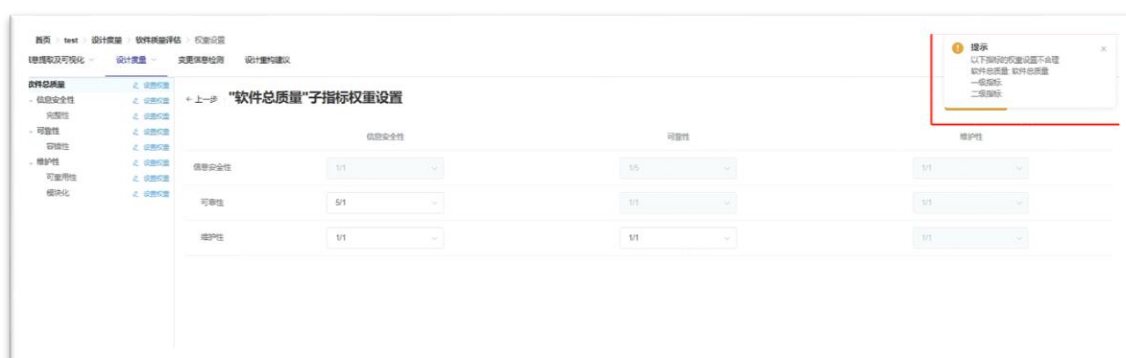


图 47 “软件质量评估” - “权重选择”，失败提示

7.3.3 结果展示

进入结果展示页面，弹出阈值设置弹框。结果展示页面部分包含计算出的 4 种层级的指标的值。右上角“选择最值”按钮支持导入和导出模板。

一级指标为软件总质量；点击一级指标右侧展示一级指标下二级指标的值和权重，点击二级指标右侧展示二级指标下三级指标的值和权重，点击三级指标右侧展示三级指标计算出的结果详情；点击四级指标右侧展示四级指标计算出的结果详情。

指标最值设置

×

导入模板

导出数据

功能覆盖率 — 3.75

最小值

0

最大值

1

使用目标的功能适合性 — 1.60

最小值

0

最大值

1

功能正确性 — 1.00

最小值

0

最大值

1

系统可用性 — 0.52

最小值

0

最大值

1

系统的功能适合性 — 1.57

最小值

0

最大值

1

平均故障通告时间 — 0.10

最小值

0

最大值

1

组件的冗余度 — 0.00

最小值

0

最大值

1

功能性的依从性 — 0.75

最小值

0

最大值

1

平均失效间隔时间(MTBF) — 1.85

最小值

0

最大值

1

图 48 “软件质量评估” – “结果展示”，阈值设置弹框

首页

总览

设计数据

软件质量评估

结果展示

数据模型及可视化

设计数据

变更信息检测

设计意图建议

软件总质量 — 0.34

可靠性 — 0.00

成熟性 — 0.00

可用性 — 0.00

容错性 — 0.00

组件的冗余度 — 0.00

易恢复性 — 0.00

可靠性的依从性 — 0.00

信息安全性 — 0.40

保密性 — 0.00

完整性 — 0.50

操作保密性 — 0.00

抗抵赖性 — 0.00

不可否认性 — 0.00

真实性 — 0.00

隐私安全性的依从性 — 0.00

维护性 — 0.53

请选择指标

设置最值

指标最值设置

×

导入模板

导出数据

取消

提交

图 49 “软件质量评估” – “结果展示”，设置最值

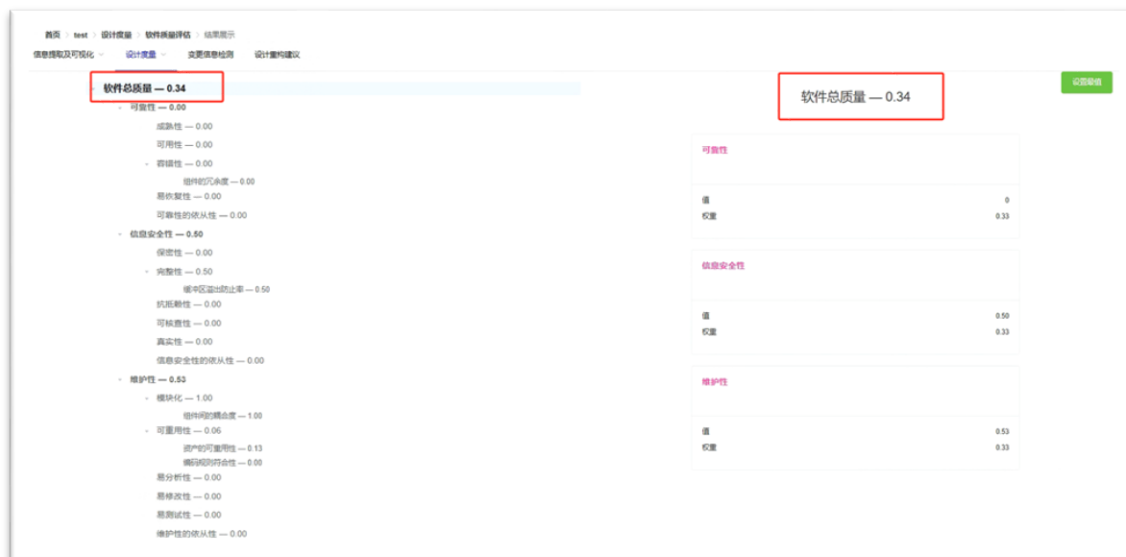


图 50 “软件质量评估” - “结果展示”，一级指标页

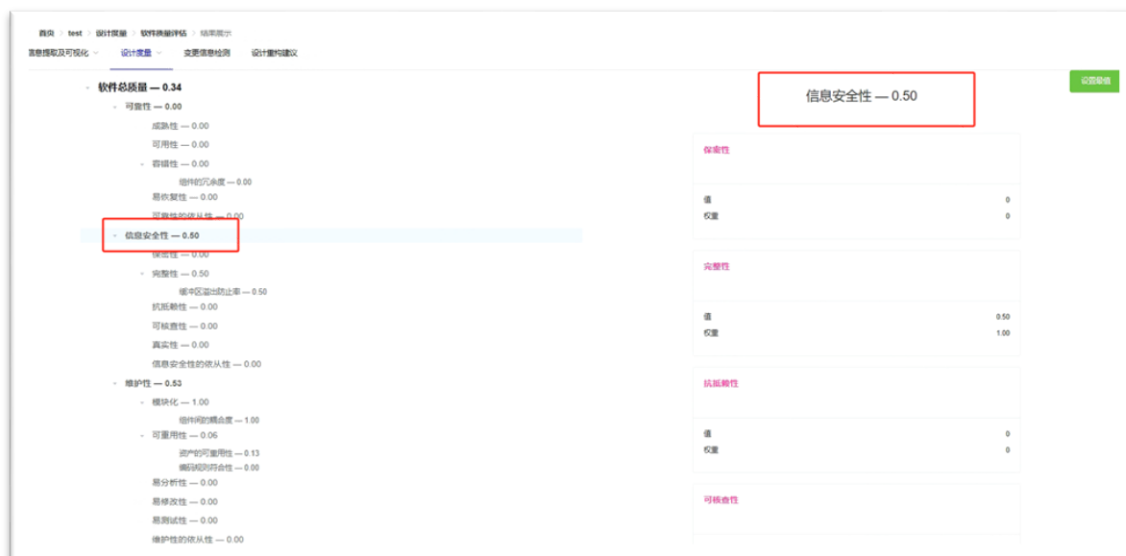


图 51 “软件质量评估” - “结果展示”，二级指标页

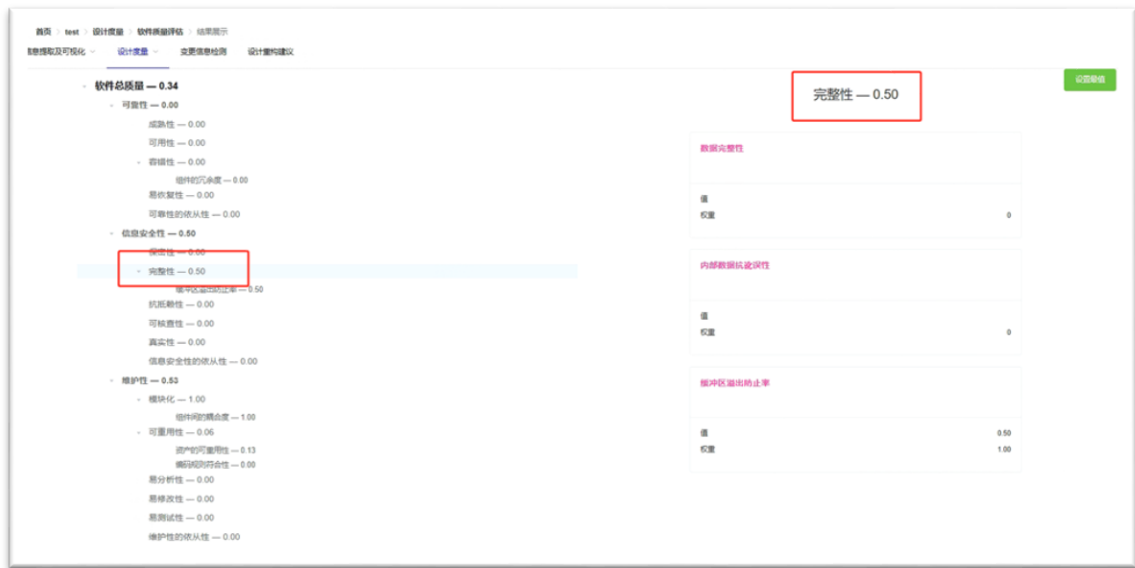


图 52 “软件质量评估” - “结果展示”，三级指标页



图 53 “软件质量评估” - “结果展示”，四级指标页

7.4 设计质量评估

7.4.1 指标选择

界面展示设计质量评估指标的名称、描述、公式以及公式因子。点击“开始评估”按钮进入结果展示页面。

设计质量评估指标					
序号	名称	描述	公式	公式因子	
1	可修改性	该指标用于描述对设计进行修改的难度。特指对设计改正和改进活动实施的难度。指标值越高，实施难度越低。	$X = 1 - \frac{\sum_{i=1}^N \frac{OD_i + ID_i}{N}}{N}$	<ul style="list-style-type: none"> OD_i = 函数 i 的出度 ID_i = 函数 i 的入度 N = 设计中函数的数量 	
2	可扩展性	该指标度量的是对设计进行扩展的难度。指标值越高，扩展难度越低。	$X = \frac{N_{API}}{N}$	<ul style="list-style-type: none"> N_{API} = 设计中 API 的数量 N = 设计中函数的数量 	
3	易测试性	该指标度量的是对软件设计进行测试的难度。指标值越高，测试难度越小。	$X = 1 - \frac{\sum_{i=1}^N \frac{OD_i}{N}}{N}$	<ul style="list-style-type: none"> OD_i = 函数 i 的出度 N = 设计中函数的数量 	
4	可替换性	该指标度量的是设计中的函数被具有相似功能的函数替换的难度，以及删除特定功能函数的难度。指标值越高，替换/删除难度越小。	$X = 1 - \frac{\sum_{i=1}^N \frac{R_i}{N}}{N}$ $R_i = \frac{1}{1 + ID_i}$	<ul style="list-style-type: none"> R_i = 函数 i 的可替换性 ID_i = 函数 i 的入度 N = 设计中函数的数量 	
5	易理解性	该指标度量的是开发人员对于设计和固有工程文件理解的难易程度。指标值越高，设计和工程文件越易于理解，越利于开发人员开展测试活动。	$X = \frac{N_{comment}}{N}$	<ul style="list-style-type: none"> $N_{comment}$ = 包含注释的函数的数量 N = 设计中函数的数量 	

图 54 “设计质量评估” - “指标选择”

7.4.2 结果展示

结果展示页面展示 API 占比、入度 vs 出度、设计质量评估概况、带注释函数占比以及函数的可替换性图表。

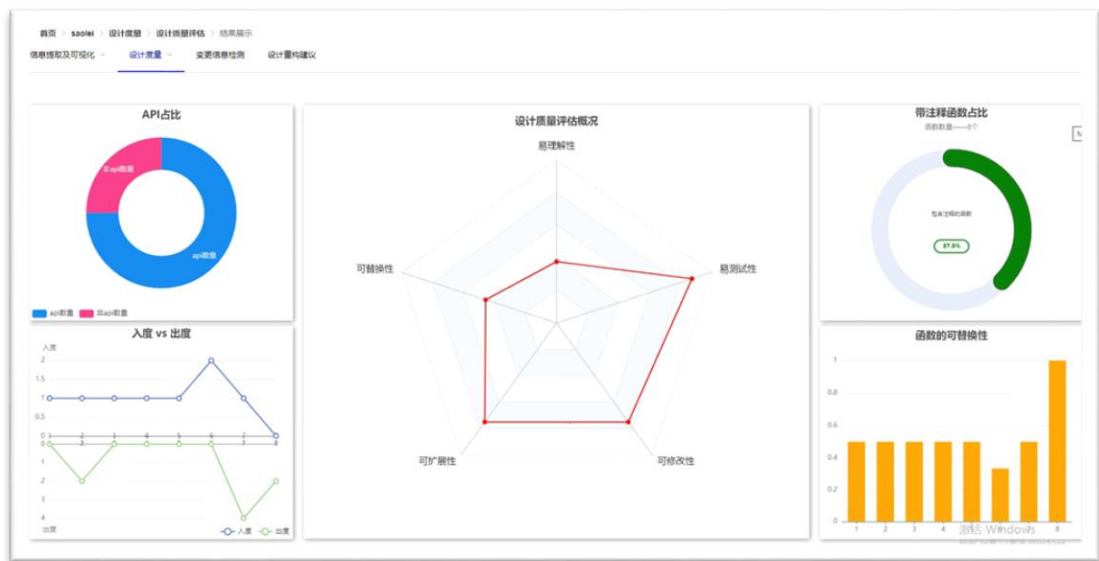


图 55 “设计质量评估” - “结果展示”

8. “变更信息检测”栏

该页面展示项目在不同版本的变更信息。进入页面后通过下拉框选择在工具中解析过的历史版本，选择后显示该项目与历史项目在系统级别（包含软件质量以及设计质量两类对比）、文件级别以及函数级别三种粒度上的对比信息。在该页面支持通过右上角的“项目名选择”切换需要对比的历史项目。支持 C 信息和 C++ 信息变更检测。

C 信息支持系统级别、文件级别、函数级别信息变更检测；C++ 信息支持系统级别、文件级别、类级别信息变更检测。

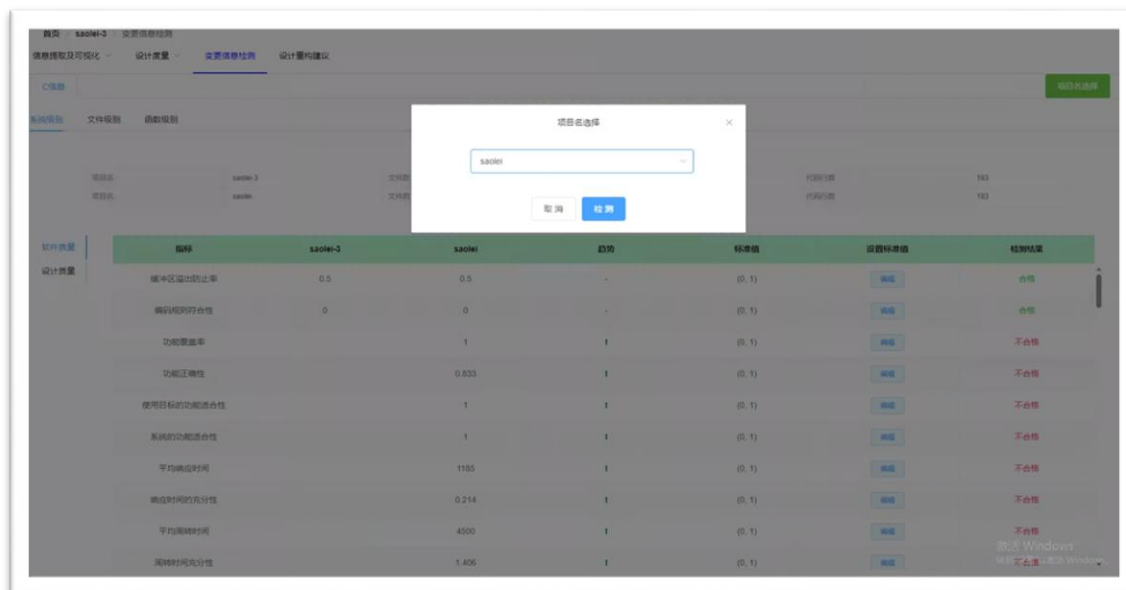


图 56 “变更信息监测”页面，选择历史项目

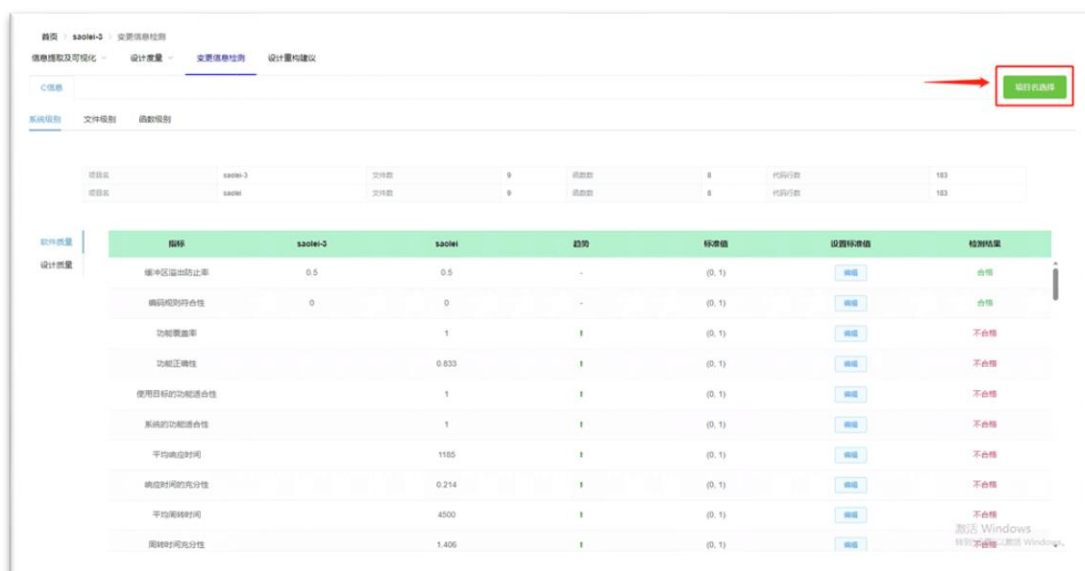


图 57 “变更信息监测”页面，右上角“项目名选择”按钮支持变更对比的历史项目

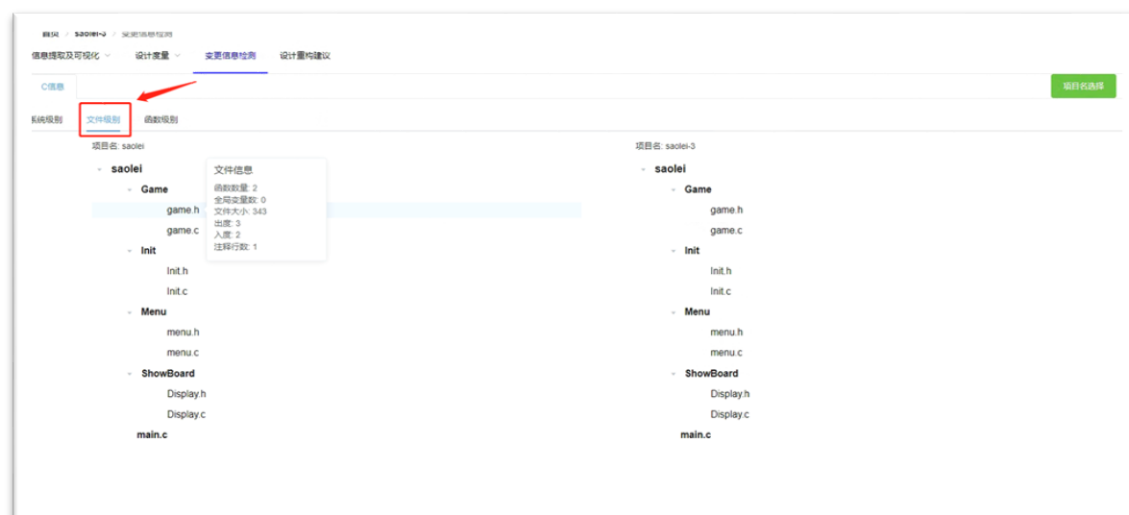
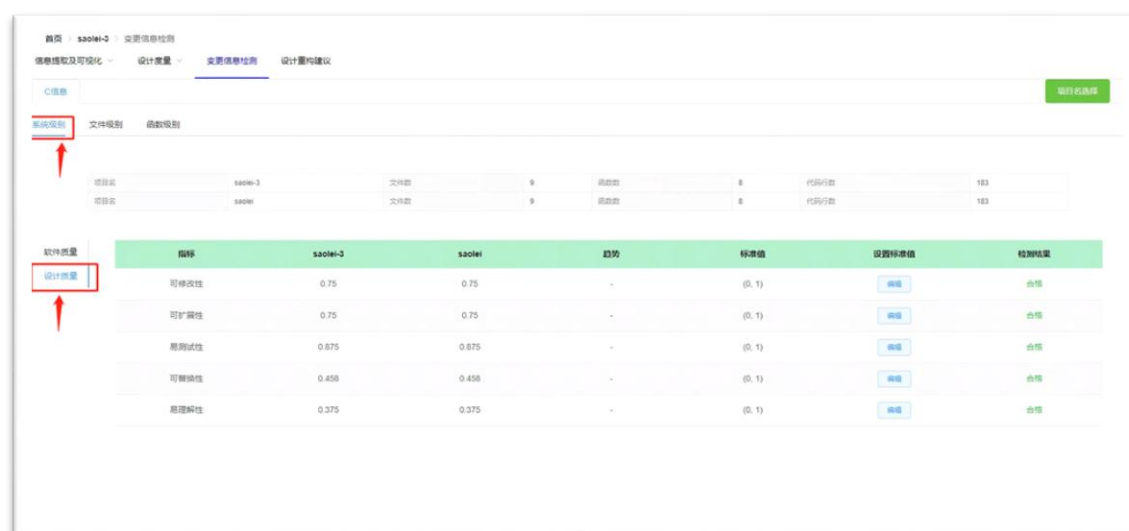
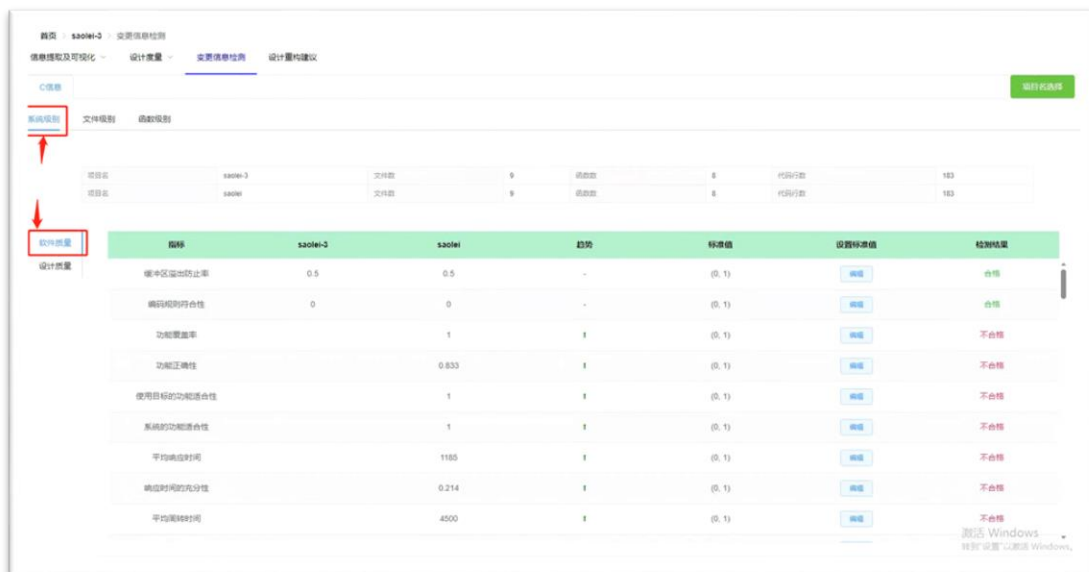


图 60 “变更信息监测” - “C 信息” - “文件级别” 页面

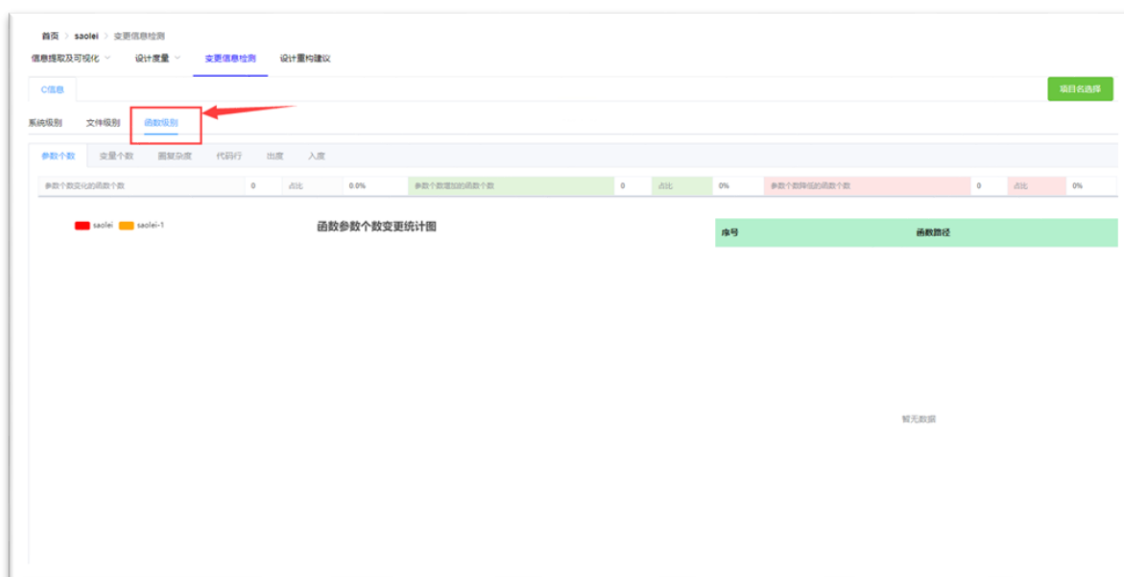


图 61 “变更信息监测” - “C 信息” - “函数级别” 页面

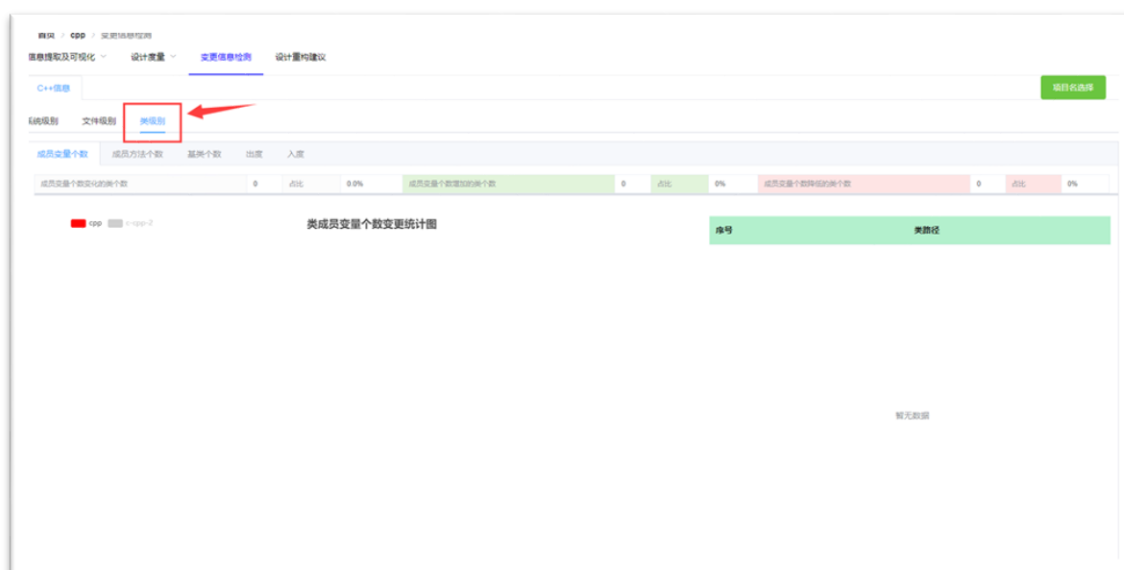


图 62 “变更信息监测” - “C++信息” - “类级别” 页面

9. “设计重构建议” 栏

“设计重构建议”栏进入后设置阈值，生成设置重构建议。设置阈值支持导入模板和导出数据。右侧信息支持查看详情，点击“详情”按钮显示具体信息。

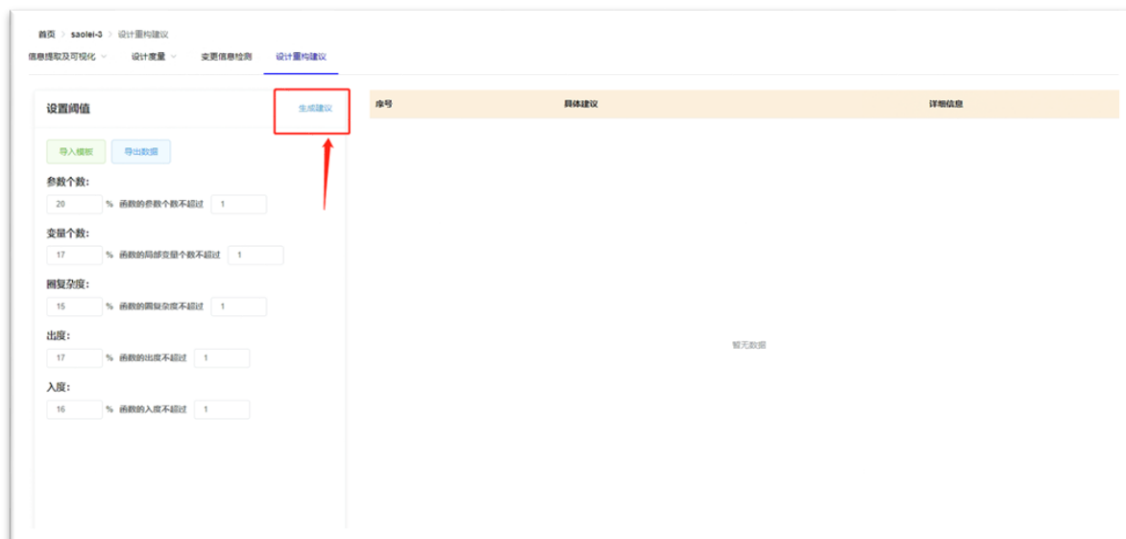


图 63 “设计重构建议”页面，设置阈值



图 64 “设计重构建议”页面，查看具体建议详情



图 65 “设计重构建议”页面，具体设计重构建议的详细信息