

阅读低代码平台Mendix之 RestServices-1 概览

一起来读开源工业软件系列2

讨论内容

- 0 本视频的观看对象和前置知识要求
- 1 阅读开源软件的流程
- 2 浏览Mendix官网
- 3 阅读Mendix GitHub代码仓库内容
- 4 在IDE里阅读Mendix之RestServices
 - 代码结构
 - 代码审查方法
 - 代码质量监测标准
 - 代码审查工具

0 视频的观看对象

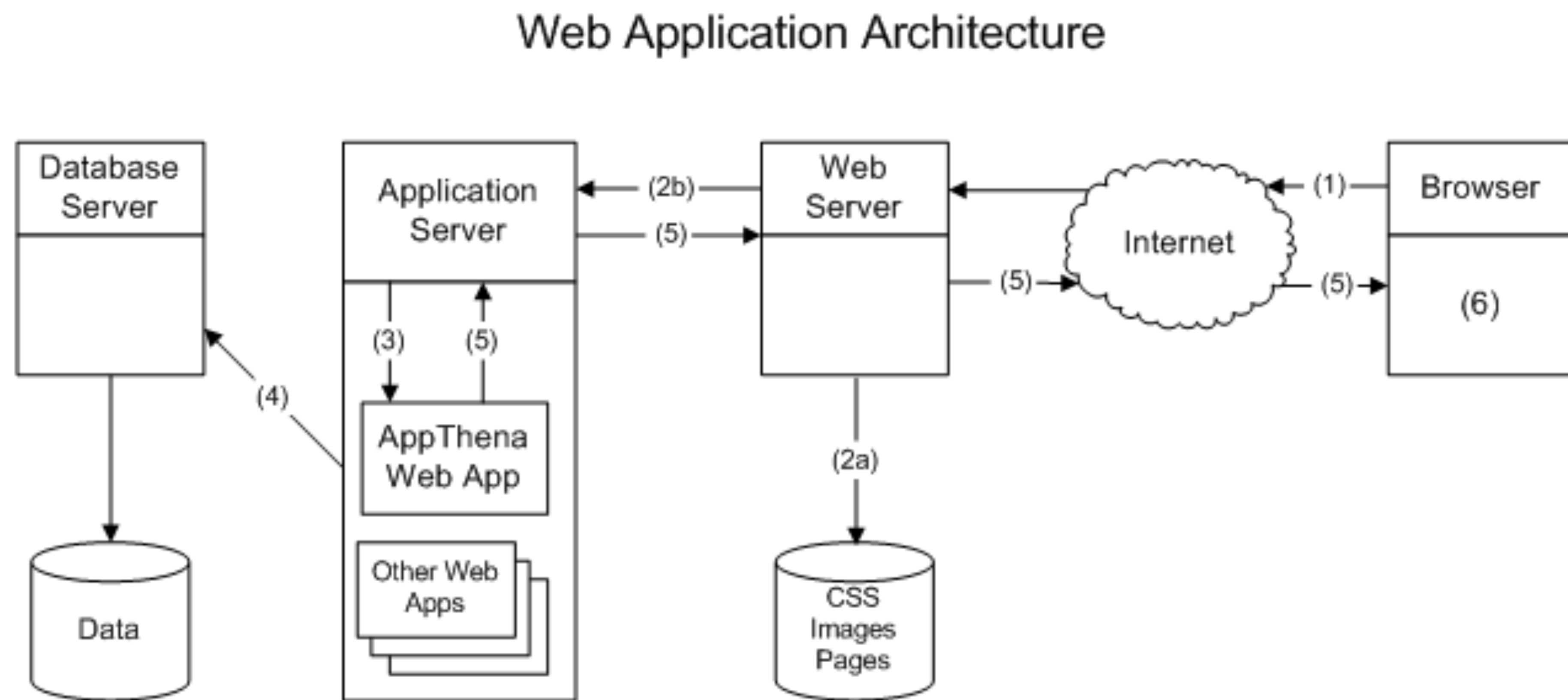
- 想学习如何阅读软件程序
- 熟悉Java语言的基础知识
- 了解Web应用系统架构
- 熟悉Java代码规范
- 了解Java开发工具

前置知识：Java语言基础

- 参考书籍《Java编程思想》（中文版） <https://github.com/apacheecn/thinking-in-java-zh>
- 基本语法、语句、类、接口与实现、 Arrays, Lists, and Structured Data
- GC、数据库应用与JDBC、Java与多线程、依赖注入
- 常见Java基础错误：Top Java Software Errors: 50 Common Java Errors and How to Avoid Them <https://stackify.com/top-java-software-errors/>

前置知识：Web应用架构

- 前端: Web页面
- 后端: 服务
- 数据库



前置知识：Java代码规范与良好实践

- [Google Java Style Guide](#)
- [Code Conventions for the Java TM Programming Language](#)
- [Google Java Programming Practices](#)
- [阿里巴巴Java开发手册\(黄山版\).pdf](#)
- <https://github.com/alibaba/p3c/tree/master>

前置知识：Java工具信息

- IDE

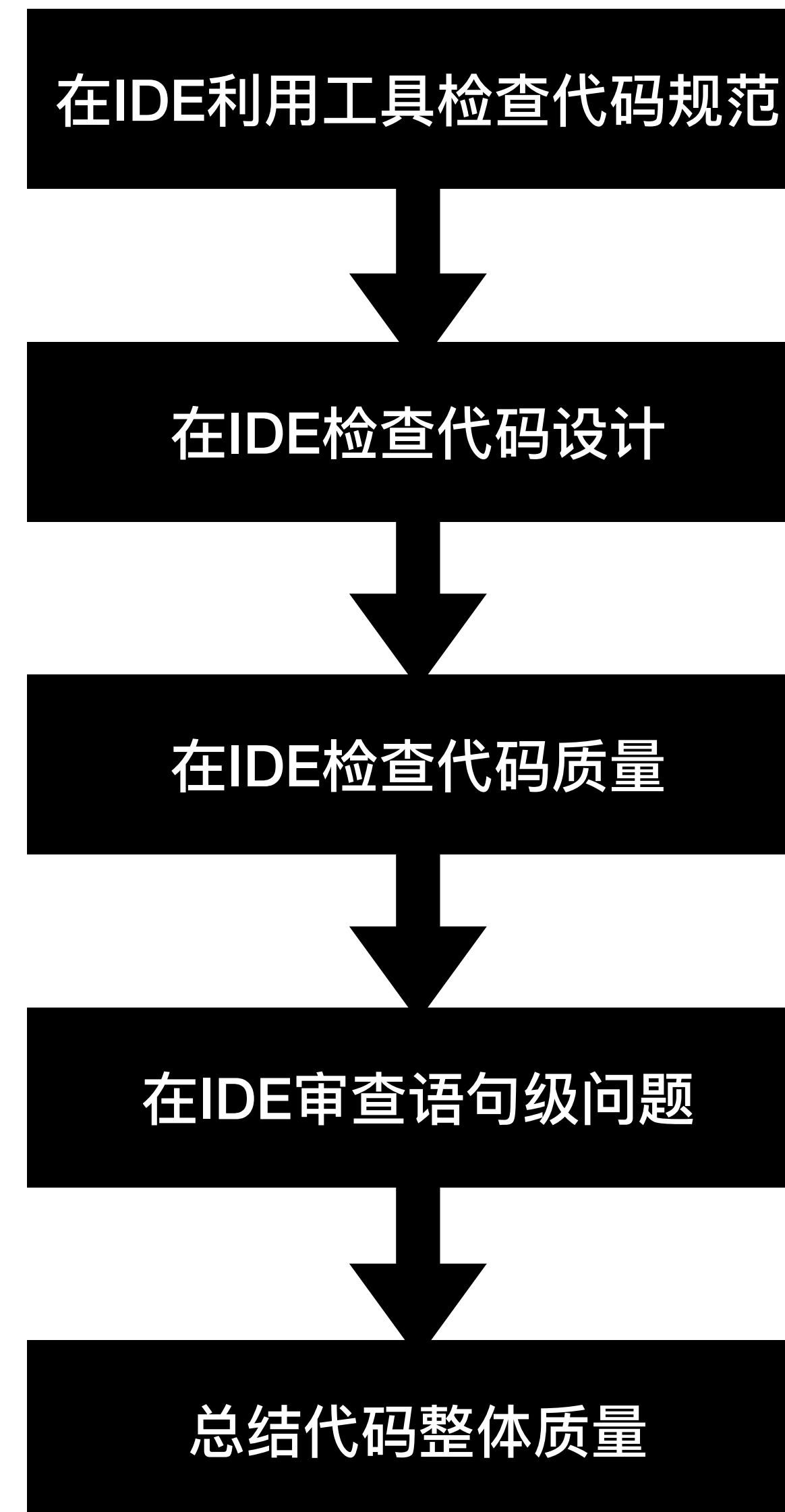
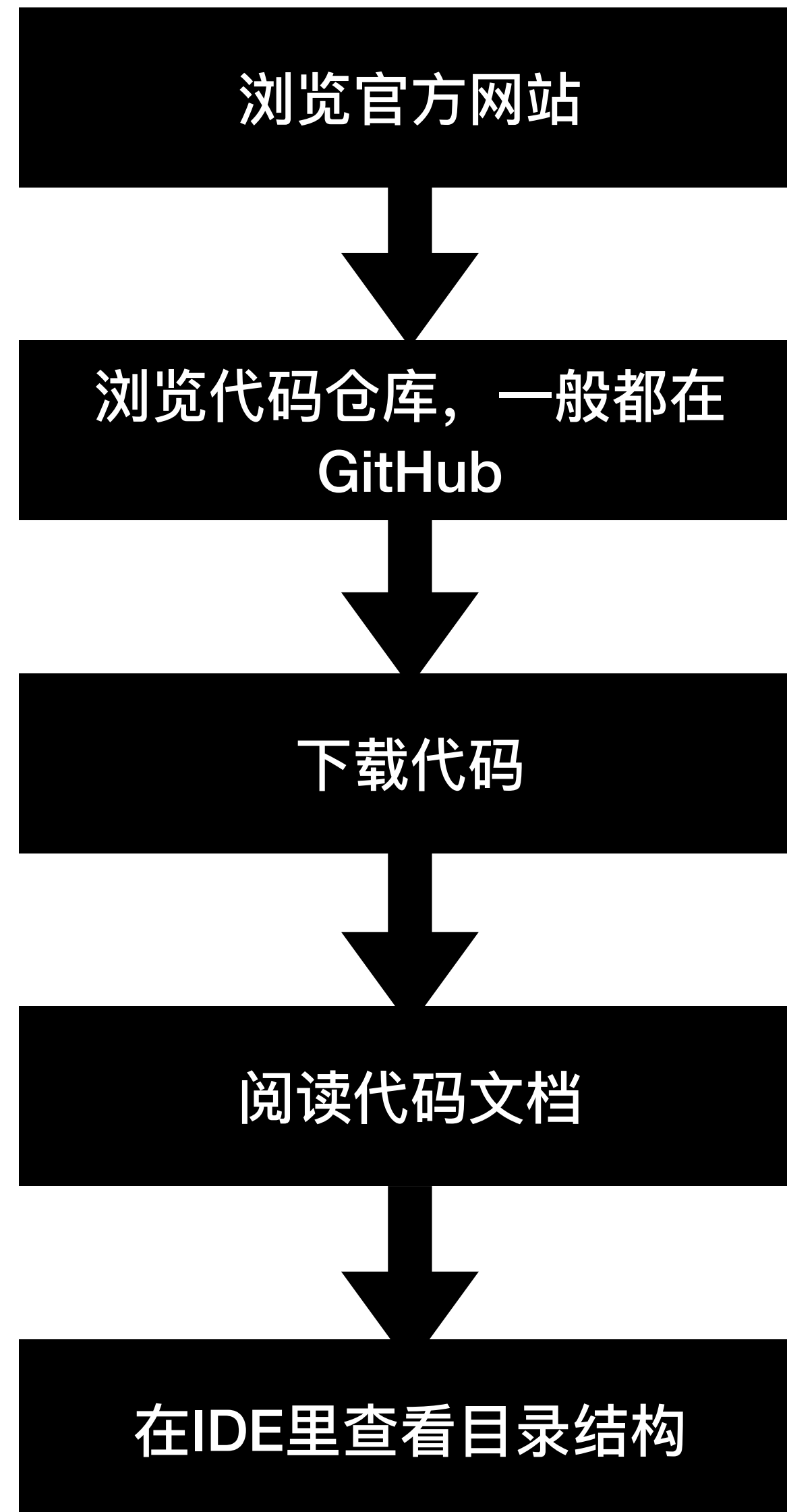
- Java 之父 James Gosling：你需要的软件可靠性越高，静态类型语言的帮助就越大 <https://www.infoq.cn/article/fhzjlty6uoscxvpltbmq>
- 从 Eclipse到IDEA，就像从金字塔到太空堡垒！ <https://cloud.tencent.com/developer/news/319900>
- 和 Eclipse 并肩十年后，我终于用IDEA 了<https://www.daimajiaoliu.com/daima/60f00c30a498400>

- 单元测试工具

- 单元测试：2021年软件测试工具总结——单元测试工具 <https://mp.weixin.qq.com/s/VKRy8cxqsfLUkTBmegHctQ>
- 在IDEA里使用JUNIT, <https://www.jetbrains.com/help/idea/junit.html>

1 阅读开源软件的流程

阅读和审查代码流程



2 浏览Mendix官网

浏览官网的要点

- 网站的结构
- 收藏经常需要查看的页面: 如案例、资源和开发者中心
- Mendix: <https://www.mendix.com/>
- 中文和英文版的选择

3 阅读Mendix GitHub代码仓库内容

代码仓库内容

- 了解代码的开源LICENSE
- 了解仓库的结构和功能
- 了解各个模块之间的关系
- 阅读代码仓库需要了解哪些语言和软件系统知识
- <https://github.com/mendix>

4 在IDE里阅读Mendix之 RestServices

RestServices

- REST service module for Mendix. Supports consuming and publishing REST based services and real-time data synchronization. Supports JSON, form-encoded, multipart and binary data transport. Mendix的REST服务模块。支持消费和发布基于REST的服务和实时数据同步。支持JSON、表格编码、多部分（或多段）和二进制数据传输。
- 文档链接：
 - <https://docs.mendix.com/refguide/integration/>
 - <https://docs.mendix.com/refguide/integration/rest-services/>
 - <https://docs.mendix.com/refguide/consumed-rest-services/>
 - <https://docs.mendix.com/refguide/published-rest-services/>
- 文件格式 <https://forum.mendix.com/p/questions/11540474045285303>
 - MPK: MPR + bundles files (JS, Java, Themes, data snapshot)
 - Can be used for exporting and sharing of full projects but also single Documents or Modules
 - MPR: The Mendix Model definition
 - MDA: Deployment package

4.1 RestServices代码结构

	主要内容	文件格式
DIST	发布版本	.mpk mendix package包含一个Mendix项目或模块
docs	文档	.md
images	图片	.png
javasource	communitycommons,restservices,system,tests,unittesting	.java
antisamy	用于防止Web应用程序中的跨站点脚本攻击（XSS）和其他安全漏洞	.xml，包名为resources.communitycommons
theme	主题风格	css,scss,json,ico,html
userlib	java依赖库	.jar
widgets	Web应用程序中的小型组件	.mpk
LICENSE	Apache 2.0	
pom.xml	Maven项目的核心文件之一，包含了项目的基本信息和依赖关系，可以帮助开发者自动化地管理项目的构建和依赖关系，提高开发效率和代码质量。	
README.md	代码说明，包含Getting Started、Consuming REST services、Publishing REST services、(Real time) data synchronization、About JSON serialization、About JSON deserialization、Working with files / binary data、Overview of all functions、Notes for contributors、Changelog	
RestServices.mpr	The Mendix Model definition	

4.2 代码审查方法：选择被审查的代码

- 正确性：
 - 选择中心组件中的核心代码；
 - 运行复杂性分析工具，对较复杂的代码进行审查；
- 效率：
 - 运行分析器，审查CPU时间用得较多的代码；
- 可靠性：
 - 呈现出高错误率的代码；
 - 缺乏经验的程序员所写的代码；
 - 。 。 。 。 。 。

Code review

CHECKLIST

Use this form to help you perform a code review.

About the code

Reviewed by: _____
Date: _____
Language: _____
Number of files: _____

Module name: _____
Version reviewed: _____
Code author: _____

Automated inspection

☐ The code compiles without errors
☐ The code compiles without warnings
☐ There are unit tests
☐ They are sufficient (include all boundary cases, etc.)
☐ The code passes them

☐ The code is kept under source control
☐ The code has been tested with inspection tools
Tool name _____ Results _____

☐ Continue to next section ☐ Stop review here

Design

☐ The code is complete (against its specification)
☐ There is a good choice of algorithms
☐ Optimizations are necessary and appropriate
☐ Any missing functionality is marked clearly in the code

General observations about the code's design
☐ The code is well structured
☐ There is design documentation
☐ The code matches the documentation

☐ Continue to next section ☐ Stop review here

General code comments

Style
☐ The code layout is clear
☐ It follows project style guidelines
☐ There is a good (unambiguous) public API
☐ There is a good choice of names

Defensive programming
☐ Array access is guarded and safe (C/C++)
☐ There is a correct choice of types
☐ All input is validated
☐ There is no use of compiler-specific features

General comments

Error handling
☐ Error conditions are routinely handled
☐ Assertions are used to validate logic
☐ The code is exception safe
☐ Errors are propagated, not hidden
☐ There are no resource leaks

☐ The code uses multiple threads
☐ It is thread safe
☐ There isn't potential for deadlock

Structure
☐ There is no redundant code
☐ There is no cut-and-paste programming

☐ Continue to next section ☐ Stop review here

Statement-level review

Fill out the table below, and move on to a new sheet as required. Rate issues on a scale from 0 (cosmetic/nice to have) to 5 (must fix).

File	Line	Issue	Rating
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

Continue on a separate sheet (or mark up a paper copy of the code)

Follow-up

Record the outcome of the review here

Conclusion:
☐ Code OK
☐ Rework and verify
☐ Rework and re-review

Complete work by: _____
Assigned verifier: _____

代码审查：关于代码、自动检查和设计

检查项目	结果	说明
模块名	DB代理申请	
审查人员	张三李四	
版本	1.0	
日期	2022-07-04	
语言	JAVA	
作者	王二	
文件数	1	

2.自动检查

检查项目	结果	说明
代码通过编译，没有任何错误	是	
代码通过编译，没有任何警告信息	是	
存在单元测试	否	
代码处于源代码控制之下	是	
使用检查工具对代码进行了检查	是	
工具名称	P3C	
结果	CRITICAL (0)	

3.设计

检查项目	结果	说明
代码已完成（相对于规范来说）	是	
选择的算法很好	不适合	
采用了恰当的设计模式	否	
任何缺失的功能都已在代码中清晰的标出	不适合	
代码的结构很好	是	
存在设计文档	是	
代码与这些文档相符	是	

代码审查4：对代码编写质量的总体意见

风格

检查项目	结果	说明
代码的布局很清晰	是	
代码遵循项目的风格准则	是	
存在良好（无歧义）的公共API	是	
选择的名称都很好	是	

防御性编程

检查项目	结果	说明
数组访问得到了保护（C/C++）	不适用	
数据类型选择正确	否	
所有输入都有效	是	
不存在特定编译器功能（C/C++）	不适用	

错误处理

检查项目	结果	说明
错误情况都得到了处理	是	
使用断言来验证逻辑	否	
代码是异常安全(exception safe)的	是	
错误将被传播，而不会被隐藏	是	
本地有日志文件	否	
没有任何资源被泄露	未知	
代码使用多线程	不适用	
它是线程安全的	不适用	
不会发生死锁	是	

结构

检查项目	结果	说明
不存在冗余的代码	是	
不存在剪切粘贴式的编程	否	

代码审查：清单和跟踪调查

5.代码审查 清单

语句级审查

文件	行	问题	说明
DBAgentServiceImpl	223	if条件判断语句过长	
DBAgentServiceImpl	1171	switch语句可以优化	
DBAgentServiceImpl	2502	if条件判断语句过长	
DBAgentServiceImpl	2537	使用System.err输出，未用log	

6.跟踪调查

结论

检查项目	结果	说明
代码不错	是	
重新编写并验证		
重新编写并重新审查		

在这里记录审查的结果

检查项目	结果	说明
审查		
校验		

4.3 代码质量监测指标

- 指标体系 http://www.mccabe.com/iq_research_metrics.htm
- MOOD Metrics for Object-Oriented Design
 - 介绍 <https://www.aivosto.com/project/help/pm-oo-mood.html>
 - 参考 <https://www.javatpoint.com/mood-factors-to-assess-a-java-program>
- 指标
 - Method Hiding Factor (MHF)
 - Attribute Hiding Factor (AHF)
 - Method Inheritance Factor (MIF)
 - Attribute Inheritance Factor (AIF)
 - Coupling Factor (COF)
 - Polymorphism Factor (POF)

代码复杂度： 认知复杂度和圈复杂度

- cognitive-complexity 认知复杂度
 - Nested conditional and iterations Iterations Conditionals with more than one different logical operators
 - Refactoring Code to Pass Sonar Qube's Cognitive Complexity Checks: <https://levelup.gitconnected.com/refactoring-code-to-pass-sonar-qubes-cognitive-complexity-checks-8433433d83a1>
 - What is Cognitive Complexity in sonar report?: <https://stackoverflow.com/questions/52741725/what-is-cognitive-complexity-in-sonar-report>
 - Cognitive Complexity Lab : <https://hs.umd.edu/politicalcognition/labs/cognitive-complexity.php>
- cyclomatic complexity: 程序的圈复杂度是对各个独立的逻辑路径的测量。它也确定了以确保每个程序的语句至少执行一次所需的测试用例数量
 - 采用巧妙的测试数据，在用例数量较少的情况下也可以实现同样的覆盖度。
 - 只包含一个语句序列的程序的圈复杂度为1。每个能影响程序流程的逻辑成分（理论上被称为谓词），例如if、while和for的参数，都会给该值加1。

代码复杂度：CogV, ev, iv, v

名称	定义
Essential Complexity Metric (ev(G))	基本复杂度（ev(G)）是衡量一个模块包含非结构化构造的程度。这个指标衡量结构化的程度和代码的质量。它被用来预测维护工作并帮助模块化进程。
Cyclomatic Complexity Metric (v(G))	循环复杂性(v(G))是衡量一个模块的决策结构的复杂性。它是线性独立路径的数量，因此，应该测试的最小路径数量。
Module Design Complexity Metric (iv(G))	模块设计复杂性(iv(G))是设计还原后的模块的复杂性，反映了模块对其直接下属模块的调用模式的复杂性。这个指标区分了那些会使任何程序的设计严重复杂化的模块和那些仅仅包含复杂计算逻辑的模块。它是计算程序设计和集成复杂度（S0和S1）的基础。
Cognitive Complexity (CogV(G))	是一种用于衡量代码可读性和可维护性的指标。它基于认知心理学的原理，通过分析代码中的控制流程和条件语句等因素，来评估代码的复杂度和难度。计算方法比较复杂，它考虑了代码中的条件语句、循环语句、异常处理、递归等因素，并对它们进行了加权。

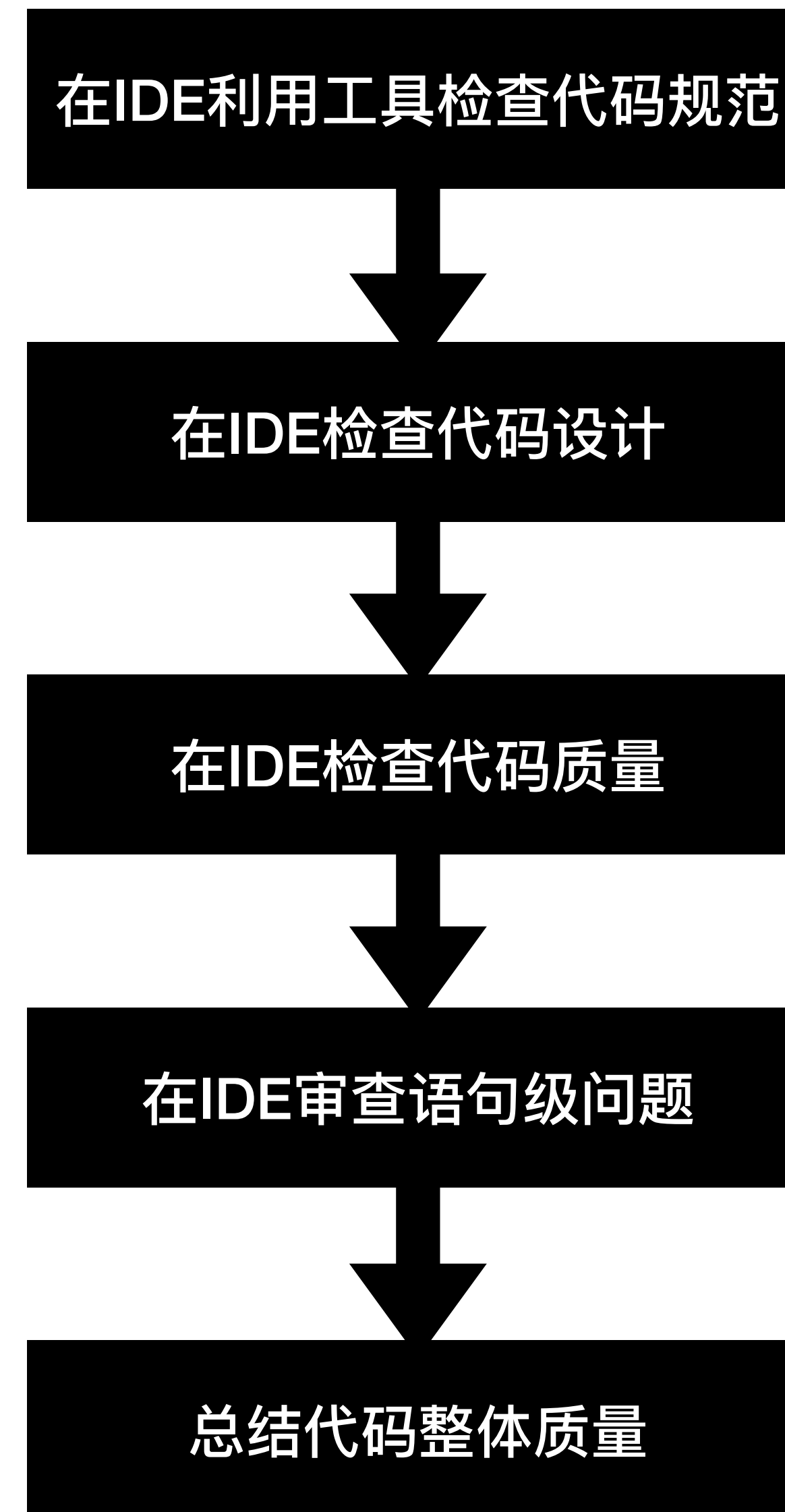
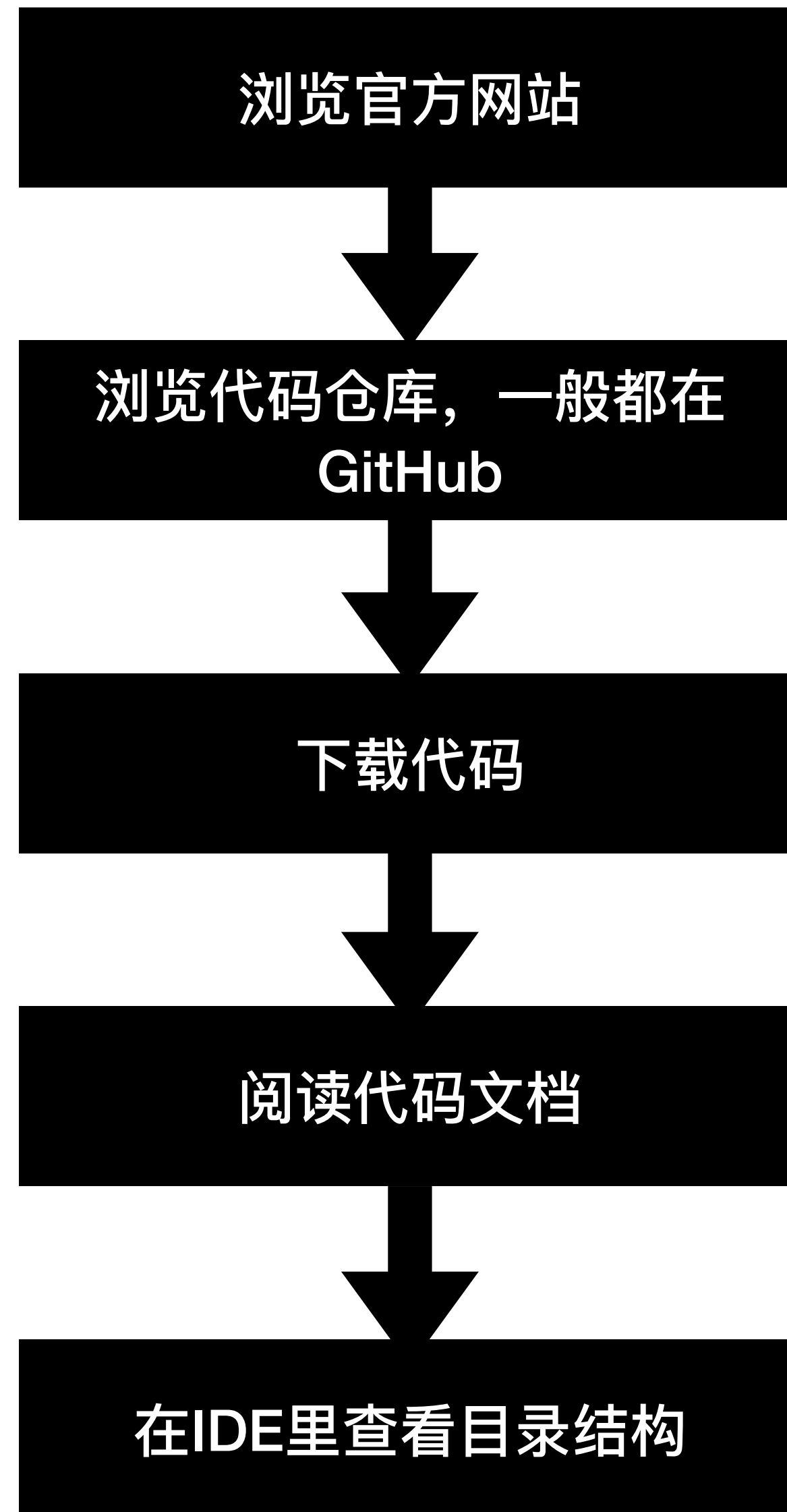
4.4 代码审查工具

- 统计工具Statistic: <https://plugins.jetbrains.com/plugin/4509-statistic>
- 代码规范工具P3C: <https://github.com/alibaba/p3c/tree/master>
- 代码复杂度工具
 - MetricsReloaded: <https://plugins.jetbrains.com/plugin/93-metricsreloaded>
 - CodeMetrics: <https://plugins.jetbrains.com/plugin/12159-codemetrics> 在程序中显示复杂度

总结

- 0 本视频的观看对象和前置知识要求
- 1 阅读开源软件的流程
- 2 浏览Mendix官网
- 3 阅读Mendix GitHub代码仓库内容
- 4 在IDE里阅读Mendix之RestServices
 - 代码结构
 - 代码审查方法
 - 代码质量监测标准
 - 代码审查工具

阅读和审查代码流程



建议练习

- 用统计工具Statistic了解代码构成和规模： <https://plugins.jetbrains.com/plugin/4509-statistic>
- 用代码规范工具P3C检查代码： <https://github.com/alibaba/p3c/tree/master>
- 用代码复杂度工具检查代码复杂度
 - MetricsReloaded： <https://plugins.jetbrains.com/plugin/93-metricsreloaded>
 - CodeMetrics： <https://plugins.jetbrains.com/plugin/12159-codemetrics> 在程序中显示复杂度

下一节，
详细阅读RestServices代码