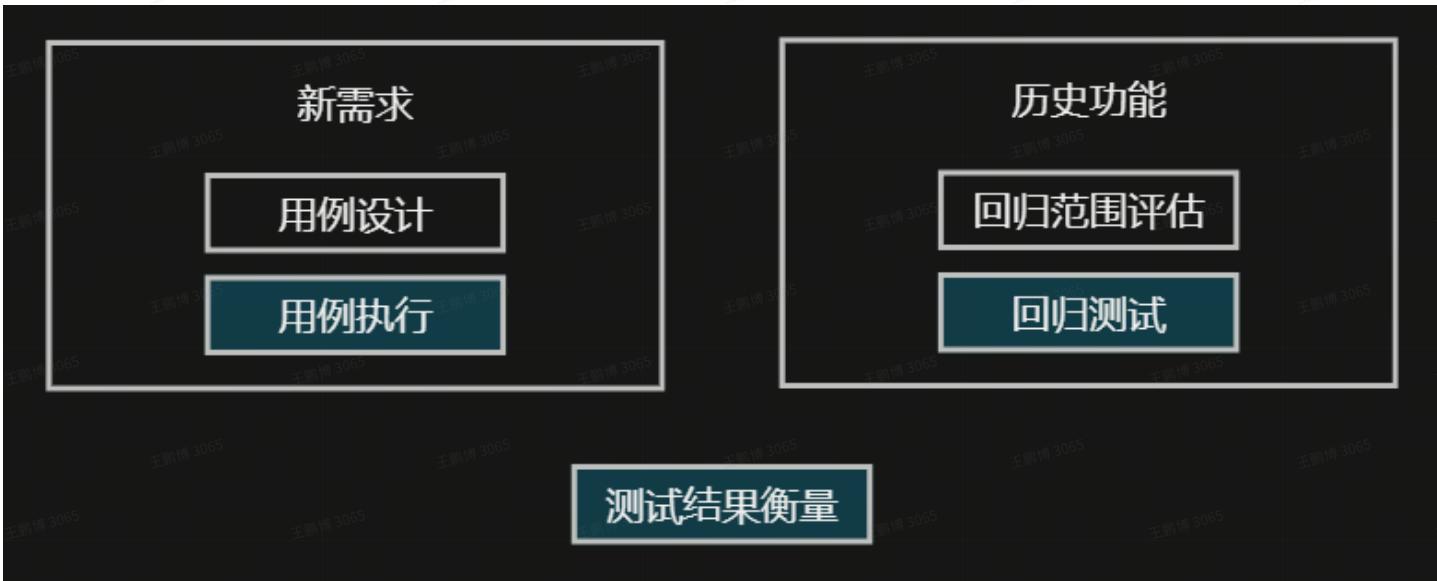


01.精准测试蓝图

测试现状



目前现状：无论新需求还是历史功能，一些场景，测试同事借助自动化测试工具、脚本或程序模拟用户与应用程序进行交互，然后评估和验证程序的功能、性能和稳定性。

自动化测试用例是否可以解决全部问题？显然不能。但是没有自动化测试是不可以的，所以我们需要让自动化测试变得更加优秀，如何评价自动化测试呢？指标如何定义？

- 自动化测试用例 **覆盖率** 如何？覆盖率低~
- 自动化测试用例 **执行成功率** 如何？
- 自动化测试用例 **执行效率** 如何？
- 自动化测试用例 **执行耗时** 多久？

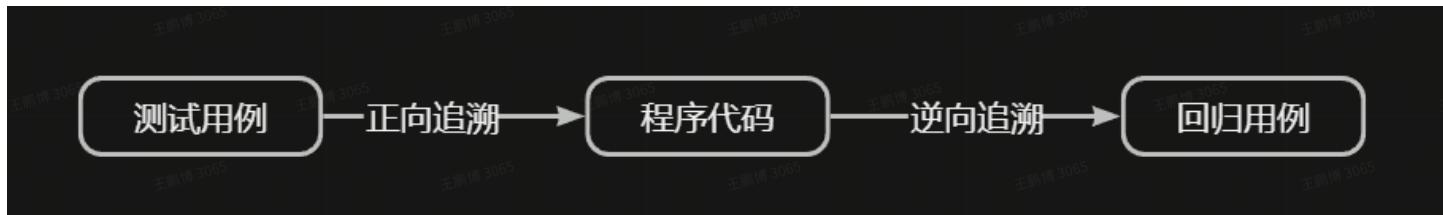
核心思想

关于精准测试技术发展、背景、历史不做过多阐述，可以阅[“精准测试技术十年发展漫谈 · TesterHome”](#)文章。

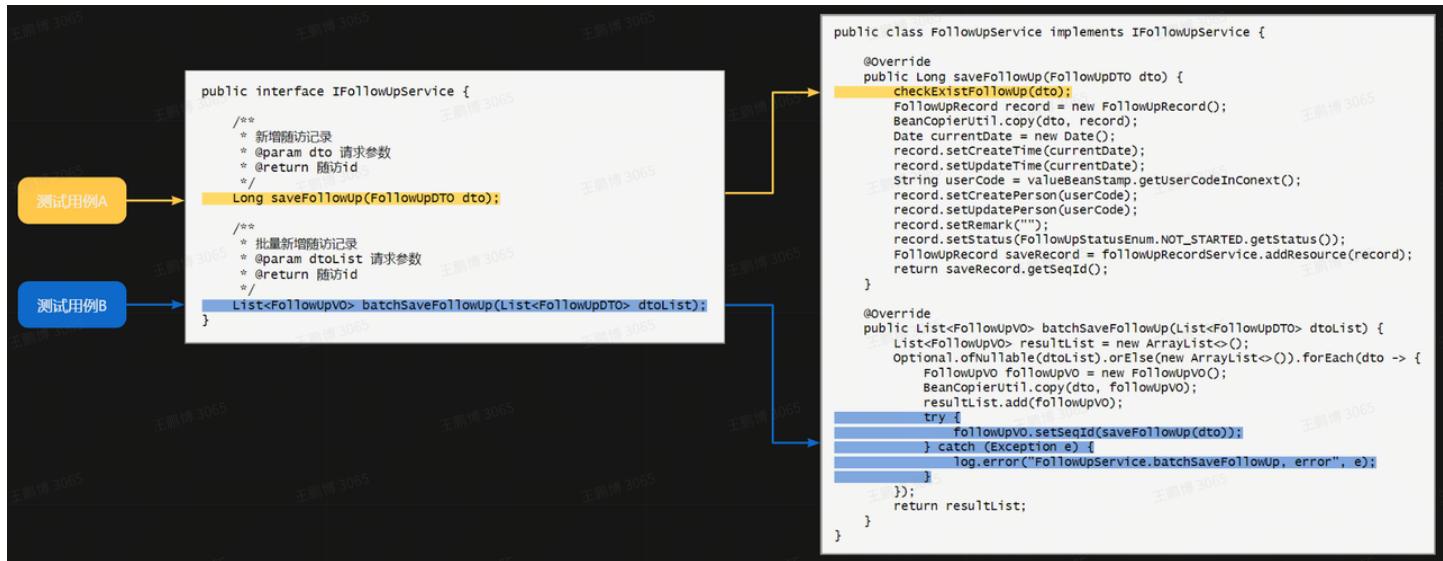
为什么引入精准测试？换句话说，精准测试解决了什么问题？

- **我要测什么** --- 代码变更影响范围内的测试用例可以自动筛选

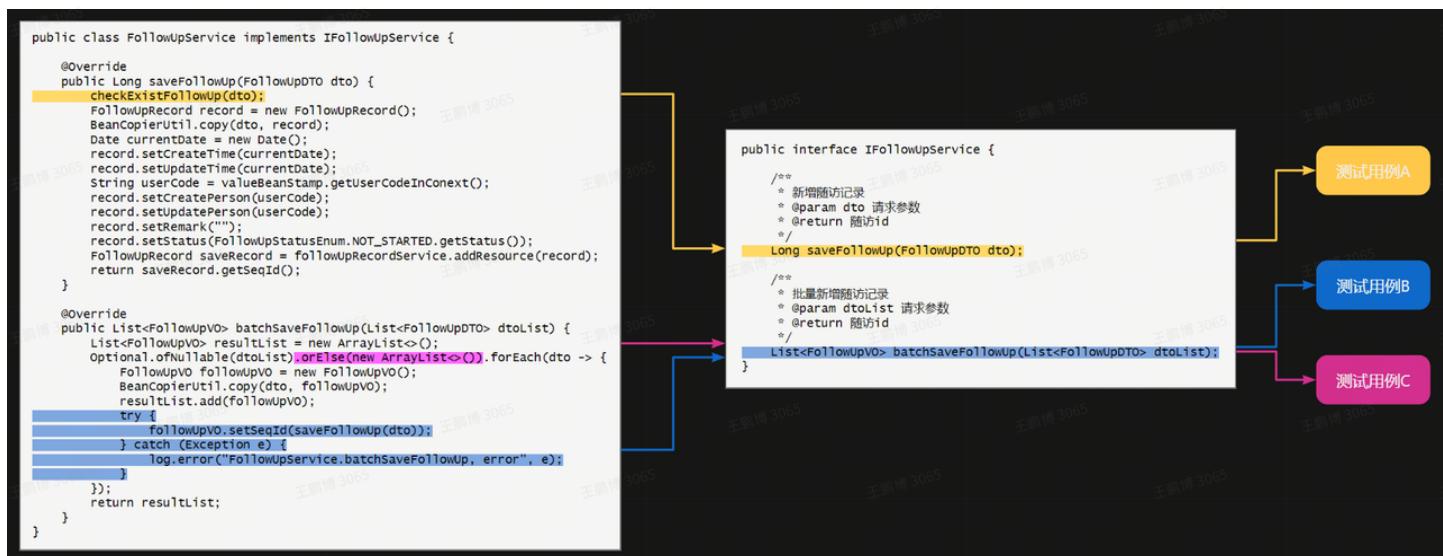
• 我测的怎么样 --- 测试全量或增量覆盖率情况如何？还需要补充哪些用例？



正向追溯：通过测试用例执行后的覆盖率建立测试用例与代码的关联关系，并计算当前版本测试的总体覆盖率，通过覆盖率结果评估“我测的怎么样”。



逆向追溯：通过对指定方法或版本、分支进行代码diff计算找出差异代码，再根据全链路分析找出受影响的其他服务/接口/方法；通过代码和测试用例的关联关系确定用例回归范围，从而准确评估“我要测什么”。



精准测试用一句话概括就是：测试用例和代码之间的追溯（精准测试的本质）。

开源框架：暂无！！！暂无！！！暂无！！！之所以强调三遍，是因为真的没有，所以在自研实现复杂度和难度会很高~

因为没有开源框架可以直接使用，所以以下设计都是基于自研的情况下进行方案梳理~

同时，在初期阶段，顺带也可以去探索些提高开发-测试效率的其它工具~

产品设计

首页

源版本

未选择文件

目标版本

未选择文件

比较范围

com.lachesis.datasync.controller.*&com.lachesis.datasync.service.*

历史比对记录

编号	源版本	目标版本	创建时间	操作
1	WRMSSync-v1.2.jar	WRMSSync-v1.1.jar	2023-09-20 14:10:34	<input type="button" value="详情"/> <input type="button" value="删除"/>
2	WRMSSync-v1.1.jar	WRMSSync-v1.0.jar	2023-09-20 14:00:12	<input type="button" value="详情"/> <input type="button" value="删除"/>

比对详情页

比对详情

源版本: WRMSSync-v1.2.jar | 目标版本: WRMSSync-v1.1.jar

比较范围: com.lachesis.datasync.controller.* & com.lachesis.datasync.service.*

开始比对

比对结果

100% 版本比对完成

变更项	新增	修改	删除
类	0	3	0
方法	1	4	0

查看报告

详细日志

```

2023-09-20 15:13:11 源文件: WRMSSync-v1.2.jar
2023-09-20 15:13:11 目标文件: WRMSSync-v1.1.jar
2023-09-20 15:13:11 比较范围: com.lachesis.datasync.controller.* & com.lachesis.datasync.service.*
2023-09-20 15:13:11 扫描源文件class文件完成, 文件总数: 667
2023-09-20 15:13:11 扫描目标文件class文件完成, 文件总数: 667
2023-09-20 15:13:11 比对差异明细
2023-09-20 15:13:11 变更类: data-sync-service/src/main/java/com/lachesis/datasync/service/impl/LxMqService.java
2023-09-20 15:13:11 修改 = sendMq2LxService
2023-09-20 15:13:11 修改 = writeDataToProcedure
2023-09-20 15:13:11 变更类: data-sync-service/src/main/java/com/lachesis/datasync/util/DataParserXmlUtils.java
2023-09-20 15:13:11 修改 = formatMessage
2023-09-20 15:13:11 新增 = isAllReplace
2023-09-20 15:13:11 变更类: data-sync-service/src/main/java/com/lachesis/datasync/util/DataSourceDao.java
2023-09-20 15:13:11 修改 = getJdbcTemplateByConfigure
2023-09-20 15:13:11 版本比较结束,耗时: 34秒
2023-09-20 15:13:11 分析影响用例开始
2023-09-20 15:13:11 ...
2023-09-20 15:13:11 分析影响用例结束
2023-09-20 15:13:11 生成报告开始
2023-09-20 15:13:11 生成报告结束

```

报告详情

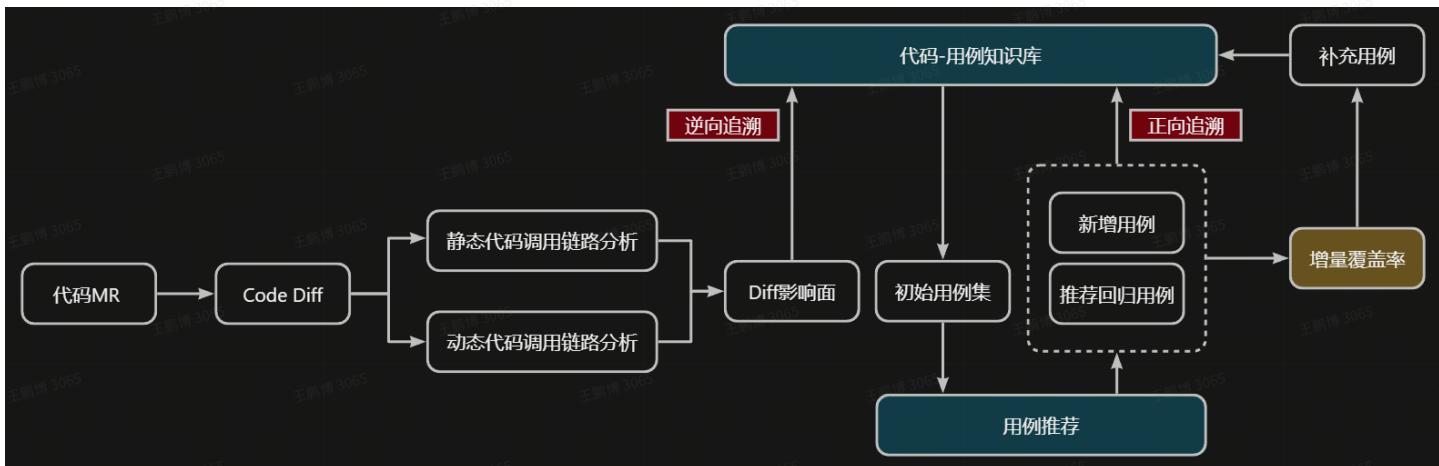
变更项

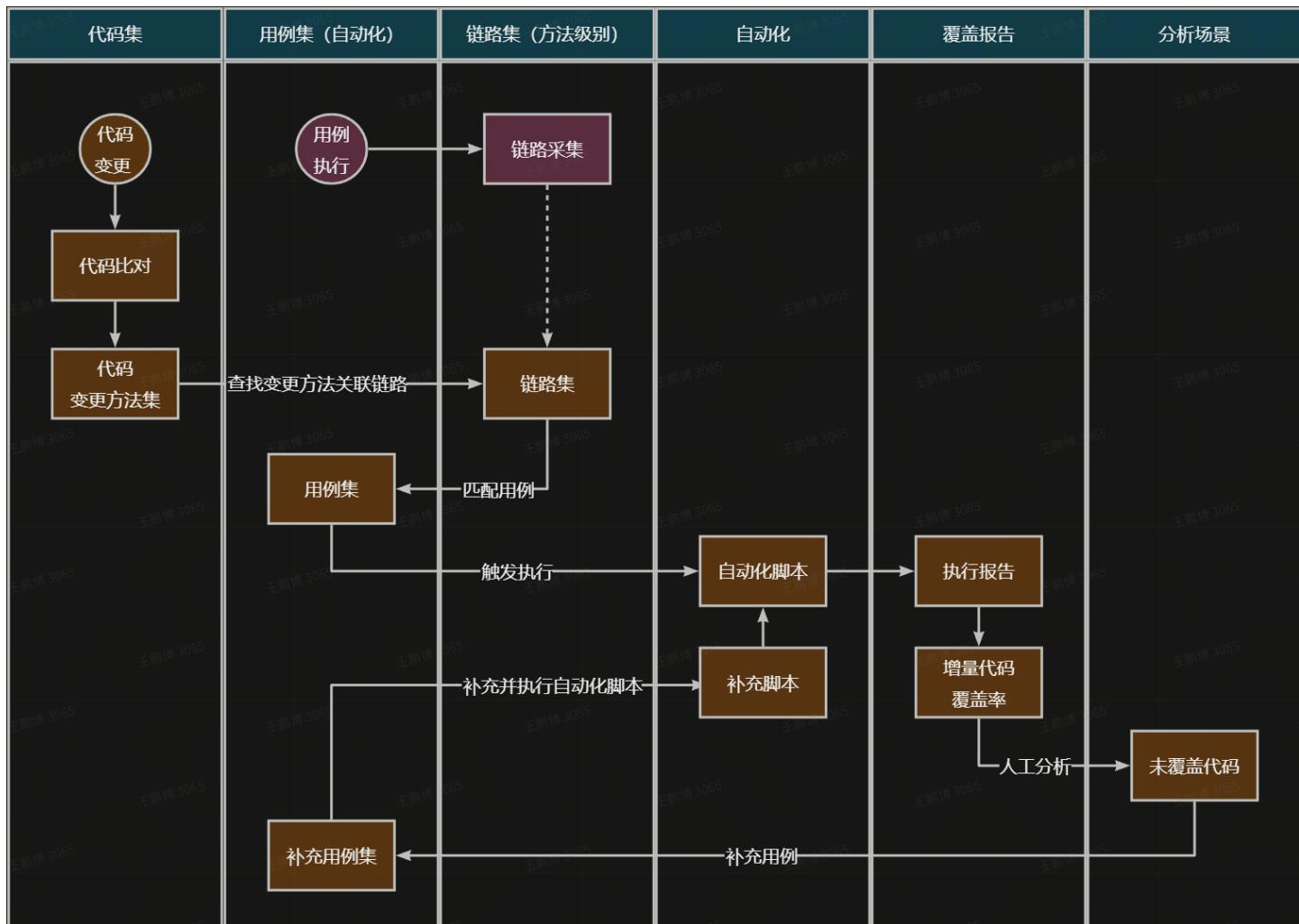
编号	用例名称	用例描述	影响点

影响用例

编号	用例名称	用例描述	影响点

实现方案





服务端

1. 调用链路分析



服务端调用链路分析主要分为三个方面：

- 代码差异比对，通过代码差异比对得到最终的影响面。

- 静态链路分析，通过静态链路分析得到最终的静态调用关系。

- 动态链路分析，通过动态链路分析得到最终的动态调用关系。

为什么有了静态链路分析，还需要动态链路分析呢？各自的优缺点是？

- 静态链路分析，**发生在程序运行之前**，只是基于静态代码分析出来的链路关系，类似开发在使用IDEA工具时，通过快捷键追踪方法的调用链路。

- **优点 代码覆盖**：静态分析可以分析整个代码库，包括很少执行的代码路径，以及特定情况下的代码。

- **缺点 无法捕获运行时信息**：静态分析不能提供有关代码实际运行时行为的信息。

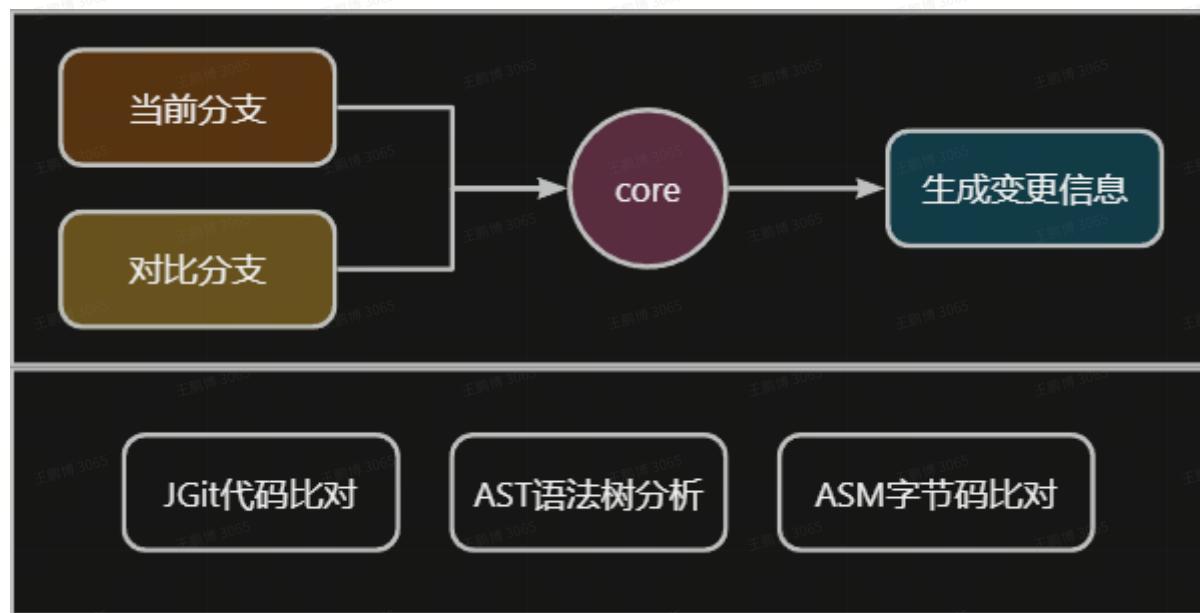
- 动态链路分析：**发生在程序运行之后**，提供了关于程序实际的执行情况，反馈了程序实际真实的运行情况。

- **优点 实际运行时信息**：动态代码分析提供有关代码在实际运行时的行为、性能和资源利用情况的信息。

- **缺点 覆盖范围有限**：动态分析可能无法覆盖所有代码路径，尤其是很少执行的路径。

因此，静态链路分析和动态链路分析是相辅相成，在帮助开发团队更全面地理解、测试和优化软件系统。

代码差异比对



1. 通过JGit Diff计算代码改动；

2. 通过ASM获取方法级别改动信息（当前暂定改动为方法级别）

静态链路分析

TODO

动态链路分析

TODO

2.用例和代码关联

TODO

客户端

SHELF

当前不考虑，需要求助前端外援~

增量覆盖率获取

TODO

总结

精准测试+自动化测试=降本增效

自动化测试：

- 单元测试
- 测试驱动开发

精准测试：

- 调用链路分析
- 用例代码关联
- 用例智能推荐

计划

阶段	内容	排期
第一阶段	需求整理与技术预研	4个工作日（实际投入时间）
第二阶段	Code Diff计算引擎实现	
第三阶段	静态链路分析	
第四阶段	动态链路分析	
第五阶段	用例关联与推荐	

Original Confluence page attachments

Name	Size	Created by	Created on	Labels	Comments
 image-2023-9-21_13-39-... 108.89KB	108.89 KB	王鹏博	2023-09-21T13:33:11.000+08:00		
 image-2023-9-20_15-46-... 245.97KB	245.97 KB	王鹏博	2023-09-20T15:40:44.000+08:00		
 image-2023-9-20_15-20-... 169.75KB	169.75 KB	王鹏博	2023-09-20T15:14:17.000+08:00		
 image-2023-9-20_15-19-... 57.08KB	57.08 KB	王鹏博	2023-09-20T15:13:19.000+08:00		

 image-2023-9-18_18-22-... 17.10KB	17.10 KB	王鹏博	2023-09-18T18:16:34.000+08:00		
 image-2023-9-18_18-21-... 17.35KB	17.35 KB	王鹏博	2023-09-18T18:16:00.000+08:00		
 image-2023-9-7_16-49-... 18.79KB	18.79 KB	王鹏博	2023-09-07T16:44:34.000+08:00		
 image-2023-9-7_16-46-... 13.98KB	13.98 KB	王鹏博	2023-09-07T16:41:54.000+08:00		
 image-2023-9-7_16-43-... 14.79KB	14.79 KB	王鹏博	2023-09-07T16:38:37.000+08:00		
 image-2023-9-7_15-58-... 288.48KB	288.48 KB	王鹏博	2023-09-07T15:53:41.000+08:00		
 image-2023-9-7_15-57-... 323.09KB	323.09 KB	王鹏博	2023-09-07T15:52:50.000+08:00		
 image-2023-9-7_12-48-7.png 59.10KB	59.10 KB	王鹏博	2023-09-07T12:43:09.000+08:00		
 image-2023-9-7_12-44-... 58.38KB	58.38 KB	王鹏博	2023-09-07T12:39:44.000+08:00		
 image-2023-9-7_11-27-... 9.26KB	9.26 KB	王鹏博	2023-09-07T11:22:		

