京东平台化实践-移动端接口测试

张伟@京东

目录

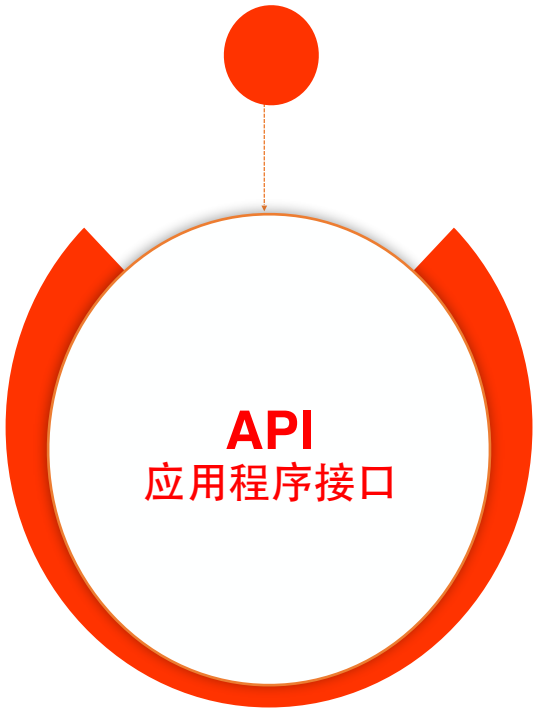
CONTENTS

01

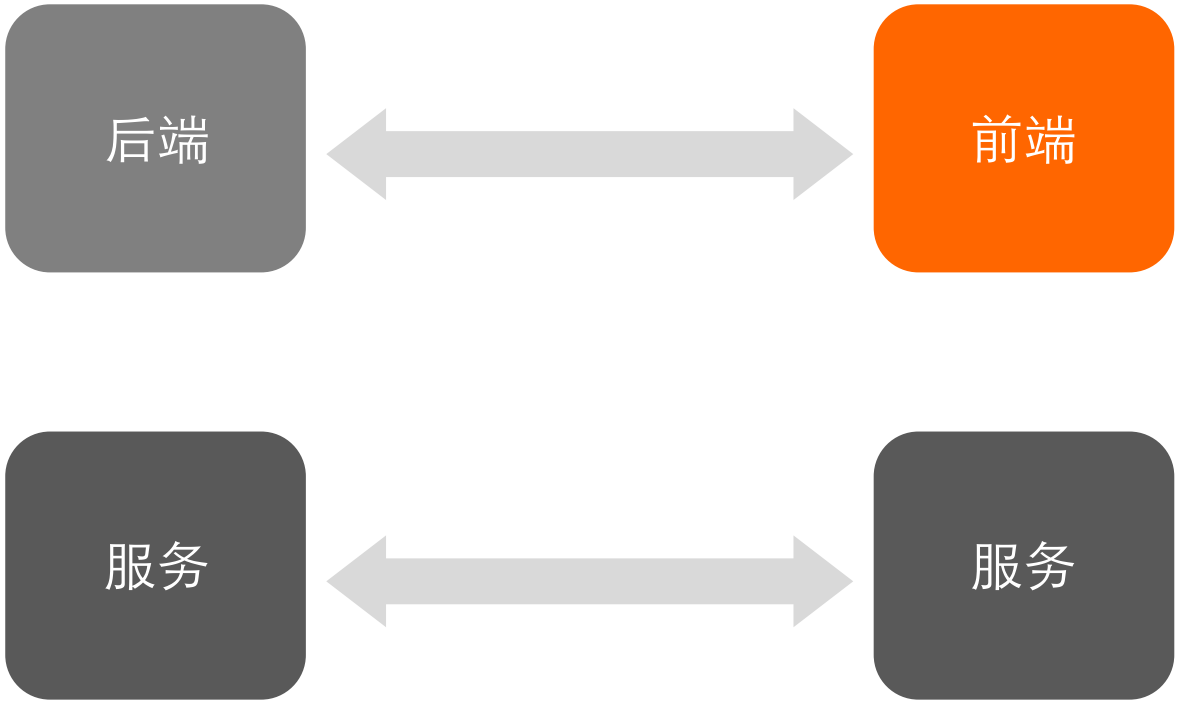
接口及接口测试


崩溃

了解一下接口



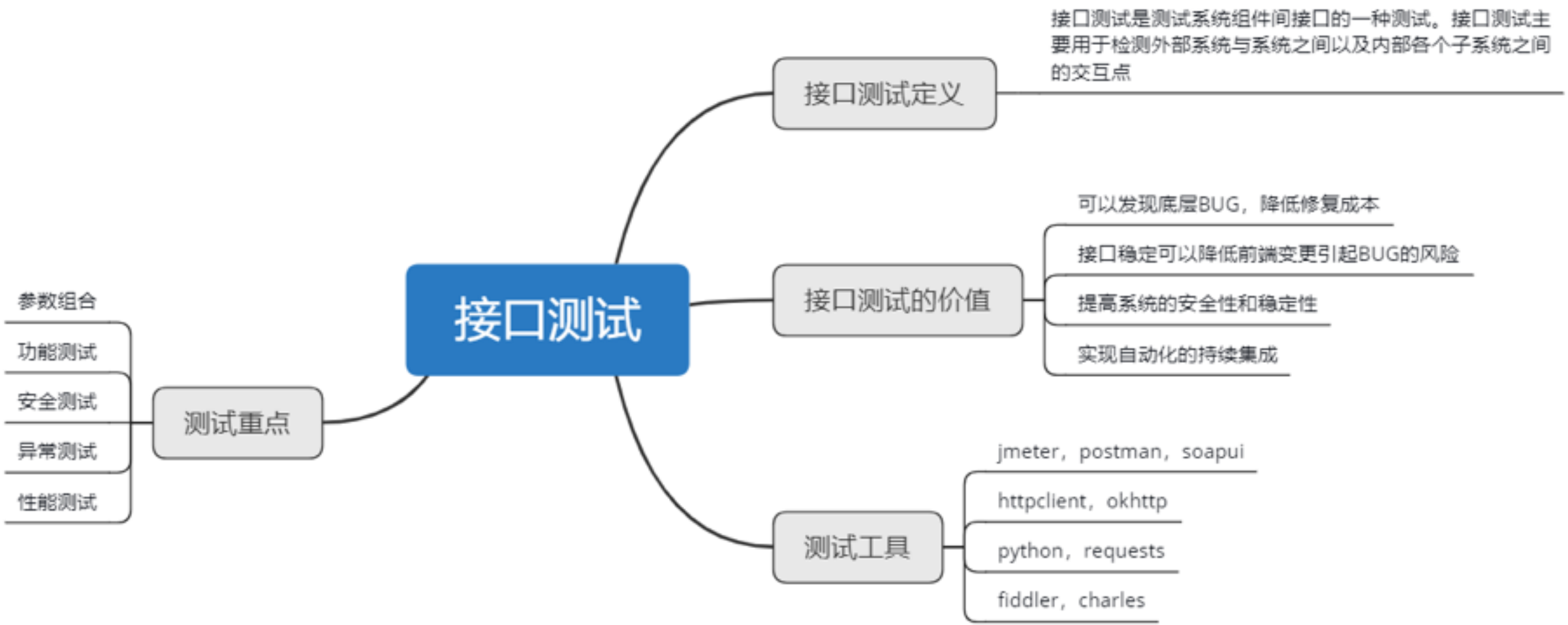
- 由一套陈述、功能、选项、其它表达程序结构的形式、以及程序员使用的程序或者程序语言提供的数据组成的接口



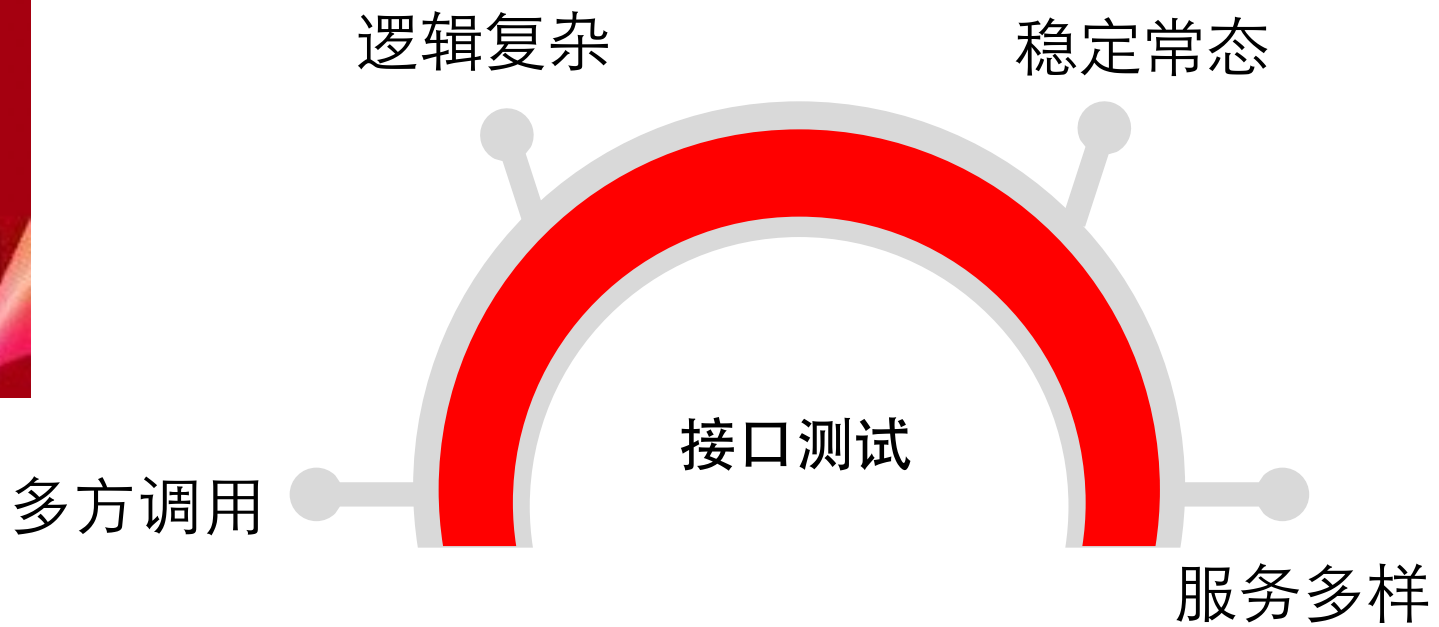
HTTP

JSF杰夫

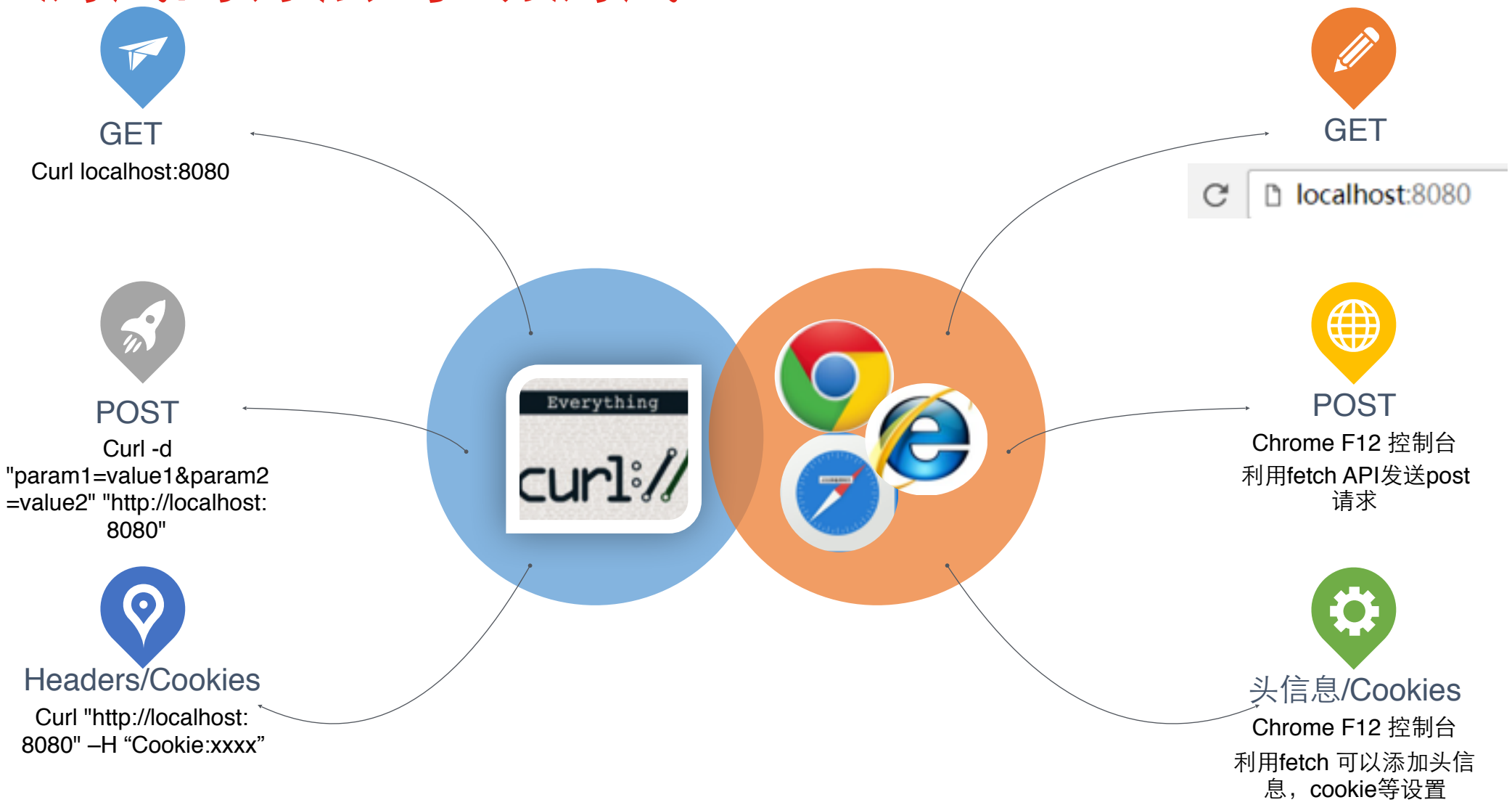
了解一下接口测试



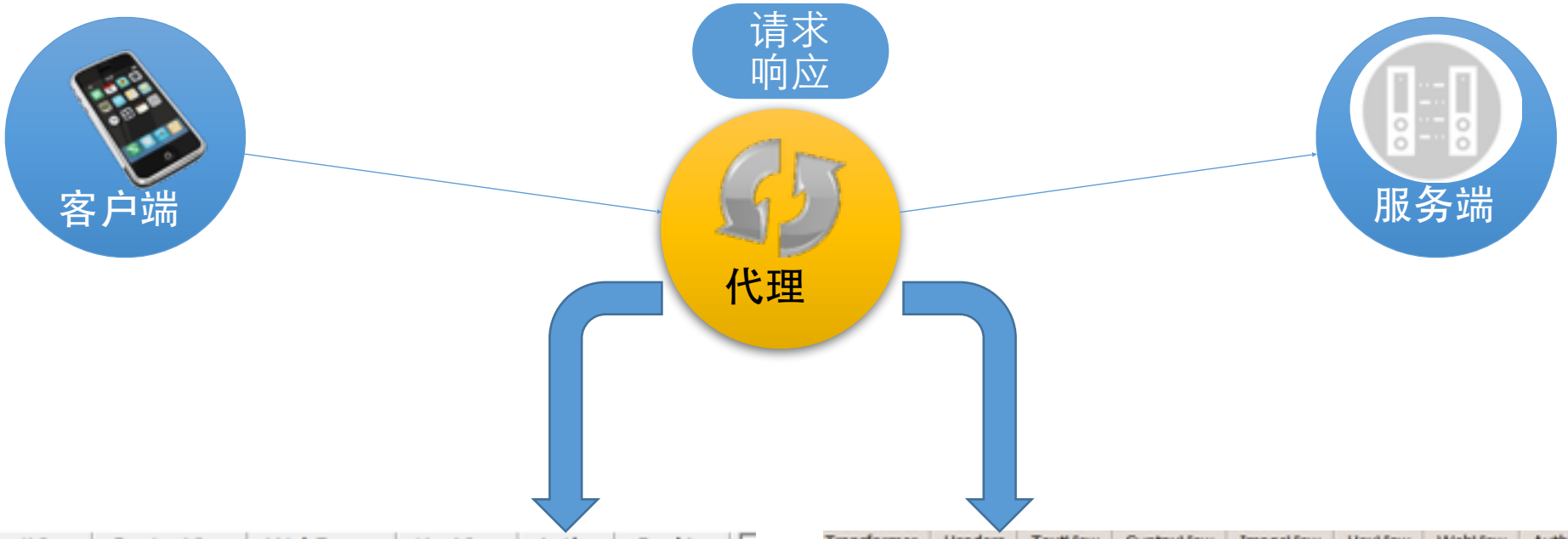
为什么进行接口测试



接口测试的方法-手动测试



接口测试的方法-借助工具



Headers | TextView | SyntaxView | WebForms | HexView | Auth | Cookies

```
GET http://baike.baidu.com/api/openapi/BaikeLemmaCardApi?scope=103&
Host: baike.baidu.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:59.0) Gecko
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3
Accept-Encoding: gzip, deflate
Cookie: BAIDUID=A590CD550980C75967972DE9809F7571:FG=1; BIDUPSID=A04
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

Transformer | Headers | TextView | SyntaxView | ImageView | HexView | WebView | Auth | Caching | Cookies | Raw | JSON

- JSON
  - abstract=关键字又称保留字（keyword）是指在搜索引擎行业中，希望访问者了解的产品、服务或者公司等内容名称的用语。另外，关键
  - card
    - {}
      - format
        - 关键字
      - key=m25_nameC
      - name=中文名
      - value
        - 关键字

02

接口测试脚本化

需要什么样的脚本

监控相关

实时监控生产环境的业务功能

快速响应

满足业务需求

业务相关

不同分类维度的脚本

回归相关

生产环境与测试环境的数据对比

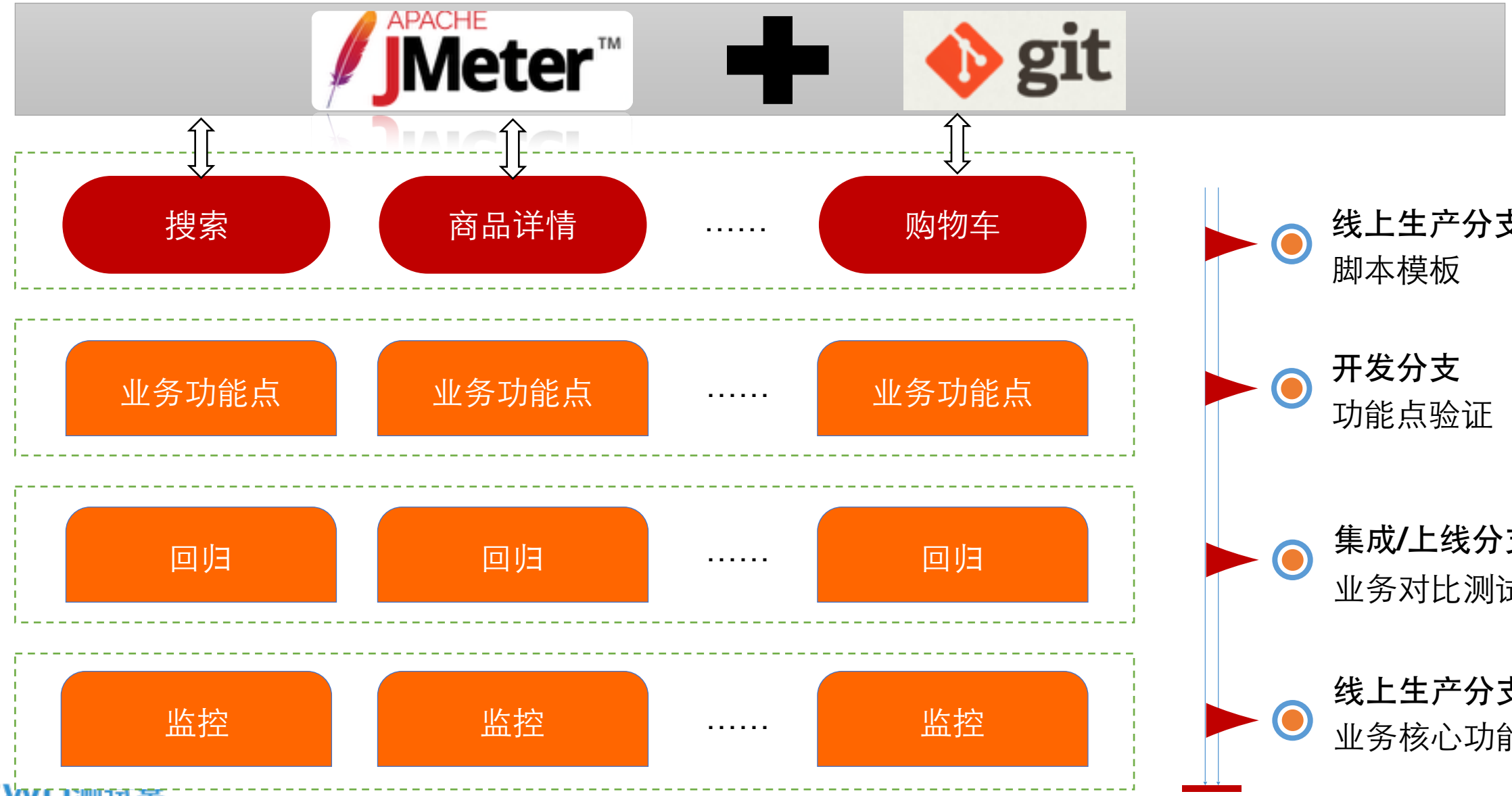
结果准确可靠

高可复用

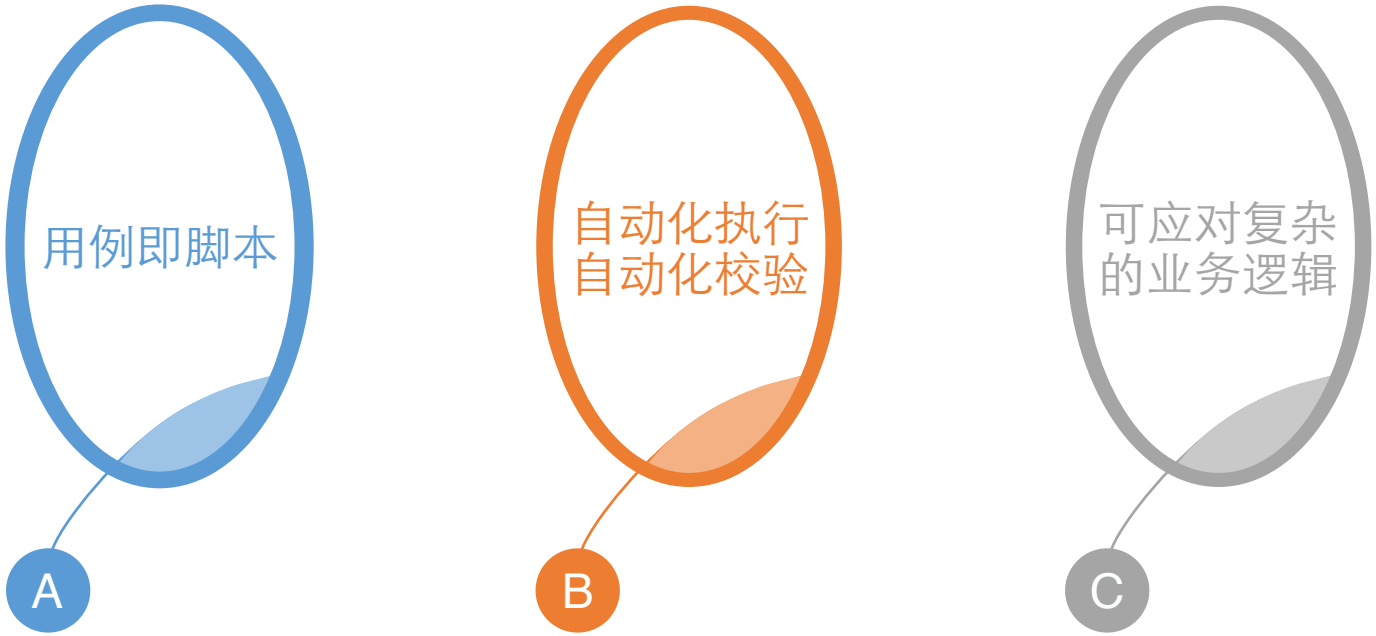
基础模板

遵循什么编写规范?
参数化，统一登录，统一校验

如何组织测试脚本



质量保障



只需要把脚本写好，就能保障业务功能质量

03

接口测试平台化

现有成熟平台

DOClever

SWAGGER

EasyAPI

TestNG

python

Requests

Java

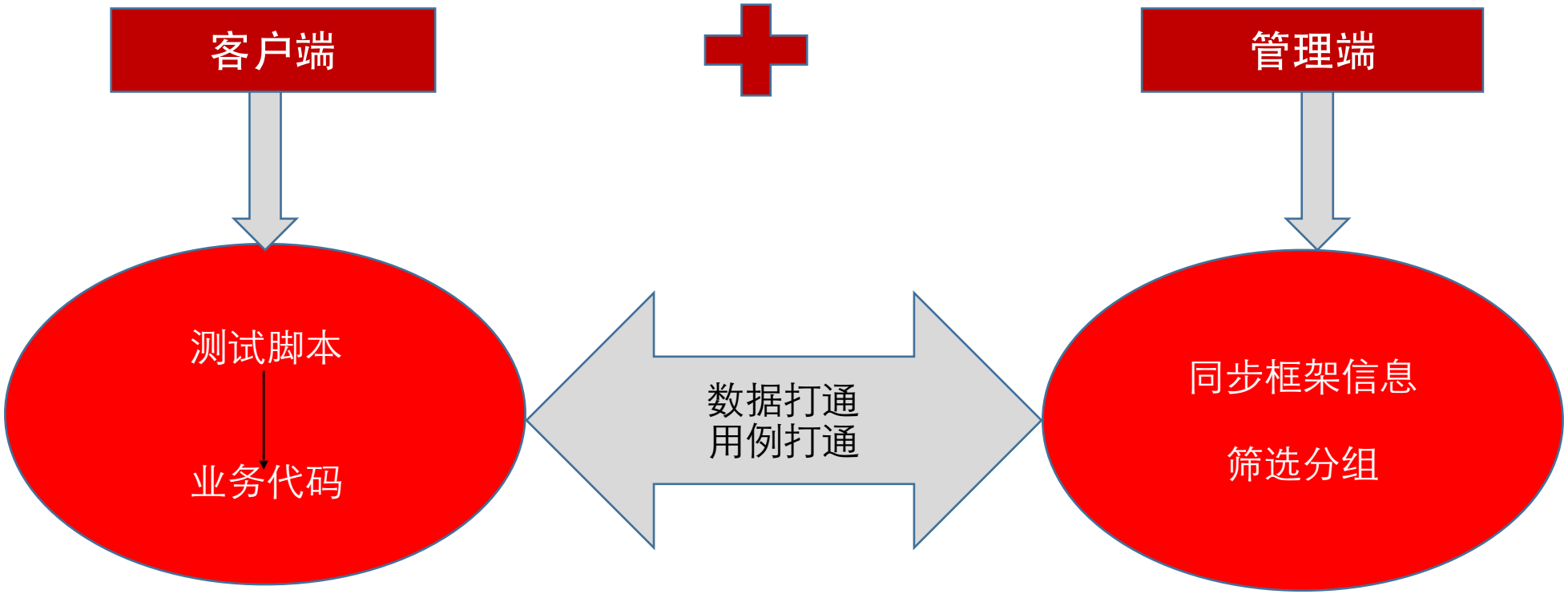
unittest — Unit testing framework

JAKARTA COMMONS HTTPCLIENT

OkHttp

需要什么样的平台

大黄蜂测试平台架构



大黄蜂测试平台技术栈

客户端	数据端	管理端
TestNG	Logstash	Spring Boot
Logback	Redis	Maven
Maven	Mysql	ibatis
OKHttp, JSF, TCP	Zookeeper	iView
-	-	OKHttp, JSF, TCP
-	-	Quartz
Jenkins		

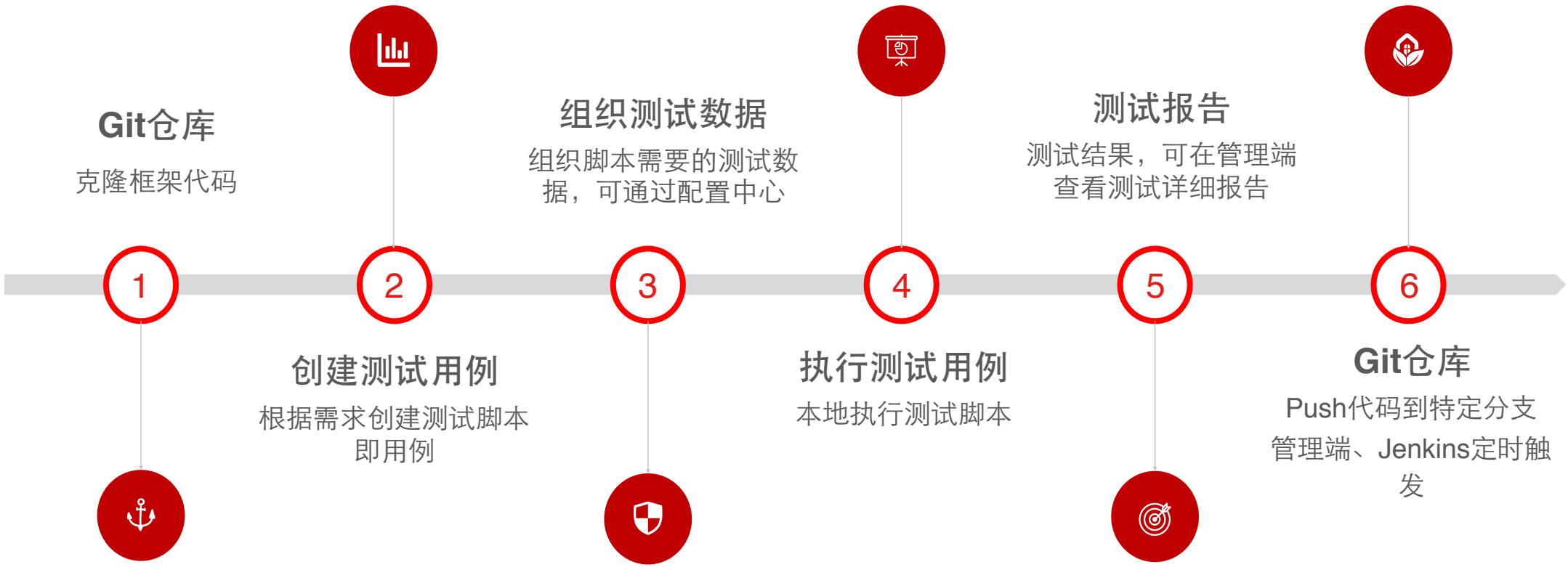
大黄蜂测试平台客户端

```
@Test(dataProvider = "getStarted08", dataProviderClass= HttpGetStartedDat
@Info([redacted], module = "[redacted]", release = "7.0.10", desc
public void getStared08(HttpEntity httpEntityInfo) {
    siteConfig.setResponseType(ResponseType.JSON);
    Response response = Request.call(siteConfig, [redacted]).post();
    Assert.assertTrue(response.string().contains("code"));
}
```

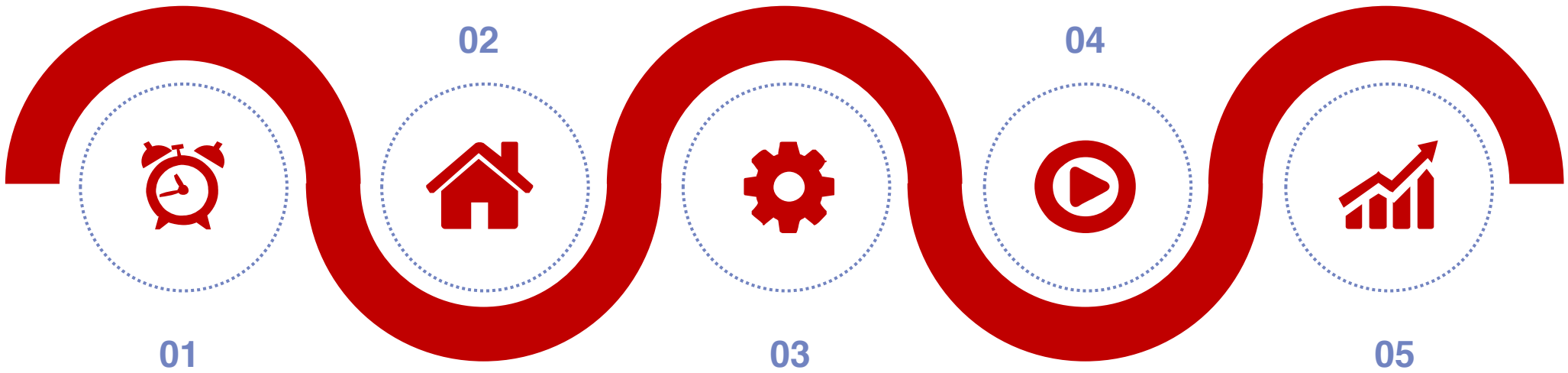
```
@Test(dataProvider = "getStarted01", dataProviderClass= MixGetSt
@Info([redacted], module = "[redacted]", release = "7.0.0
public void getStared01(DataEntity dataEntity) {
```

大黄蜂测试平台管理端

大黄蜂测试平台客户端工作流



大黄蜂测试平台管理端工作流



01

外部触发

基于事件
基于定时

02

用例选择

GIT克隆客户端代码
历史记录过滤

03

数据配置

配置中心
指定数据源

04

在线执行

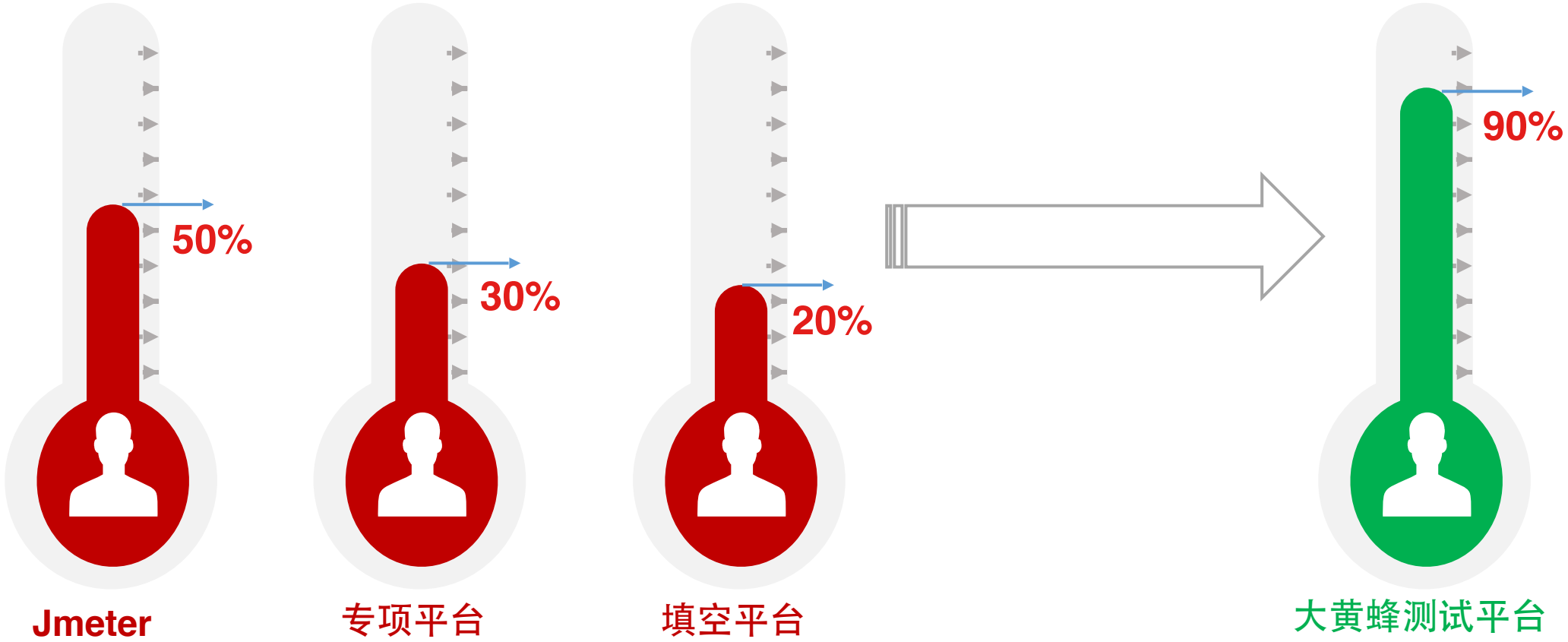
按需执行用例脚本

05

结果输出

测试完成
输出测试报告
邮件周知

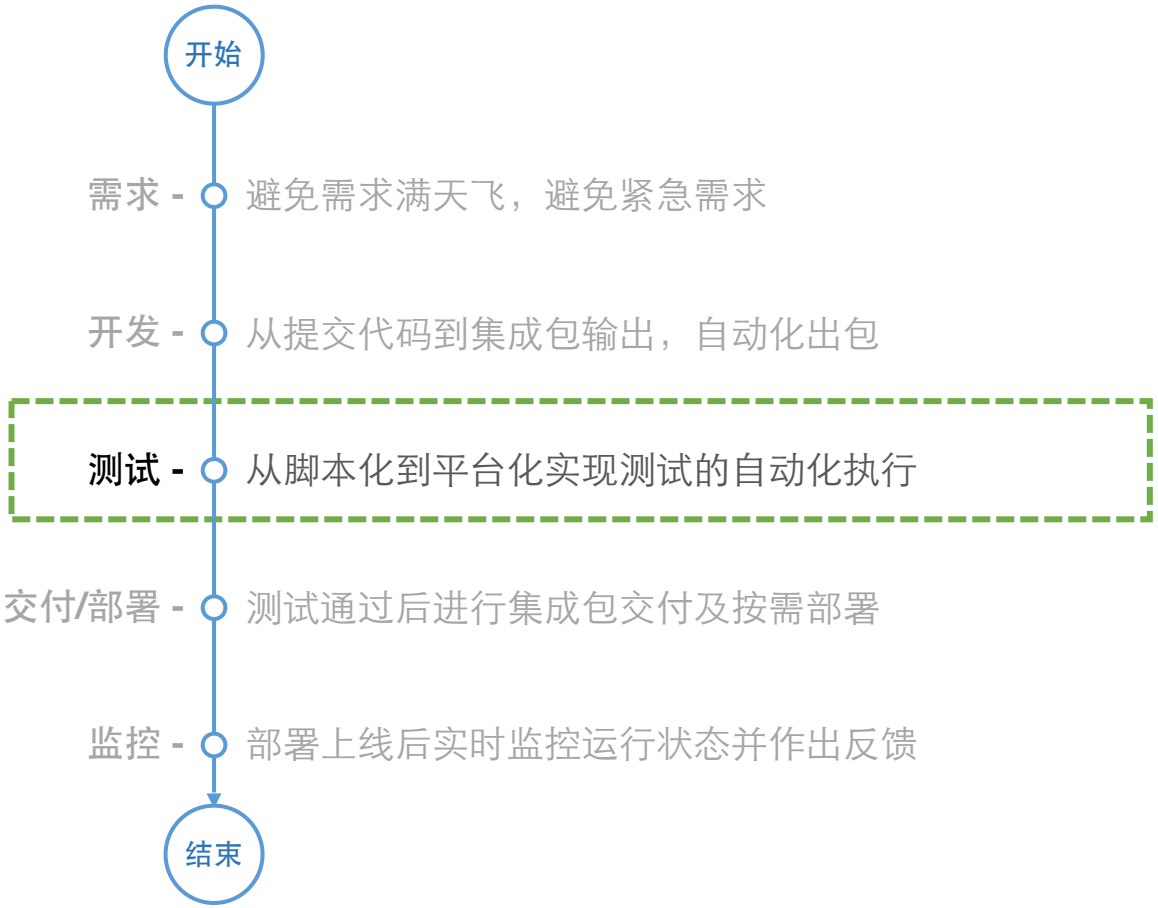
大黄蜂测试平台运营效果



04

接口测试持续化

接口测试如何持续化



通过平台化完成了测试执行的自动化

交付流程的自动化才是我们的目的

接口测试Jenkins编排

构建

测试

交付

代码拉取，平台编译打包，单元测试

代码拉取，平台编译打包，单元测试，静态代码扫描，测试环境配置，自动化测试，覆盖率评估

生产环境配置，自动化测试，覆盖率评估，性能评估，交付war包

生产环境自动配置，按需进行自动化部署，按需灰度发布

开发分支

自动合并

集成分支

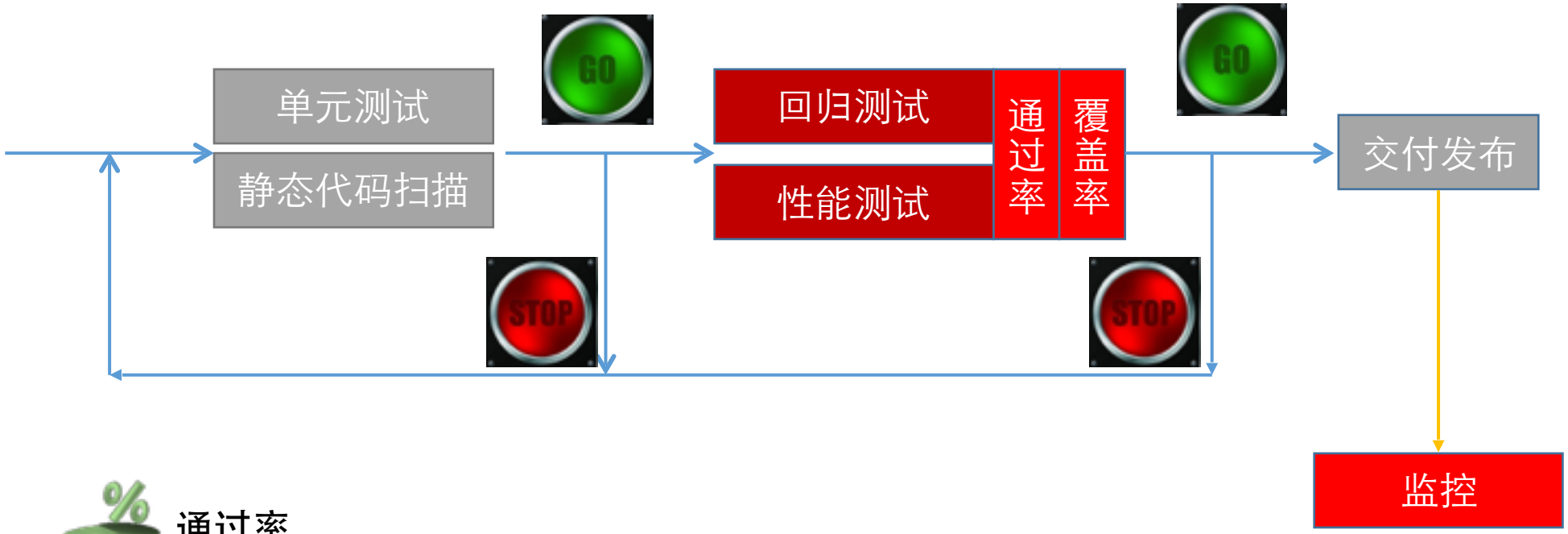
自动合并

交付分支

自动合并

主干

持续化测试工作流



通过率

测试结束会生成测试通过率报告，通常通过率不能低于99%

覆盖率

回归测试跑完，对应的代码覆盖率输出，通常覆盖率会是一个趋势指标，不能低于上一个版本

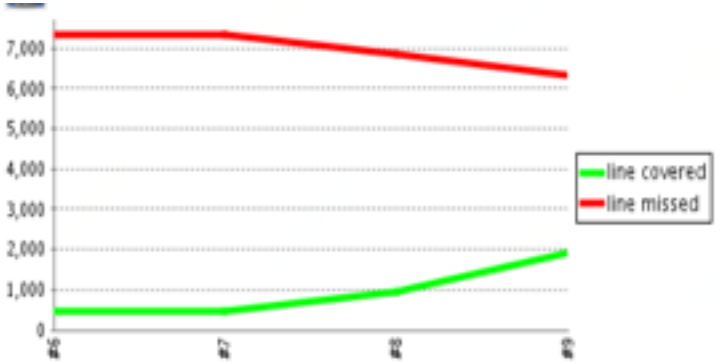
如何整合覆盖率

整合方法

使用Jacoco
开源的覆盖率工具
使用JavaAgent技术监控Java程序

Jacoco包含了多种尺度的覆盖率计数器

- 指令级(Instructions,C0coverage)
- 分支（Branches,C1coverage）
- 圈复杂度(CyclomaticComplexity)
- 行(Lines)
- 方法(non-abstract methods)
- 类(classes)



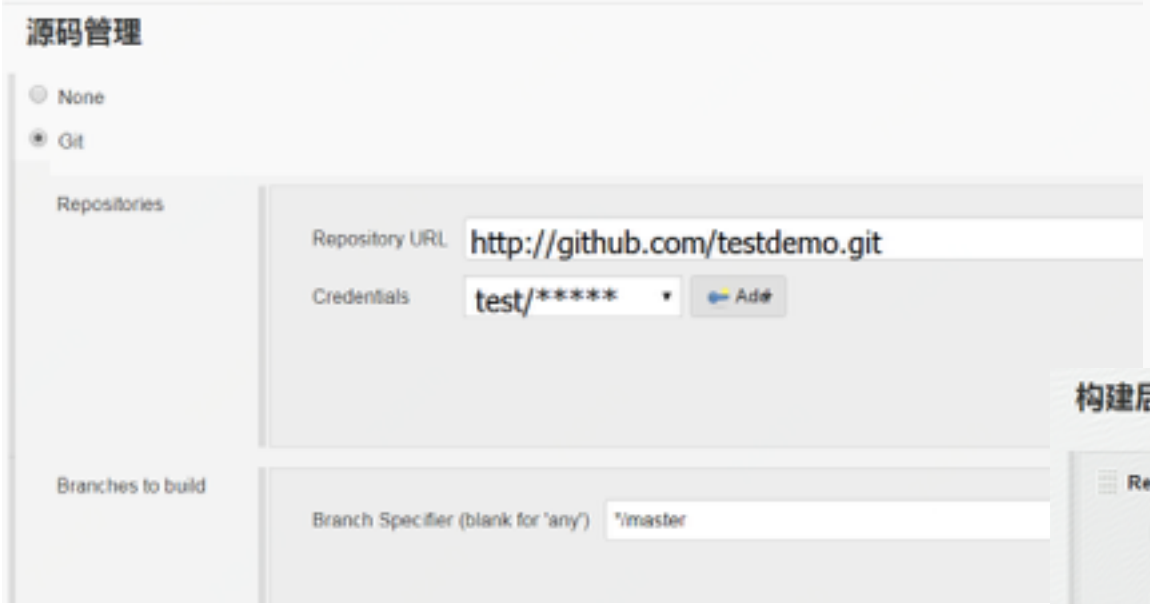
Overall Coverage Summary

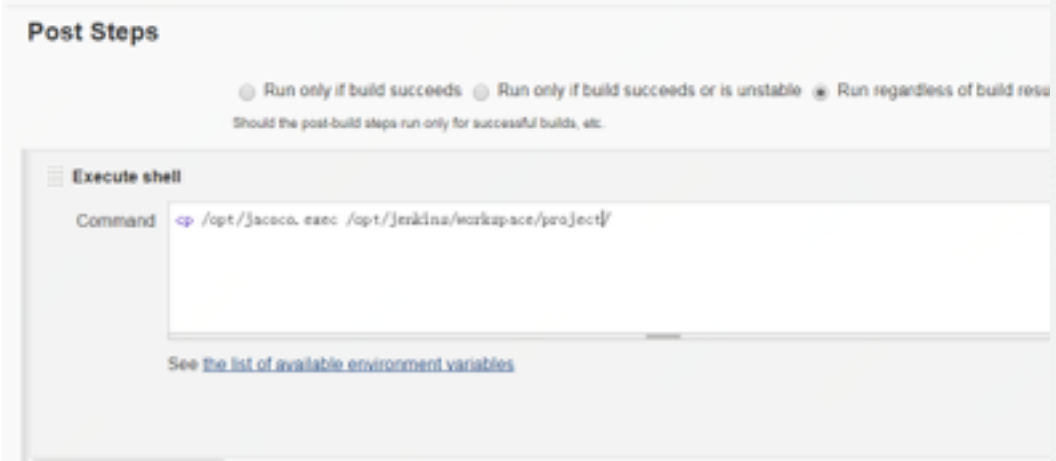
name	instruction	branch	complexity	line	method	class
all classes	21% M: 31883 C: 8627	15% M: 2048 C: 356	27% M: 1931 C: 723	23% M: 6339 C: 1931	44% M: 810 C: 642	71% M: 38 C: 92

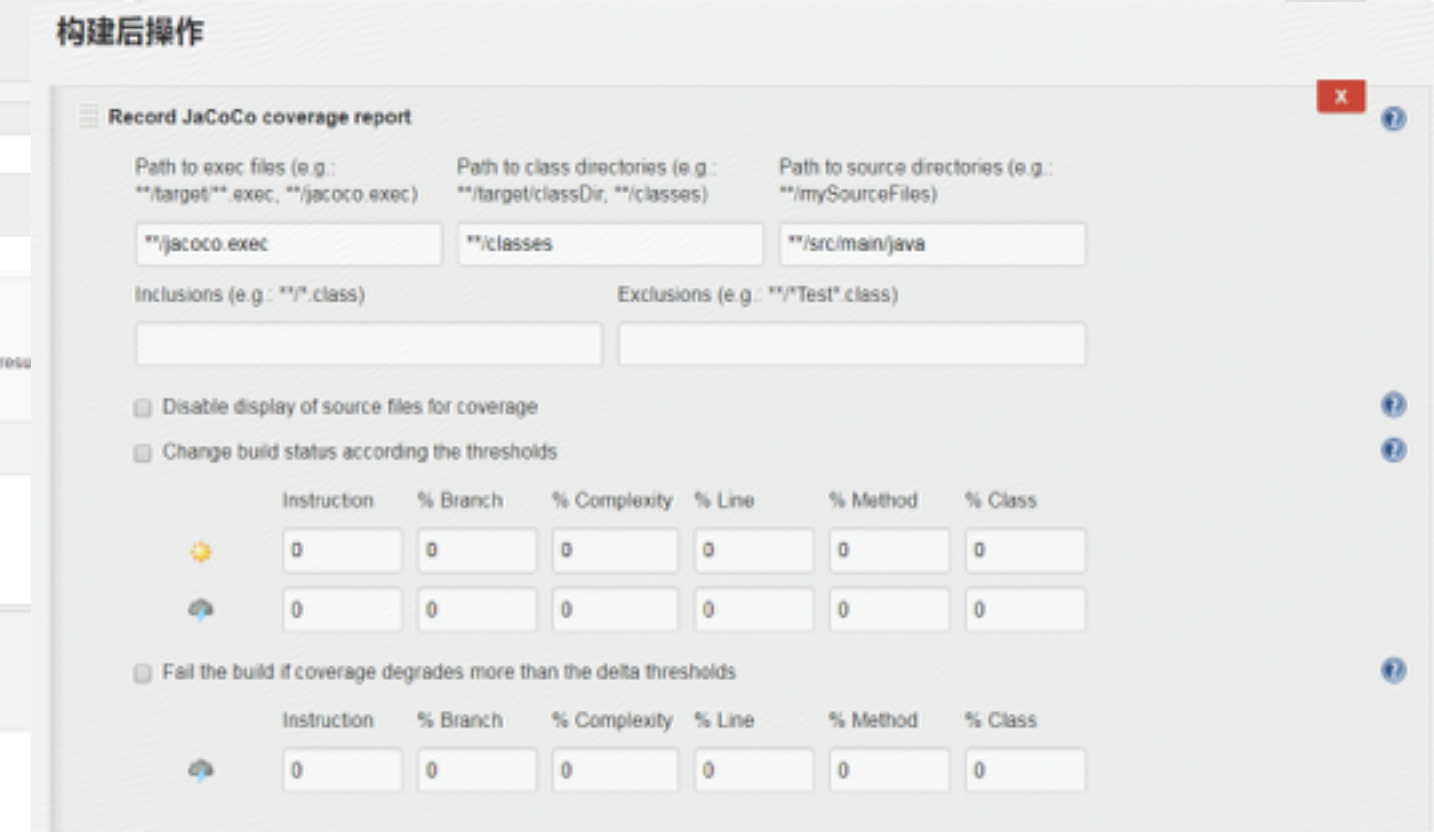
Jacoco - Overall Coverage Summary

INSTRUCTION	21%
BRANCH	15%
COMPLEXITY	27%
LINE	23%
METHOD	44%
CLASS	71%

覆盖率Jenkins配置







接口测试展望

- 大黄蜂测试平台每天会产生大量的测试数据，测试记录，如何使用?
- 如何根据业务需求适配测试用例及数据来源，从而提升自动化测试的成功率?

改变的同时也伴随着问题的产生

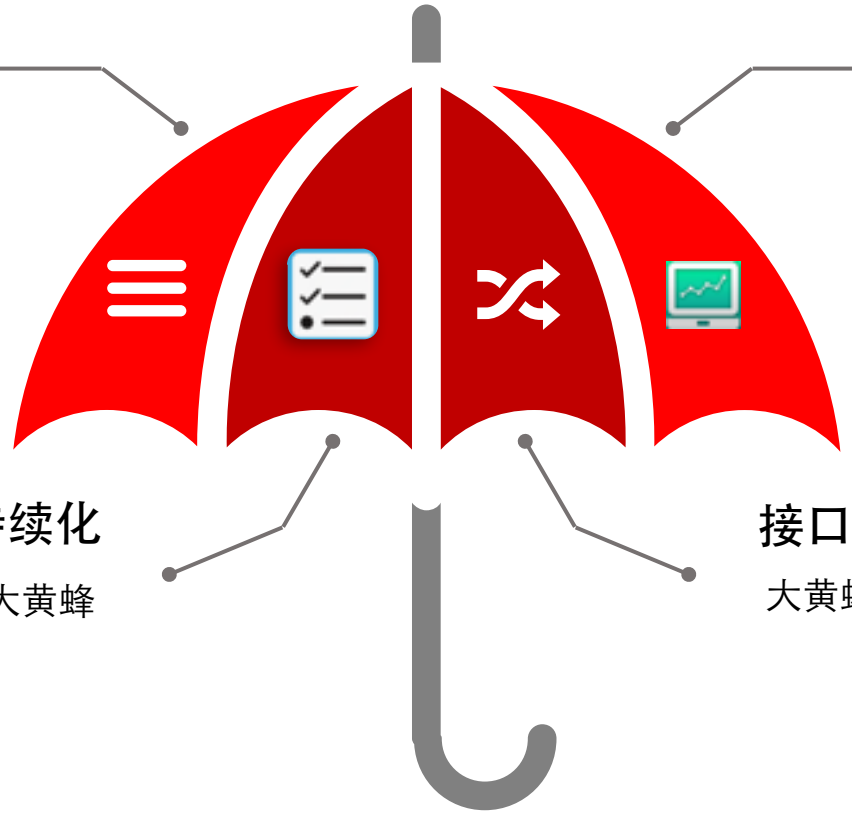
解决问题也是在驱动改变的发生

接口及接口测试

接口及测试的定义
接口测试的方法

接口测试脚本化

脚本类型
Jmeter脚本组织



接口测试持续化

Jenkins+大黄蜂

接口测试平台化

大黄蜂测试平台

质量保障 持续改进

感谢您的参与！

关注公众号获得更多软件测试知识号