DevOpsDays

2017 DevOpsDays Beijing

时间：2017.03.18 地点：北京朝阳区万达索菲特大酒店

主办单位：高效运维社区 GreatOPS Community　Best Practice 最佳实践

手机京东: 持续集成中的自动化实践

方成

自我介绍

移动端测试开发： fangcheng@jd.com

简单介绍：
9年的开发，测试经验，专注于测试的各个方面

目录

为什么做持续集成，痛点在哪里？

面临的问题

开发

1. 手动打包，费时费力易出错；
2. 各个开发手动出包环境不统一，出包内容有差别
3. Build记录无法追溯，问题定位困难

为什么做持续集成，痛点在哪里？

面临的问题

测试

1. 我可能用了个假的测试包
2. 为什么只有我的测试有问题
3. 谁有xxx tag的包，我想验证一个问题
4. 这些工具用起来好麻烦

为什么做持续集成，痛点在哪里？

面临的问题

QA

1. 灰度期间这些问题都是什么类型
2. 集成测试阶段与灰度阶段的测试数据对比
3. 现在的流程真的完美无缺，无法改进么？

为什么做持续集成，总结一下

- 提升开发效率节省人力资源
- 统一了测试入口标准并规范化
- 降低工具上手难度
- 增加测试维度覆盖
- 使测试介入开发流程阶段
- 提前降低BUG修复成本
- 自动化部署，流程更透明

手机京东的自动化实践，为了自动化而自动化？

手机京东的自动化实践，为了自动化而自动化？

单元测试

Android	Robolectric/Junit4 + Mockito + PowerMock
iOS	XCTest
SOA	Junit4 + Mockito + PowerMock

手机京东的自动化实践，为了自动化而自动化？

单元测试

Android 单元测试案例：

Junit4 + Mockito + PowerMock

1. 痛点：JVM上运行纯JUnit单元测试时没有 android context，里面只定义了一些接口,所有方法的实现都是 throw new runtimeException(“stub”)，因此需要使用Android提供的Instrumentation系统，将单元测试代码运行到模拟器或真机上，执行效率有所折扣

2. 为什么使用PowerMock：Mockito是通过创建proxy的方式来实现mock，因此对于static,final,private方法都是不能mock的，而PowerMock是通过使用CGLib来操纵字节码的方式实现mock，所以可以弥补Mockito的不足

手机京东的自动化实践，为了自动化而自动化？

单元测试

Demo Script

```
public class LoginPresenterTest {
    @Test
    public void testLogin() throws Exception {
        UserManager mockUserManager = Mockito.mock(UserManager.class);
        /**
         * 创建类的Mock对象
         */
        LoginPresenter loginPresenter = new LoginPresenter();
        loginPresenter.setUserManager(mockUserManager);
        loginPresenter.login( username: "test",  password: "password");
        /**
         * 验证某一对象某个方法的调用情况
         * 注意点：需要将mock获得的对象应用到正式代码中。setter或者依赖注入
         */
        Mockito.verify(mockUserManager).performLogin( userName: "test",  passWord: "password");
        /**
         * 指定mock对象的某些方法的行为
         * 如：
         * 指定方法返回特定值
         * 指定方法执行特定动作
         */
        /**
         * Mockito.when(mockObject.targetMethod(args)).thenReturn(desiredReturnValue);
         */
        //当调用mockValidator的verifyPassword方法时，返回true，无论参数是什么
        /**
         * Mockito.when(validator.verifyPassword(anyString())).thenReturn(true);
         * 说明：当调用mockUserManager的performLogin方法时，会执行answer里面的代码
         我们上面的例子是直接调用传入的callback的onFailure方法，同时传给onFailure
         方法500和"Server error"。
         Mockito.doNothing()
         Mockito.doThrow(desiredException)
         Mockito.doCallRealMethod()//调用真实的逻辑
         */
    }
```

手机京东的自动化实践，为了自动化而自动化？

单元测试

iOS 单元测试：

XCTest：

1. XCode自带单测工具， 在XCode5时引入，XCode6时增加了对性能测试，异步测试的支持
2. 每个配置的Unit Test Class均是继承于XCTestCase

最重要的三个方法：

～～～～～～～～～～～～～～～～～～～～～～～～

(void)setUp:准备测试环境；

(void)tearDown:脚本结束后执行，清理测试影响；

measureBlock: 性能测试方法，用来计算测试block中方法执行时间，通过与期望性能值对比可以判断是否能够通过性能测试

手机京东的自动化实践，为了自动化而自动化？

单元测试

Demo Script

```
@interface TestWJLoginFileHelper : WJLoginIOSClientUnitTesting

@end

@implementation TestWJLoginFileHelper

-(NSString *)path
{
    return [[NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask, YES) firstObject]
        stringByAppendingPathComponent:@"JDLoginInfo.plist"];
}

/**
 创建文件测试
 */
-(void)testCreateFile
{
    NSError *error;
    [[NSFileManager defaultManager] removeItemAtPath:[self path] error:&error];
    if (![[NSFileManager defaultManager] fileExistsAtPath:[self path]]) {
        bool create = [WJLoginFileHelper createLoginFile];
        XCTAssertTrue(create,@"createLoginFile error");
        bool exist = [WJLoginFileHelper isLoginFileExisted];
        XCTAssertTrue(exist,@"isLoginFileExisted error");
    }
}
```

手机京东的自动化实践，为了自动化而自动化？

代码静态扫描

Infer / Findbugs

Infer / Xcode Analyze

Infer

Facebook开源的一款代码静态分析工具

Java/Objective-C中均捕捉的BUG类型如下：

1. 资源泄露问题(Resource Leak)
2. 空指针/引用问题(Null Dereference)

Objective-C中捕捉的BUG类型如下：

1. 内存泄露问题(Memory Leak)
2. 内存死锁问题(Retain cycle)
3. 参数非空检查(Parameter Not Null Checked)
4. 本地变量非空检查(Ivar Not Null Checked)

手机京东的自动化实践，为了自动化而自动化？

代码静态扫描

Infer Tips:

1. 初次运行时，确保项目是清理过的。可以通过 (make clean，gradle clean etc)

2. 两次运行之间，需清理项目，否则--incremental选项会因为增量编译而无结果输出，如使用的是非增量编译方式，则无需如此，比如：infer -- javac Hello.java，编译 Java 文件。

3. 成功运行之后，在同一目录下，你可以通过inferTraceBugs命令浏览更加详细的报告

手机京东的自动化实践，为了自动化而自动化？

代码静态扫描

Infer Report:

```
Found 228 issues

****/***Layout.java:658: error: NULL_DEREFERENCE
  object child last assigned on line 657 could be null and is dereferenced at line 658
  656.                    for (int i = viewIndex; i < viewIndex + flexLine.mItemCount; i++) {
  657.                        View child = getReorderedChildAt(i);
  658. >                      LayoutParams lp = (LayoutParams) child.getLayoutParams();
  659.                        if (mFlexWrap != FLEX_WRAP_WRAP_REVERSE) {
  660.                            int marginTop = flexLine.mMaxBaseline - child.getBaseline();

****/***Util.java:71: error: CONTEXT_LEAK
  Context class com.jingdong.xxx.xxxActivity may leak during method com.jingdong.***.xxxActivity:
  Static field com.jingdong.***.instance |->
  com.jingdong.****.activity |->
  Leaked class com.jingdong.***Activity
  69.
  70.       public BaseActivity getCurrentMyActivity() {
  71. >         return (BaseActivity) BaseFrameUtil.getInstance().getCurrentMyActivity();
  72.       }
  73.   }
```

手机京东的自动化实践，为了自动化而自动化？

代码静态扫描

FindBugs

检查class文件，将字节码与自带或用户自定义的bug pattern进行对比以发现可能的问题, problem example below:

UR_UNINIT_READ_CALLED_FROM_SUPER_CONSTRUCTOR

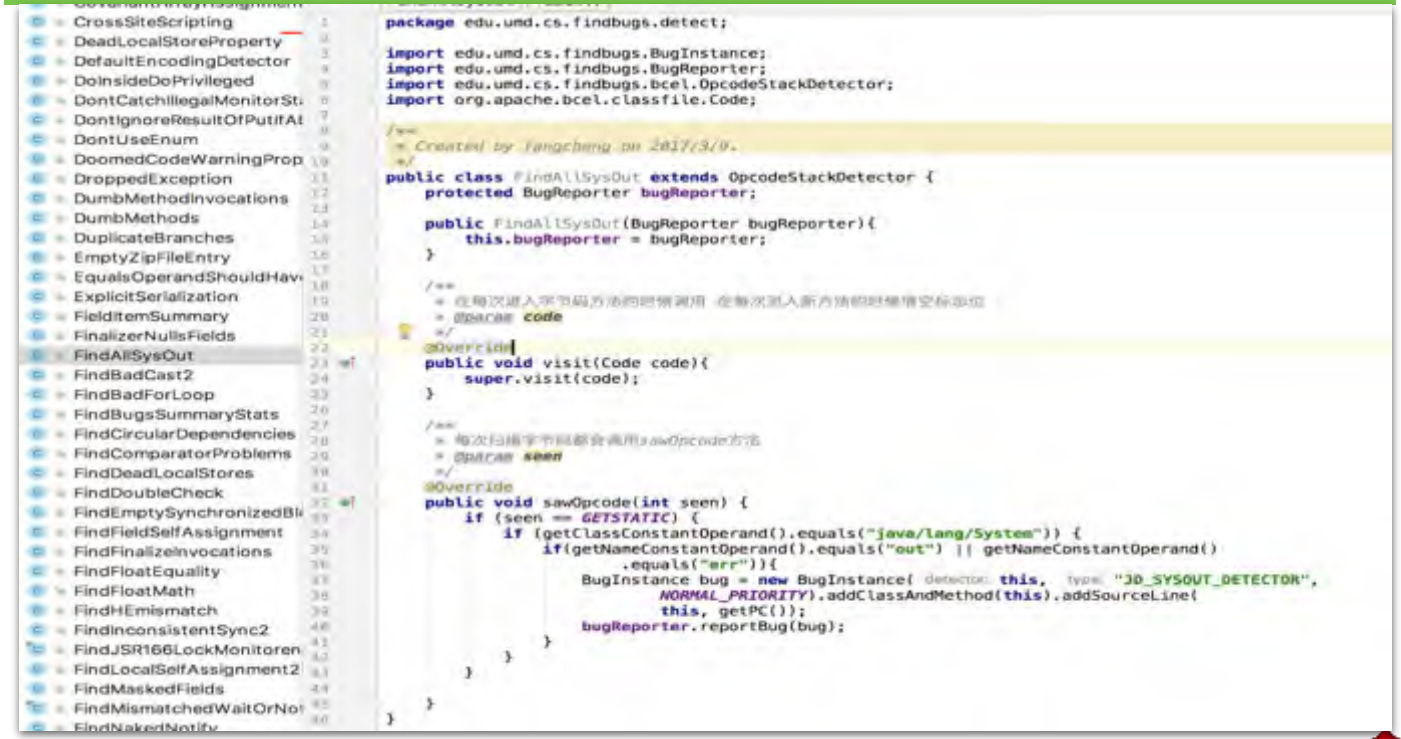
This method is invoked in the constructor of of the superclass. At this point, the fields of the class have not yet initialized. To make this more concrete, consider the following classes:

```
abstract class A {
  int hashCode;
  abstract Object getValue();
  A() {
    hashCode = getValue().hashCode();
    }
  }
class B extends A {
  Object value;
  B(Object v) {
    this.value = v;
    }
  Object getValue() {
    return value;
  }
  }
```

手机京东的自动化实践，为了自动化而自动化？

代码静态扫描

自定义扫描规则：



手机京东的自动化实践，为了自动化而自动化？

接口自动化

SwaggerUI

JMeter

手机京东的自动化实践，为了自动化而自动化？

接口自动化

SwaggerUI + Spring-boot 接口测试管理：

1. 添加依赖到gradle脚本
2. 添加SwaggerUI配置来激活对应服务

```
dependencies {
    compile('io.springfox:springfox-swagger2:2.6.1')
    compile('io.springfox:springfox-swagger-ui:2.6.1')
}
```

```
/**
 * Created by fangcheng on 2017/2/8.
 */
@Configuration
@EnableSwagger2
public class Swagger2Configuration {

    @Bean
    public Docket buildDocket(){
        return new Docket(DocumentationType.SWAGGER_2)
                .apiInfo(buildApiInfo())
                .select()
                .apis(RequestHandlerSelectors.basePackage( basePackage: "com.jd.web.swagger_controller"))
                .paths(PathSelectors.any())
                .build();
    }

    private ApiInfo buildApiInfo(){
        return new ApiInfoBuilder()
                .title("JD接口测试API文档")
                .contact(new Contact( name: "JD Interface Testing API",  url: "xxxx.jd.com", email: "fangcheng@jd.com"))
                .version("V0.1")
                .build();
    }
```

手机京东的自动化实践，为了自动化而自动化？

接口自动化

SwaggerUI + Spring-boot 接口测试管理：
Demo Api

```
@Api(value = "计算器服务",description = "提供简单的计算服务，纯demo")
@RestController
@RequestMapping("/compute")
public class CalculatorController {

    @ApiOperation("加法运算")
    @PostMapping("/add")
    public Double add(@RequestParam Double a, @RequestParam Double b){
        return a+b;
    }

    @ApiOperation("减法运算")
    @PostMapping("/sub")
    public Double sub(@RequestParam Double a,  @RequestParam Double b) { return a - b; }

    @ApiOperation("乘法运算")
    @PostMapping("/mul")
    public Double mul(@RequestParam Double a,  @RequestParam Double b) { return a * b; }

    @ApiOperation("除法运算")
    @PostMapping("/div")
    public Double div(@ApiParam("被除数")@RequestParam Double a, @ApiParam("除数")@RequestParam Double b) { return a / b; }
```

手机京东的自动化实践，为了自动化而自动化？

接口自动化

SwaggerUI + Spring-boot 接口测试管理：

JD接口测试API文档

Created by JD Interface Testing API
See more at itest.jd.com
Contact the developer

calculator-controller : 提供简单的计算服务，纯demo　Show/Hide | List Operations | Expand Operations

POST /compute/add　加法运算

Response Class (Status 200)
double

Response Content Type */*

Parameters

Parameter	Value	Description	Parameter Type	Data Type
a	(required)	a	query	double
b	(required)	b	query	double

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

手机京东的自动化实践，为了自动化而自动化？

接口自动化

JMeter

基本介绍：
Jmeter是一款非常优秀开源的接口/性能测试工具

常用控件：
测试计划、线程组、控制器、配置元件、定时器、前置处理器、后置处理器、断言、监听器

基本的测试流程如下：

Test Plan → Create Thread Group → Develop Test Scripts → Execute Test Plan → Analyze Test Result

手机京东的自动化实践，为了自动化而自动化？

接口自动化

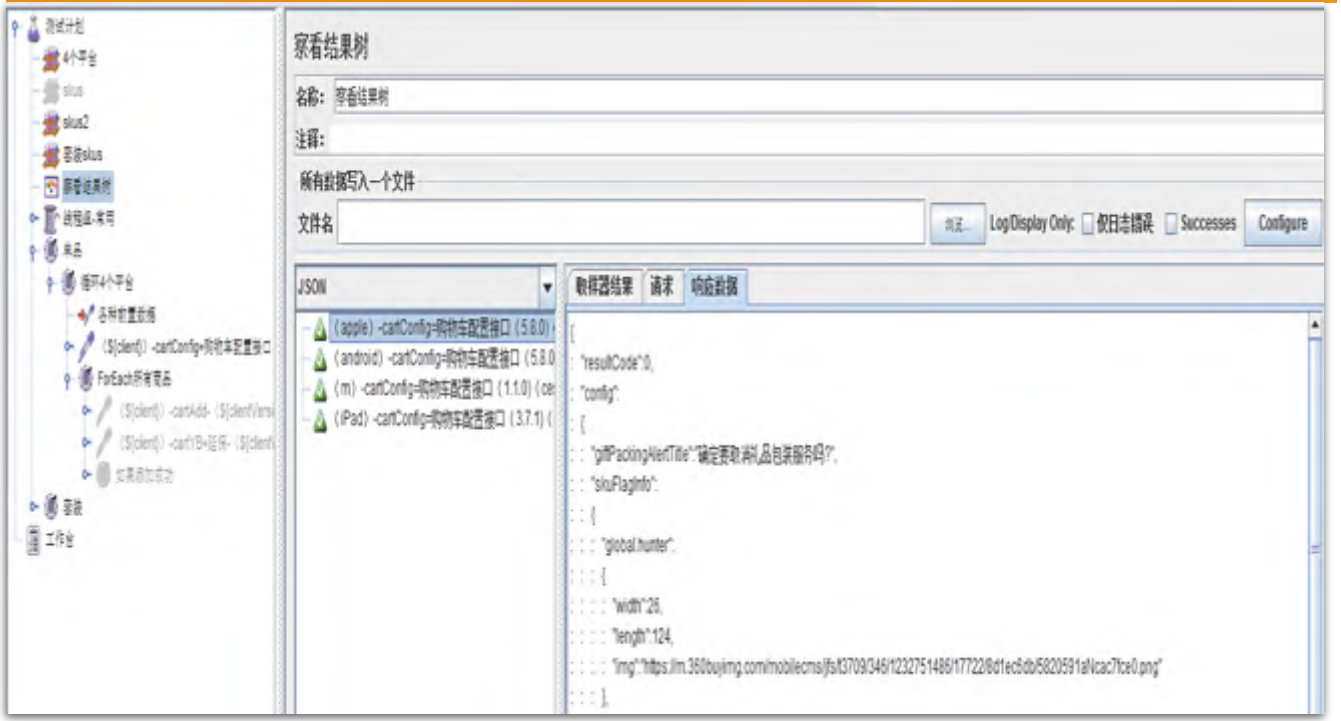
Jmeter的使用：

1. 测试计划是使用 JMeter 进行测试的起点，它是其它 JMeter 测试元件的容器。
2. 线程组代表一定数量的并发用户，它可以用来模拟并发用户发送请求。实际的请求内容在Sampler中定义，它被线程组包含。
3. 监听器负责收集测试结果，同时也被告知了结果显示的方式。
4. 逻辑控制器可以自定义JMeter发送请求的行为逻辑，它与Sampler结合使用可以模拟复杂的请求序列。
5. 断言可以用来判断请求响应的结果是否如用户所期望的。它可以用来隔离问题域，即在确保功能正确的前提下执行压力测试。
6. 配置元件维护Sampler需要的配置信息，并根据实际的需要会修改请求的内容。
7. 前置处理器和后置处理器负责在生成请求之前和之后完成工作。前置处理器常常用来修改请求的设置，后置处理器则常常用来处理响应的数据。
8. 定时器负责定义请求之间的延迟间隔。

手机京东的自动化实践，为了自动化而自动化？

接口自动化

Jmeter的使用



手机京东的自动化实践，为了自动化而自动化？

UI自动化

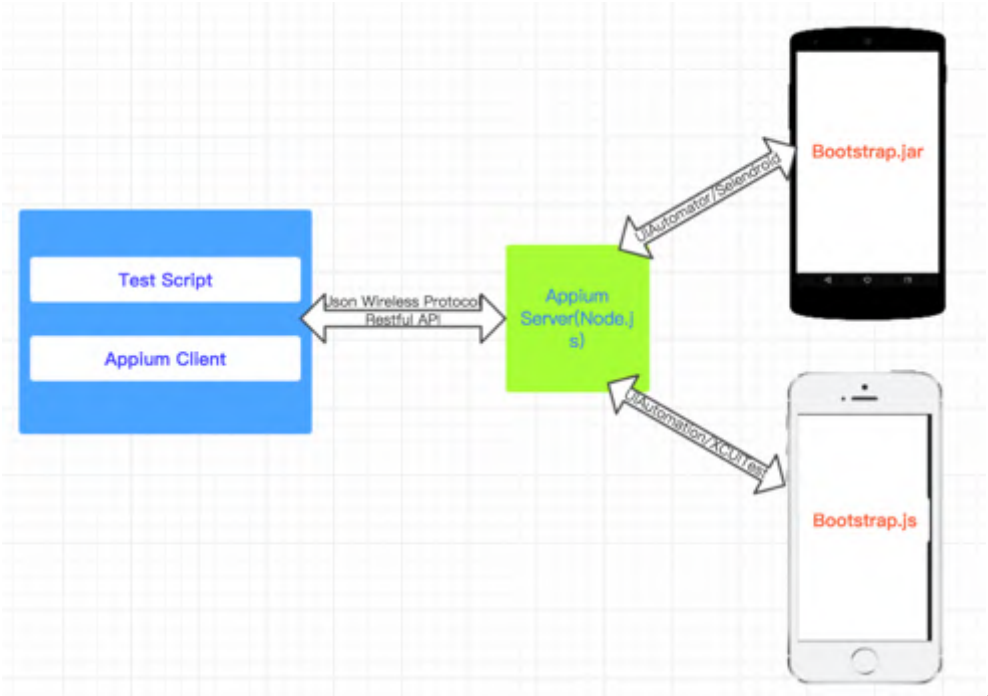
Appium

手机京东的自动化实践，为了自动化而自动化？

UI自动化

Why Appium

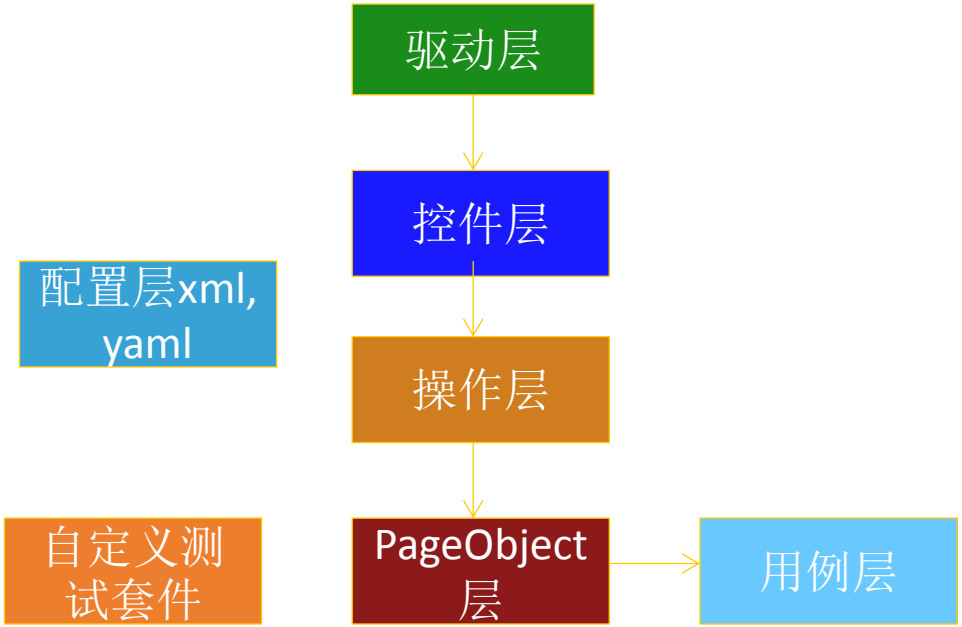
1. 跨平台
2. 跨开发语言
3. 兼容其他工具
4. 非侵入式设计



手机京东的自动化实践，为了自动化而自动化？

UI自动化

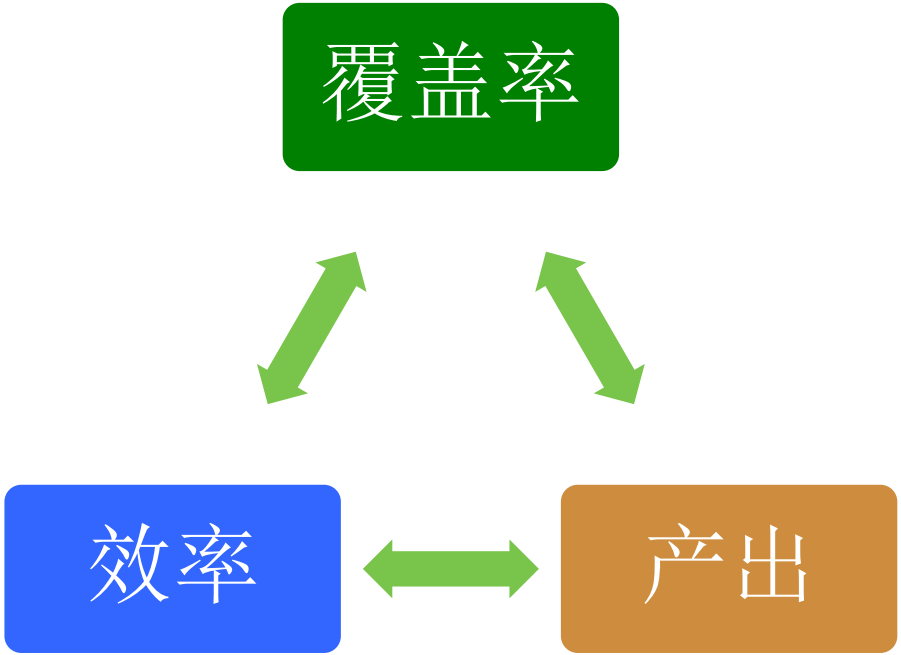
Appium + TestNG



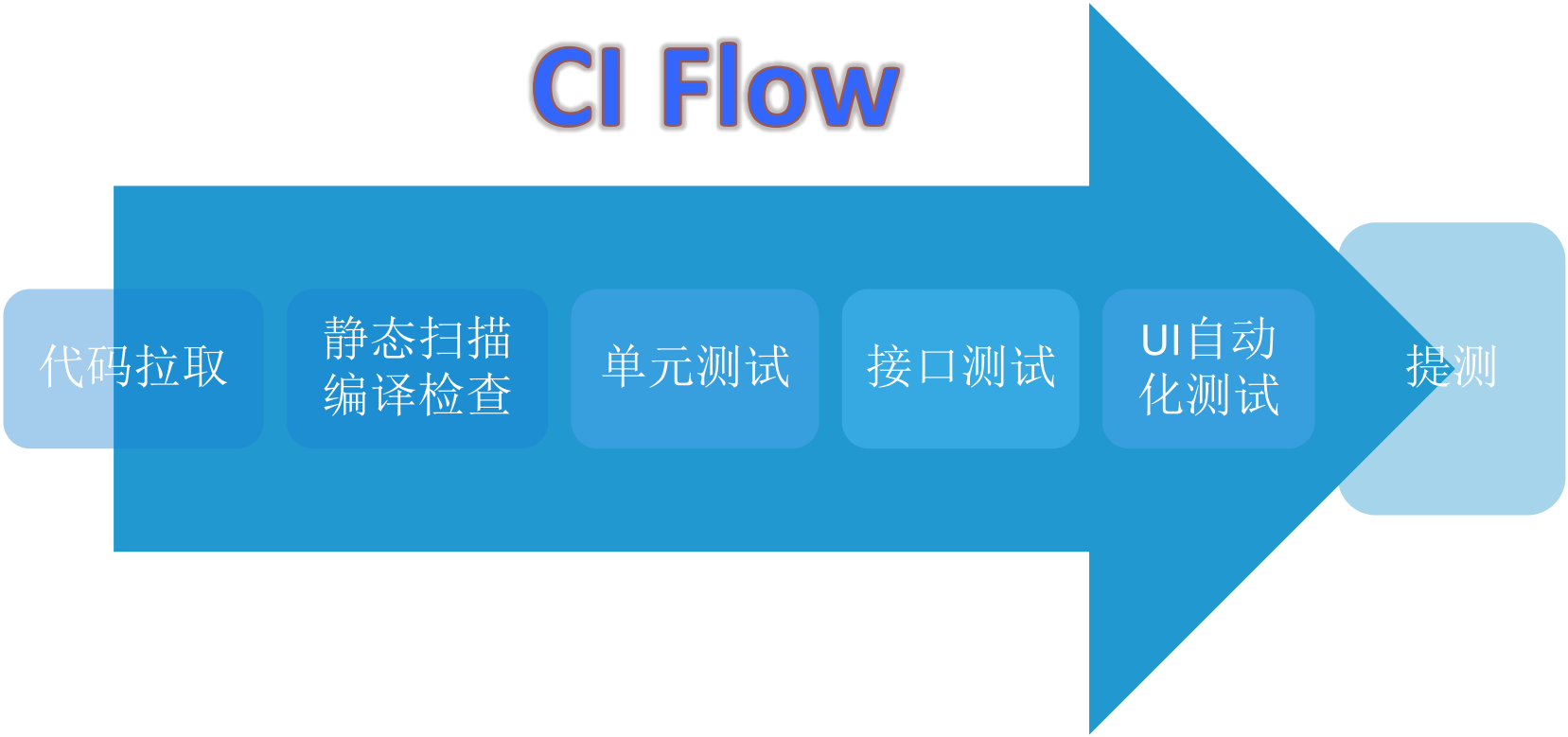
手机京东的自动化实践，为了自动化而自动化？

UI自动化

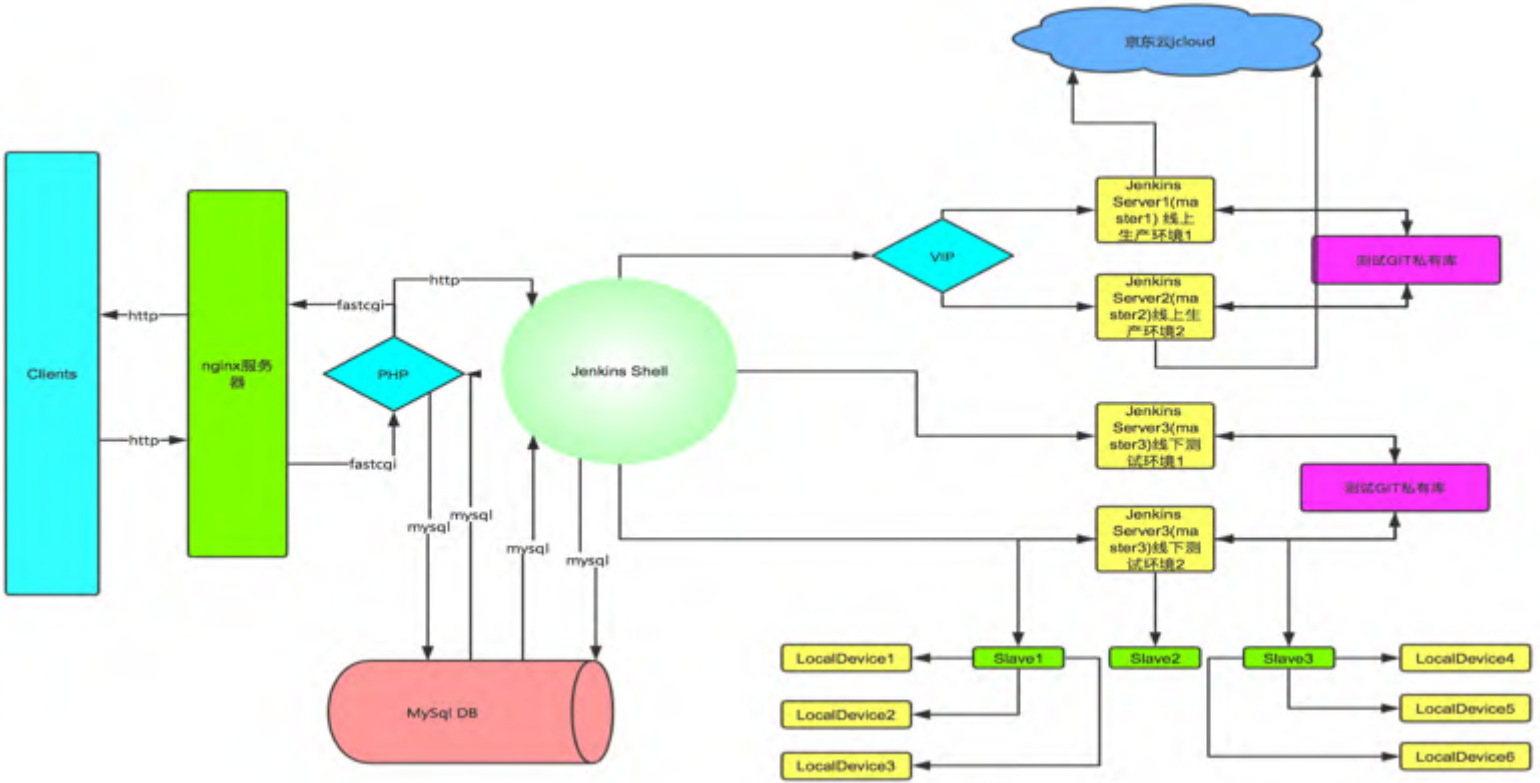
UI自动化策略

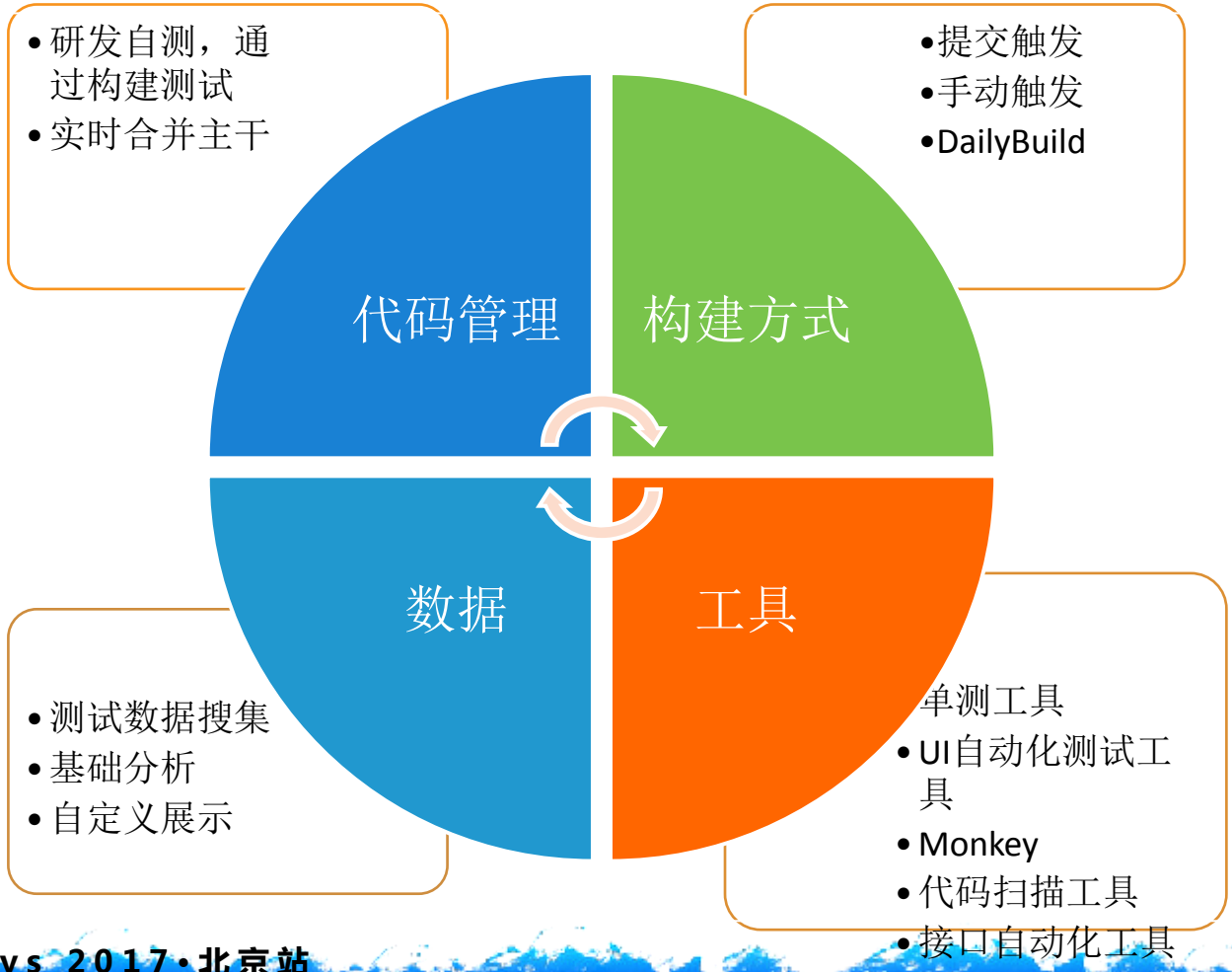


手机京东的持续集成实践，仅仅是为了效率？No



手机京东的持续集成实践，CI-Service Framework




• 研发自测，通过构建测试
• 实时合并主干
• 提交触发
• 手动触发
• DailyBuild
代码管理
构建方式
数据
工具
• 测试数据搜集
• 基础分析
• 自定义展示
单测工具
• UI自动化测试工具
• Monkey
• 代码扫描工具
• 接口自动化工具

手机京东的持续集成实践，仅仅是为了效率？No

CI实践的主要价值体现

测试提早介入项目

- 代码静态扫描&代码编译检查，使得测试介入编码阶段；
- 尽早发现代码问题，降低bug修复成本；

统一测试入口标准

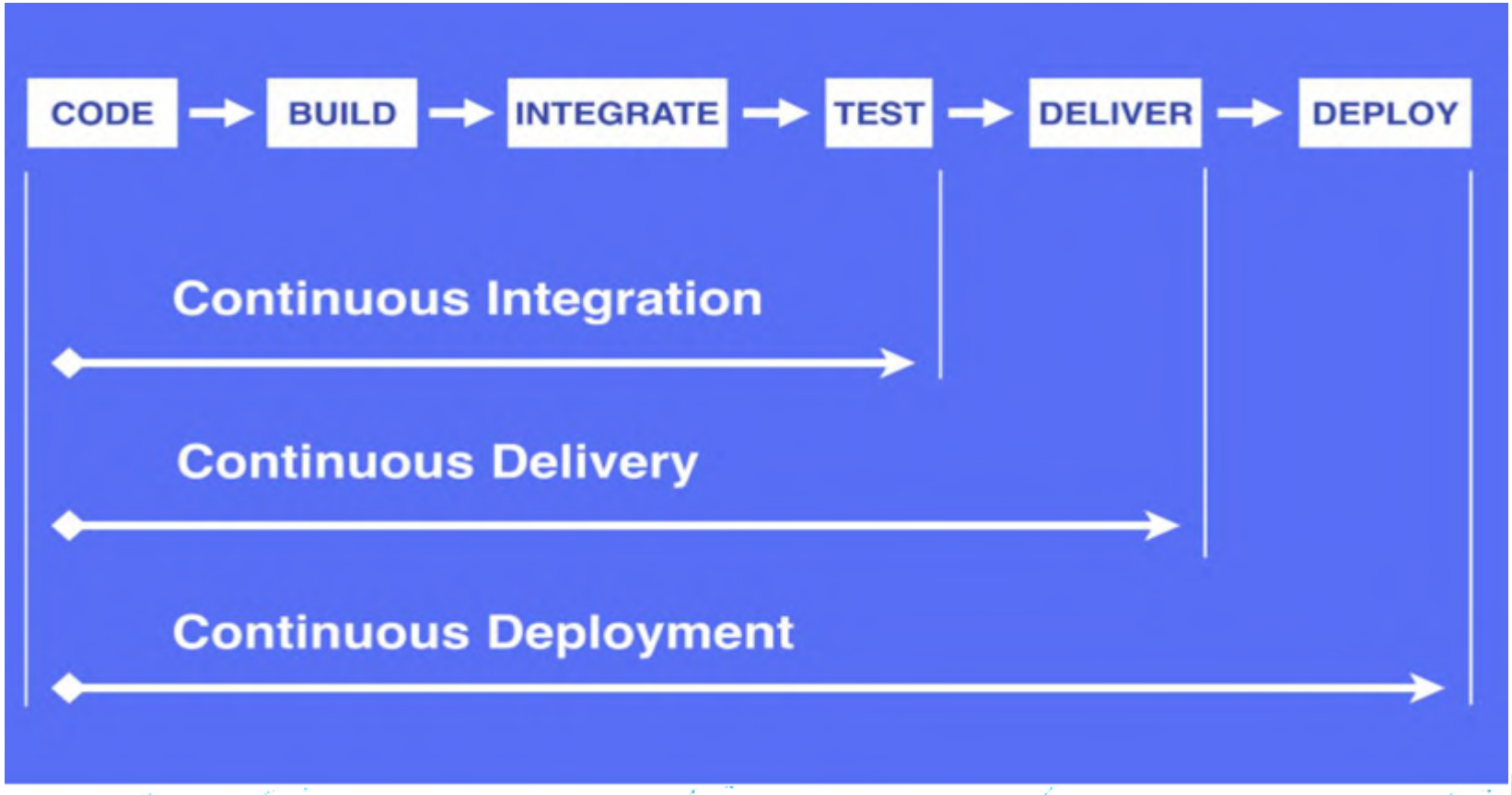
- 统一出包避免出包不一致引起的bug；
- 统一出包提高了测试包的复用性，降低重复操作和沟通成本；

提升测试效率

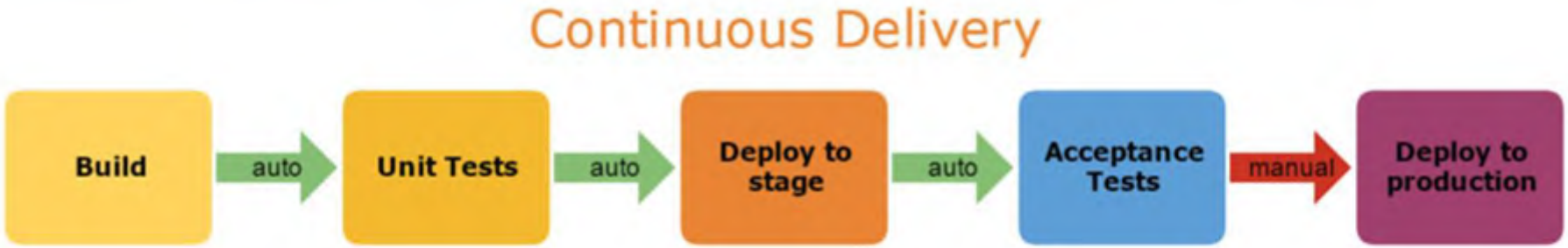
- 工具，增加产品开发过程的透明度，降低沟通成本，效
- 积累开发、测试过程中的数据，测试结果定期分析（定期报告）

未来展望，CI-CD

未来展望，CI-CD


CODE
BUILD
INTEGRATE
TEST
DELIVER
DEPLOY
Continuous Integration
Continuous Delivery
Continuous Deployment

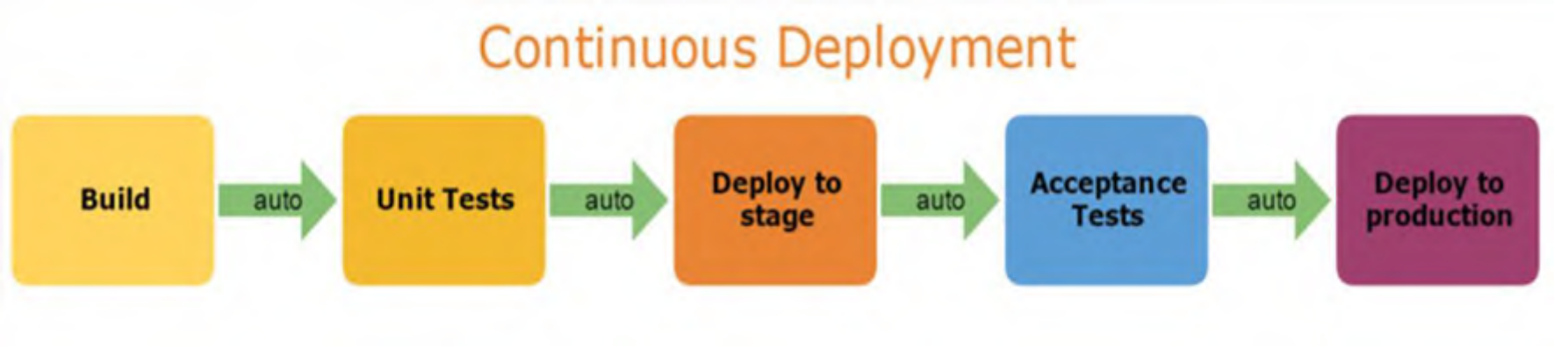
未来展望，持续集成到持续交付



1. 快速发布。能够应对业务需求，并更快地实现软件价值。
2. 编码->测试->上线->交付的频繁迭代周期缩短，同时获得迅速反馈；
3. 高质量的软件发布标准。整个交付过程标准化、可重复、可靠，
4. 整个交付过程进度可视化，方便团队人员了解项目成熟度；
5. 更先进的团队协作方式。从需求分析、产品的用户体验到交互 设计、开发、测试、运维等角色密切协作，相比于传统的瀑布式软件团队，更少浪费。

未来展望，持续集成到持续部署

持续集成到持续部署，我们需要做什么




QA

高效运维社区

GreatOPS Community

GOPS 4月深圳 / 7月北京 / 11月上海

EXIN DevOps Master 认证研修

DevOpsDays 3月北京 / 8月上海

DevOps 企业内训 / 咨询服务

DevOps China 全国巡回技术沙龙

其他量身定制服务项目

商务经理：刘静女士
电话 / 微信：13021082989
邮箱：liujing@greatops.com

Thanks

荣誉出品

高效运维社区

国际最佳实践管理联盟