

02

OPEN ORIENTED

凹凸实验室

# 复杂业务前端团队的进化之路

谢晓立

凹凸实验室

[aotu.io](http://aotu.io)





02

OPEN ORIENTED  
凹凸实验室

京东 —  
商城用户体验设计部 —  
多终端研发部



# 复杂业务前端团队的 进化之路

PC

微信手Q

营销H5

小程序

1

工具迭代

2

流程优化

# 复杂业务前端团队的 进化之路

PC

微信手Q

营销H5

小程序

1

工具迭代

2

流程优化

前端工程化

解决

如何提升 开发体验

静态资源管理

如何进行 性能优化

## 我们是如何做好前端工程化和静态资源管理

by [luckyadam wengeek](#) on 2016-07-19

随着互联网的发展，我们的业务也日益变得更加复杂且多样化起来，前端工程师也不再只是做简单的页面开发这么简单，我们需要面对的十分复杂的系统性问题，例如，业务愈来愈复杂，我们要如何清晰地梳理；团队人员愈来愈多，我们要如何更好地进行团队协作；功能愈来愈多，我们要如何保证页面的性能不至于下降，等等。所有的这些都可以归结为如何提升开发体验和性能问题。



统一前端工程化  
解决方案

## 开发体验

功能齐全的 自动化工具

轻量 组件化 功能

## 性能优化

首屏 HTML代码 直出

基于楼层的 懒加载 功能

# 自动化工具

创建项目

本地开发

上线

项目结构生成

less/scss编译

页面片生成

资源压缩

文件md5戳

资源合并

路径替换

直出处理

一键部署

热更新预览

# 轻量组件化

```
widget
├── images      - widget的图片目录
├── widget.json - widget的说明文件
├── widget.css  - widget的css
├── widget.js   - widget的js
└── widget.html - widget的ejs模板
```

组件结构

```
<%= widget.load('widget', {
  content: '楼层内容'
}) %>
```

组件引用



# 轻量组件化

## 不完全的组件化实现

```
widget
├── images      - widget的图片目录
├── widget.json - widget的说明文件
├── widget.css  - widget的css
├── widget.js   - widget的js
└── widget.html - widget的ejs模板
```

组件结构

```
widget.html  字符串替换
js、css      字符串拼接（combo）
```

编译结果

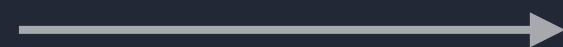
# HTML代码直出

```
<%= widget.load('floorhd', {  
  title: '楼层标题'  
}) %>
```

```
<%= widget.load('floor', {  
  content: '楼层内容'  
}) %>
```

源代码

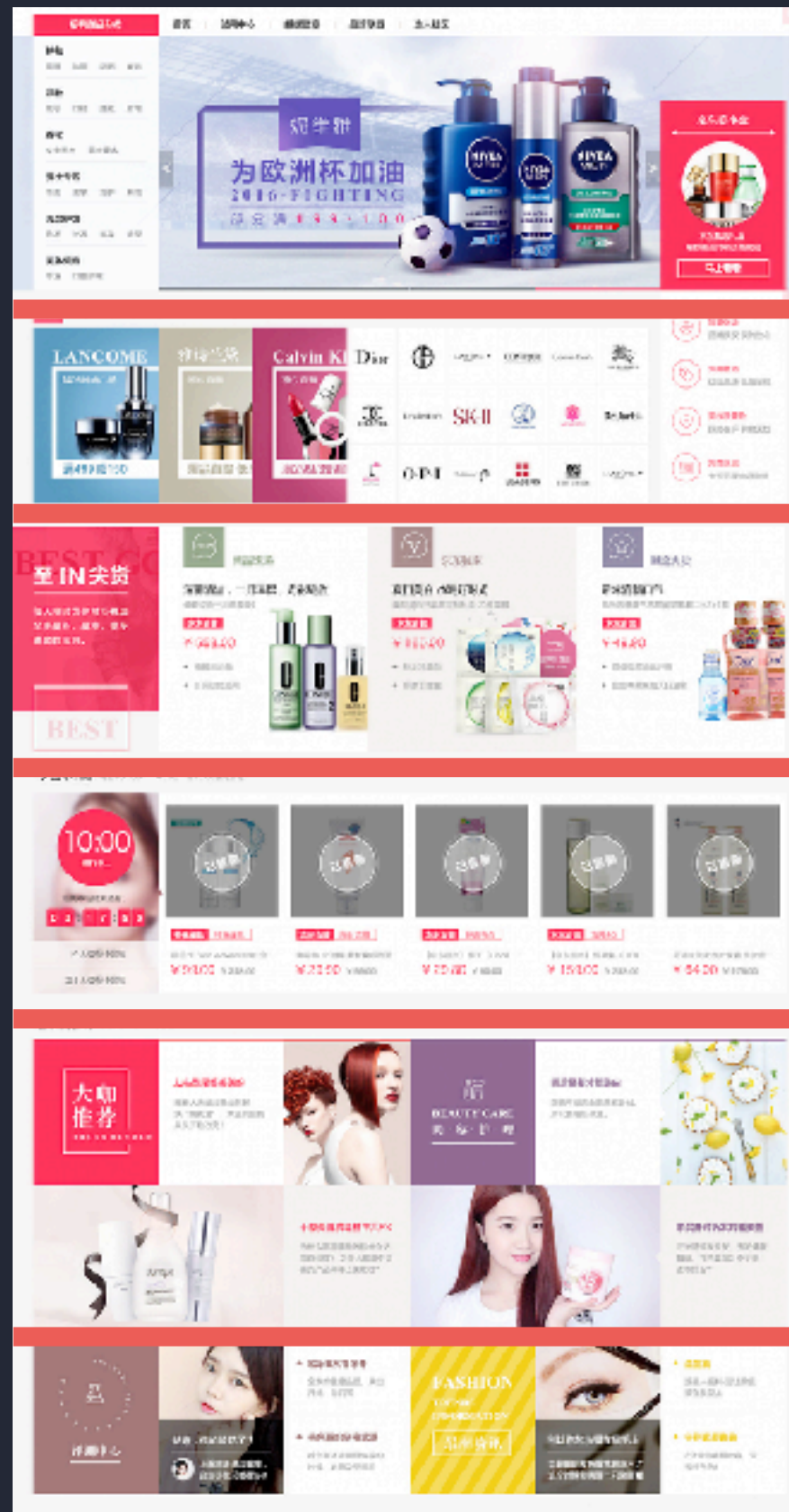
编译



```
<div class="floorhd">  
  楼层标题  
</div>  
<div class="floor">  
  楼层内容  
</div>
```

编译结果

# 楼层懒加载



页面 = 楼层 × N

```
<!-- index.html -->
<div
  class="mod_lazyload"
  data_tpl="floor1_tpl"
/>
<div
  class="mod_lazyload"
  data_tpl="floor2_tpl"
/>
<div
  class="mod_lazyload"
  data_tpl="floor3_tpl"
/>
```

编译结果





页面 = 楼层  $\times$  N?

页面 = 组件  $\times$  N!

# 如何提升 开发体验

不完全的组件化

# 如何进行 性能优化

不通用的优化方案

# 如何提升 开发体验

更完善的组件化

# 如何进行 性能优化

更通用的优化方案



更完善的组件化

## 业界流行的技术



组件化

Virtual/  
ShadowDOM

数据绑定

ESNext  
强类型

```
$( '.article' )  
  .find( 'span' )  
  .html( 'a' )
```

```
<div class="J_f J_lazyload"  
  data-tpl="portal_tpl">  
  <%= widget.load('portal') %>  
</div>
```

**VS**

```
this.setState({  
  html: 'a'  
})
```

```
<LoadablePortal />
```



老板说  
那就用起来



框架	版本	star	兼容性	性能 *	体积 *	开源协议	学习成本 *	迁移成本	源码类型
React	16.1.0	83931	IE 9+	1.52	31.9 Kb	MIT *	中	低	Flow
Angular	5.0.0	31250	IE 9+	1.46	61.7 Kb	MIT	高	高	TypeScript
Vue	2.5.0	77368	IE 9+	1.29	23.5 Kb	MIT	低	中	Flow

对了 记得测测兼容性  
至少到IE8吧





框架	版本	star	兼容性	性能 *	体积 *	开源协议	学习成本 *	迁移成本	源码类型
React	16.1.0	83931	IE 9+	1.52	31.9 Kb	MIT *	中	低	Flow
Angular	5.0.0	31250	IE 9+	1.46	61.7 Kb	MIT	高	高	TypeScript
Vue	2.5.0	77368	IE 9+	1.29	23.5 Kb	MIT	低	中	Flow



自己 造个轮子

# 怎样造轮子

几个螺丝 什么型号的螺丝

## 方法论

使用什么思想来构建组件化

使用哪些型号的轮胎

## 接口设计

如何最大化利用资源



# 方法论

使用什么思想来构建组件化

Virtual DOM

数据绑定

# 接口设计

如何最大化利用资源

```
import React, {  
  Component  
} from 'react'  
  
class App extends Component {  
  componentDidMount() {}  
  render() {  
    return <div />  
  }  
}
```

npm i material-ui ...

# 与React相同的语法

## 兼容IE8

与React相同的语法

兼容IE8

彪悍的性能

diff 算法优化  
同步 diff&patch

羞涩的体积

9kB gzipped



类React 高性能  
前端框架

# Nerv

<https://nerv.aotu.io>

NervJS / nerv

Watch 121 Unstar 4,176 Fork 204

Code Issues 38 Pull requests 0 Projects 0 Wiki Insights Settings

A blazing fast React alternative, compatible with IE8 and React 16. <https://nerv.aotu.io> Edit

nerv react preact inferno jsx virtual-dom vdom reactjs javascript framework frontend nervjs typescript Manage topics

777 commits 11 branches 59 releases 12 contributors MIT



# 如何提升 开发体验

更完善的组件化

# 如何进行 性能优化

更通用的优化方案

更通用的优化方案

## 借助开源的力量

懒加载 `npm i react-lazyload`

延迟加载 `npm i react-loadable`

状态管理 `npm i redux nerv-redux redux-thunk`

开发体验

性能优化

业务形态1

业务形态2

...

业务形态N



通用优化方案



**Nerv**  
统一技术栈



**Athena**  
统一开发工具

老板说  
做得不错  
小程序的业务 也给你们搞了



MINA

# MINA目录结构



## 存在的问题



规范不统一



孱弱的字符串模板



不完全的ESNEXT



依赖管理混乱



落后的开发方式



代码组织混乱

分隔符命名?  
全小写命名?

session-from	String		会话来源	open-type="contact"	1.4.0
send-message-title	String	当前标题	会话内消息卡片标题	open-type="contact"	1.5.0
send-message-path	String	当前分享路径	会话内消息卡片点击跳转小程序路径	open-type="contact"	1.5.0
send-message-img	String	截图	会话内消息卡片图片	open-type="contact"	1.5.0
<u>show-message-card</u>	Boolean	false	显示会话内消息卡片	open-type="contact"	1.5.0
<u>bindcontact</u>	Handler		客服消息回调	open-type="contact"	1.5.0
bindgetphonenumber	Handler		获取用户手机号回调	open-type="getphonenumber"	1.2.0

## 规范不统一

组件 lifecycle

created

attached

ready

页面 lifecycle

onLoad

onShow

onReady



## 落后的开发模式

难以使用redux

手动图片、JS压缩

无法使用TypeScript

难以使用npm依赖

不支持less/scss

来源不明的wx, Page, ...

不支持单文件组件

不完全的esnext

# React特性



代码规范



直观的JSX语法



天生组件化



开源社区的力量

开发体验

性能优化

业务形态1

业务形态2

...

业务形态N

小程序？



通用优化方案



**Nerv**  
统一技术栈



**Athena**  
统一开发工具

能否统一使用 React 写

```

import Taro, { Component } from '@tarojs/taro'
import { View, Text, Button } from '@tarojs/components'
import './A.scss'

export default class A extends Component {

  componentWillMount () { }

  componentDidMount () { }

  componentWillReceiveProps (nextProps) { }

  shouldComponentUpdate () { }

  componentWillUnmount () { }

  componentDidShow () { }

  componentDidHide () { }

  onClickHandler () {
    this.props.onClick()
  }

  render () {
    return (
      <View className='a'>
        <Button onClick={this.onClickHandler.bind(this)}>点我</Button>
        <Text>a component {this.props.t}</Text>
      </View>
    )
  }
}

```

React语法

```

Component({
  properties: {
    myProperty: { // 属性名
      type: String
    },
    myProperty2: String // 简化的定义方式
  },

  data: {}, // 私有数据, 可用于模版渲染

  // 生命周期函数, 可以为函数, 或一个在methods段中定义的方法名
  ready: function () {},

  attached: function() {},

  moved: function() {},

  detached: function() {},

  methods: {
    onMyButtonTap: function(){
      this.setData({
        // 更新属性和数据的方法与更新页面数据的方法类似
      })
    }
  }
})

```

MINA语法



多端统一开发框架

write once, run everywhere

快

速开发小程序

解决小程序开发各种痛点

多

终端适配

一套代码适配小程序/H5/RN等终端



## 这些项目都在用…



TOPLIFE



玲珑智能设计



一起有局



京东购物



罗网指数



脑洞阅读



喵呜星球



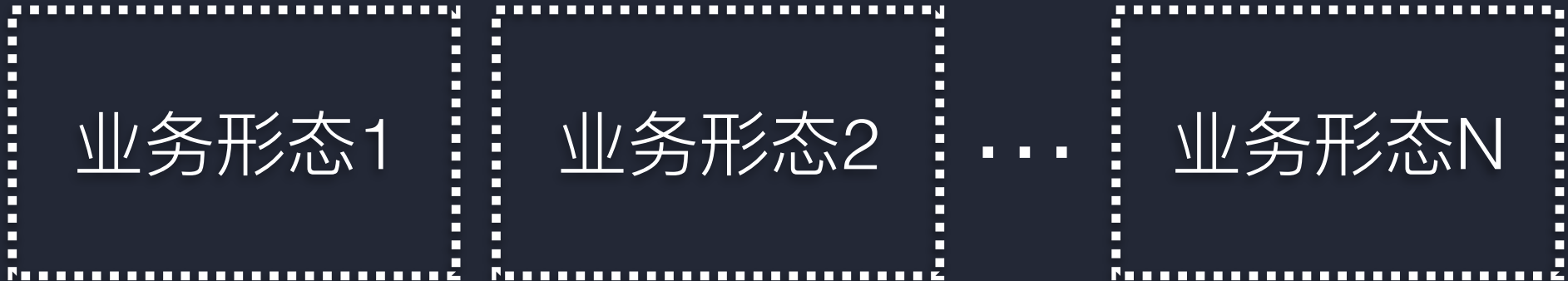
小来打卡PRO



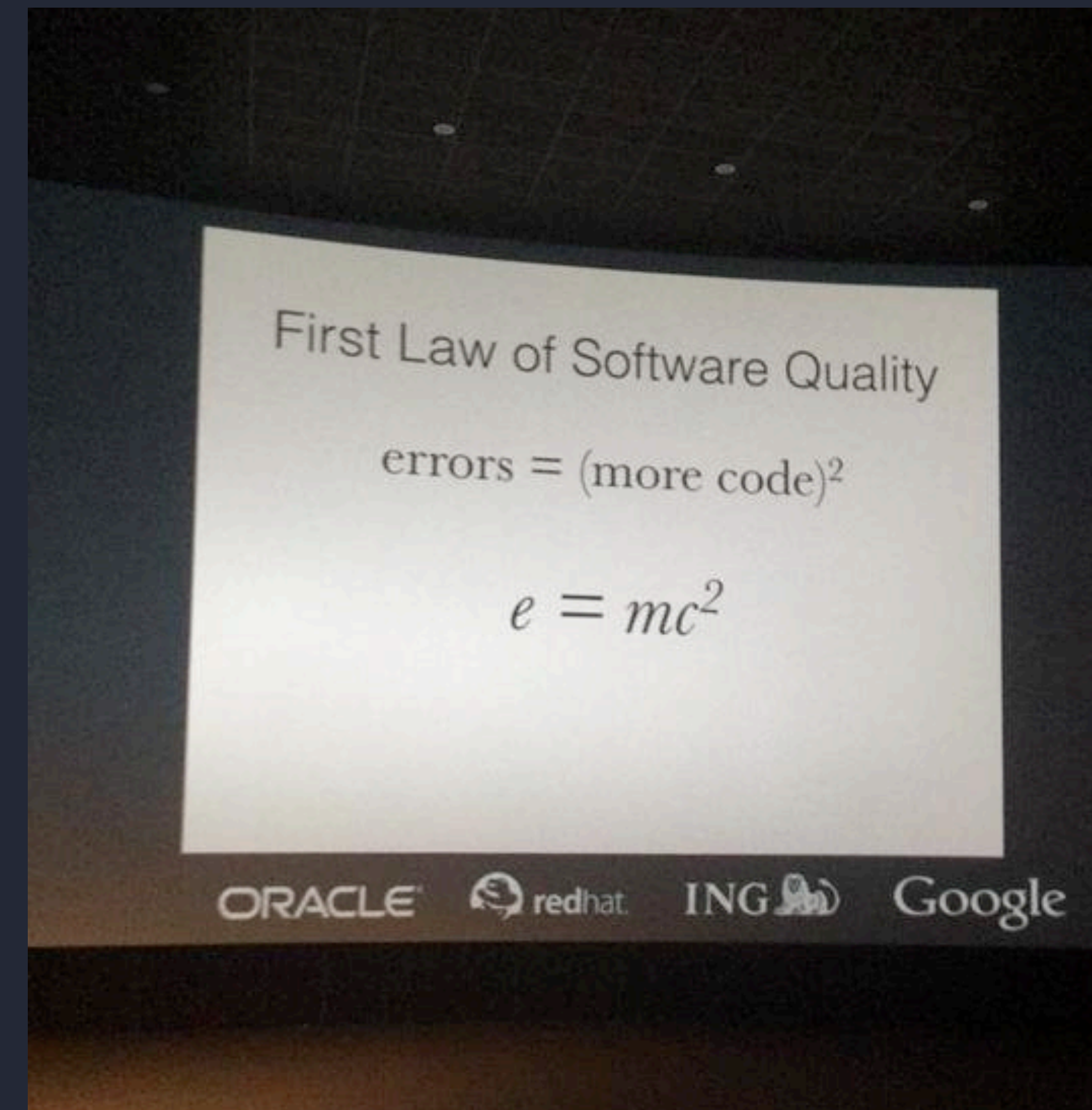
昨夜部落

更多案例  
开发/重构中  
敬请期待

开发体验  
性能优化



工作效率提升  
但开发质量如何保证？



“软件质量第一定律”  
 $\text{errors} = (\text{more code})^2$

# 复杂业务前端团队的 进化之路

PC

微信手Q

营销H5

小程序

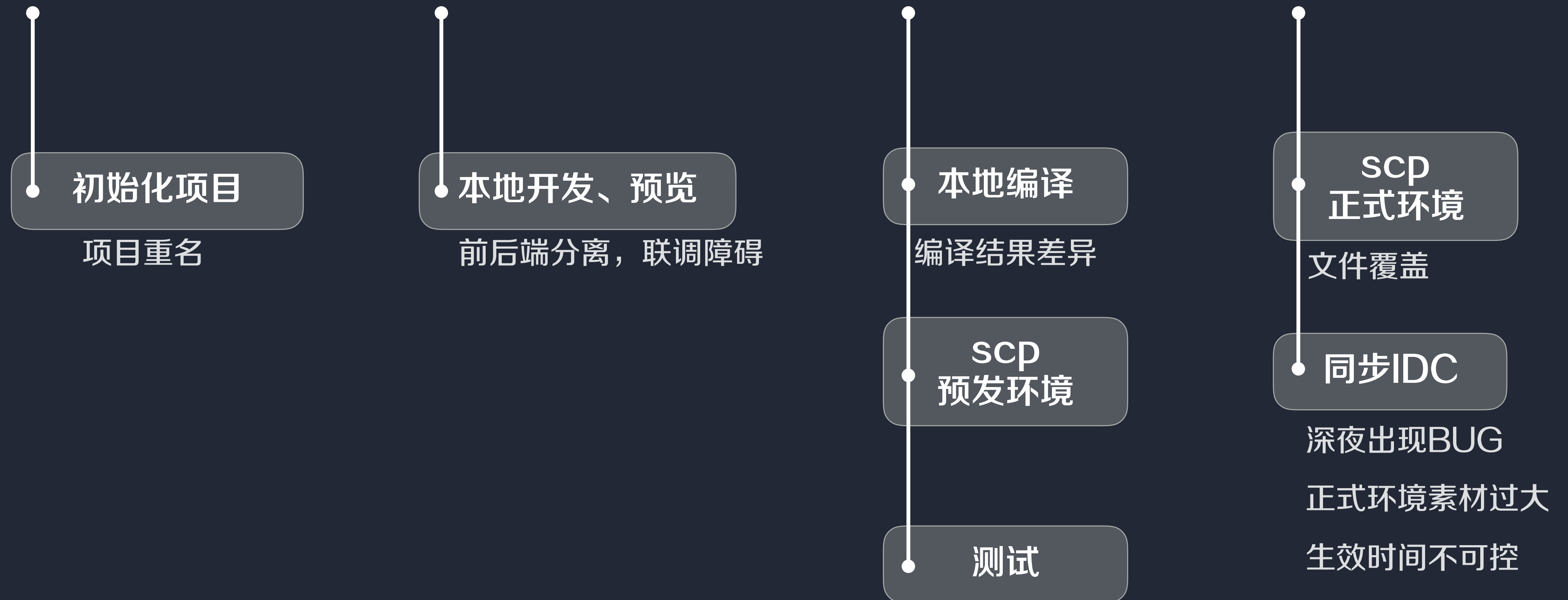
1

工具迭代

2

流程优化

# 原始的研发流程



墨菲定律之三

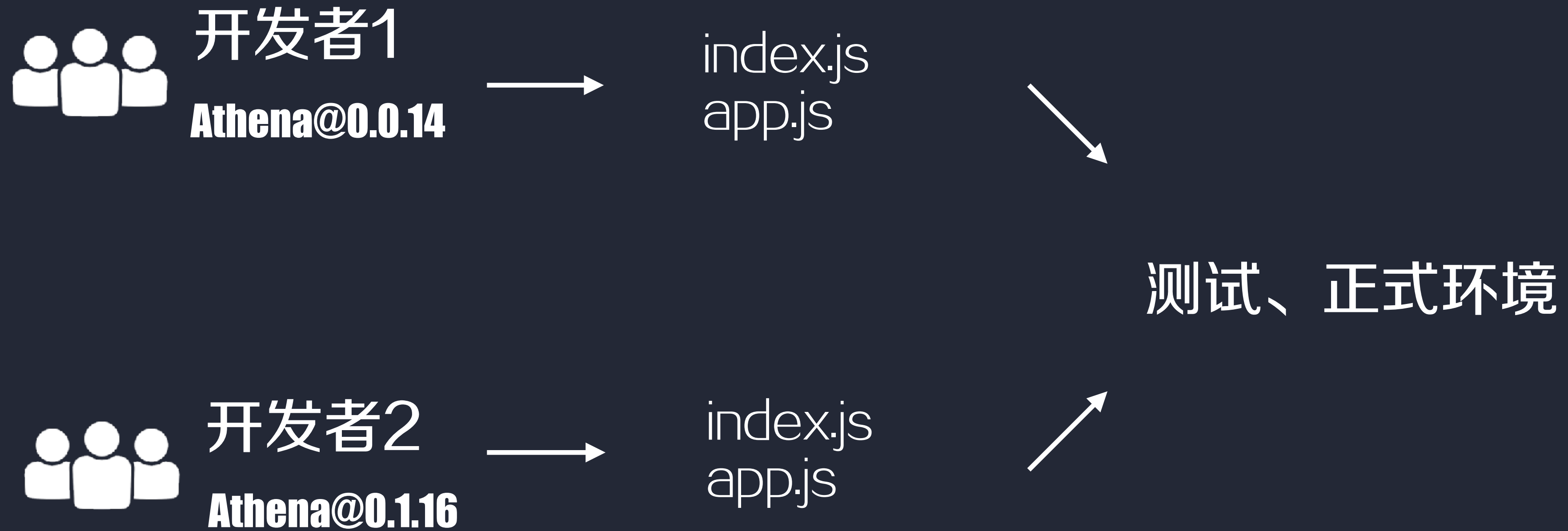
有可能出错的事情  
一定会出错



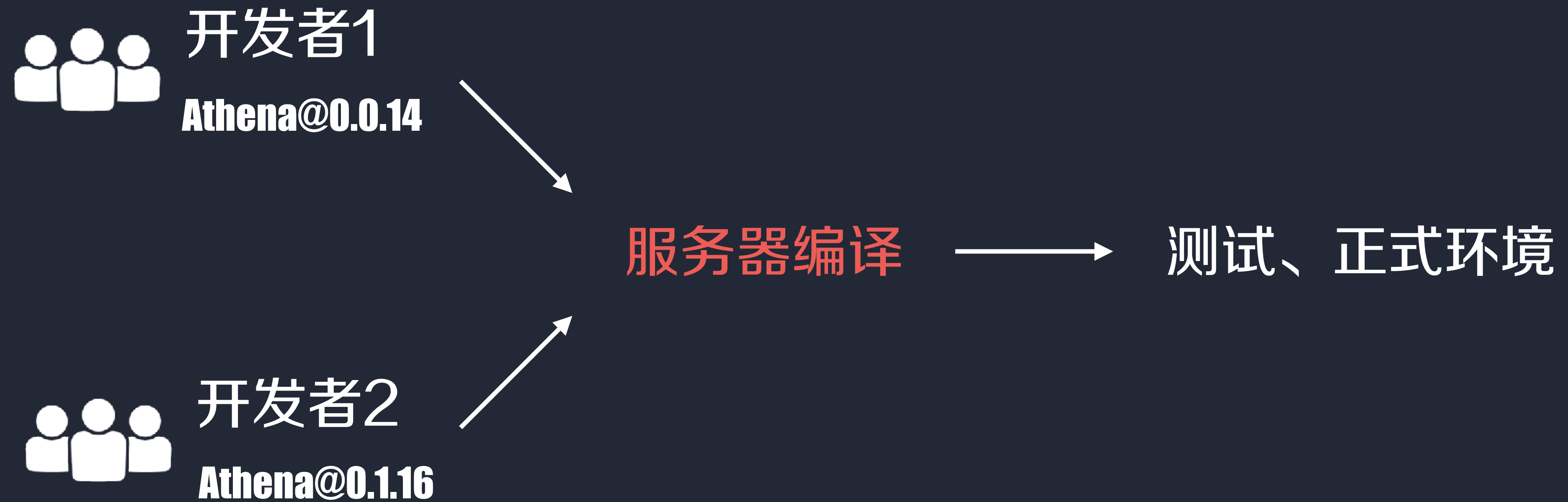
让错误不可能发生



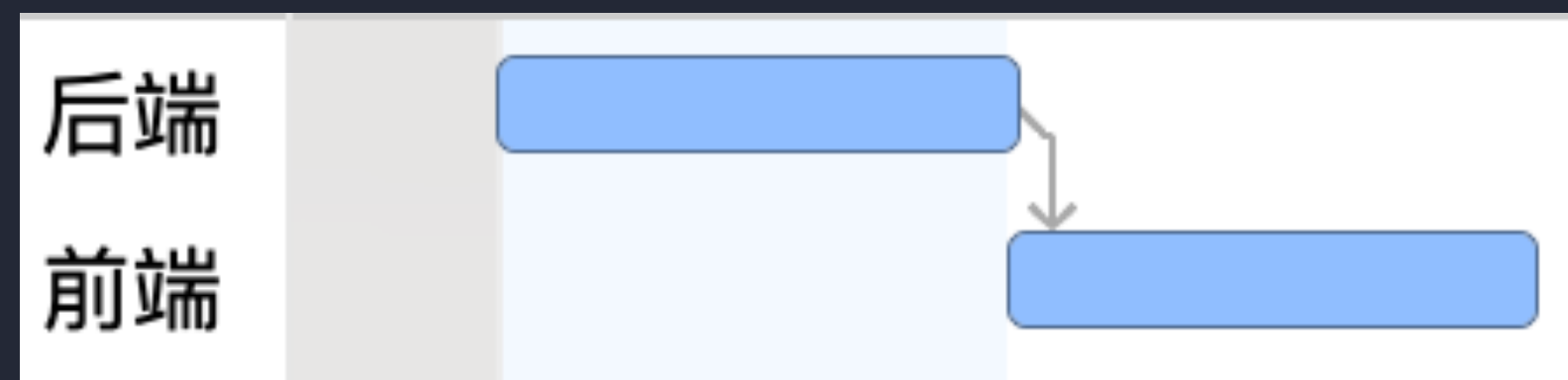
## 开发工具版本不一



## 使用服务器端编译



## 进度耦合



## 沟通分歧

```
{ "code": 0, "data": { } }
```

```
{ "code": "0", "data": { } }
```

# 接口文档&Mock平台

默认分组

**POST** Add a new pet to the store

**PUT** Update an existing pet

**GET** Finds Pets by status

**GET** Finds Pets by tags

**GET** Find pet by ID

**POST** Updates a pet in the store with for...

**DEL** Deletes a pet

**POST** uploads an image

**GET** Returns pet inventories by status

**POST** Place an order for a pet

**GET** Find purchase order by ID

## Add a new pet to the store

使用中



25 天前编辑过该接口文档

### 接口负责人

apiteam@swagger.io

### 请求 URL

**POST** https://petstore.swagger.io/v2/pet

深夜 BUG

数据监控

增加监控

监控页面	监控节点	监控阈值	操作
基础信息	onLoad时间	3000	 
兜底请求	秒杀	0.03	 
兜底请求	首焦	0.15	 
兜底请求	优惠券	0.01	 
兜底请求	今日推荐	0.03	 
兜底请求	大促	0.01	 

数据监控

检测模块可用性  
保证页面 安全

素材监控

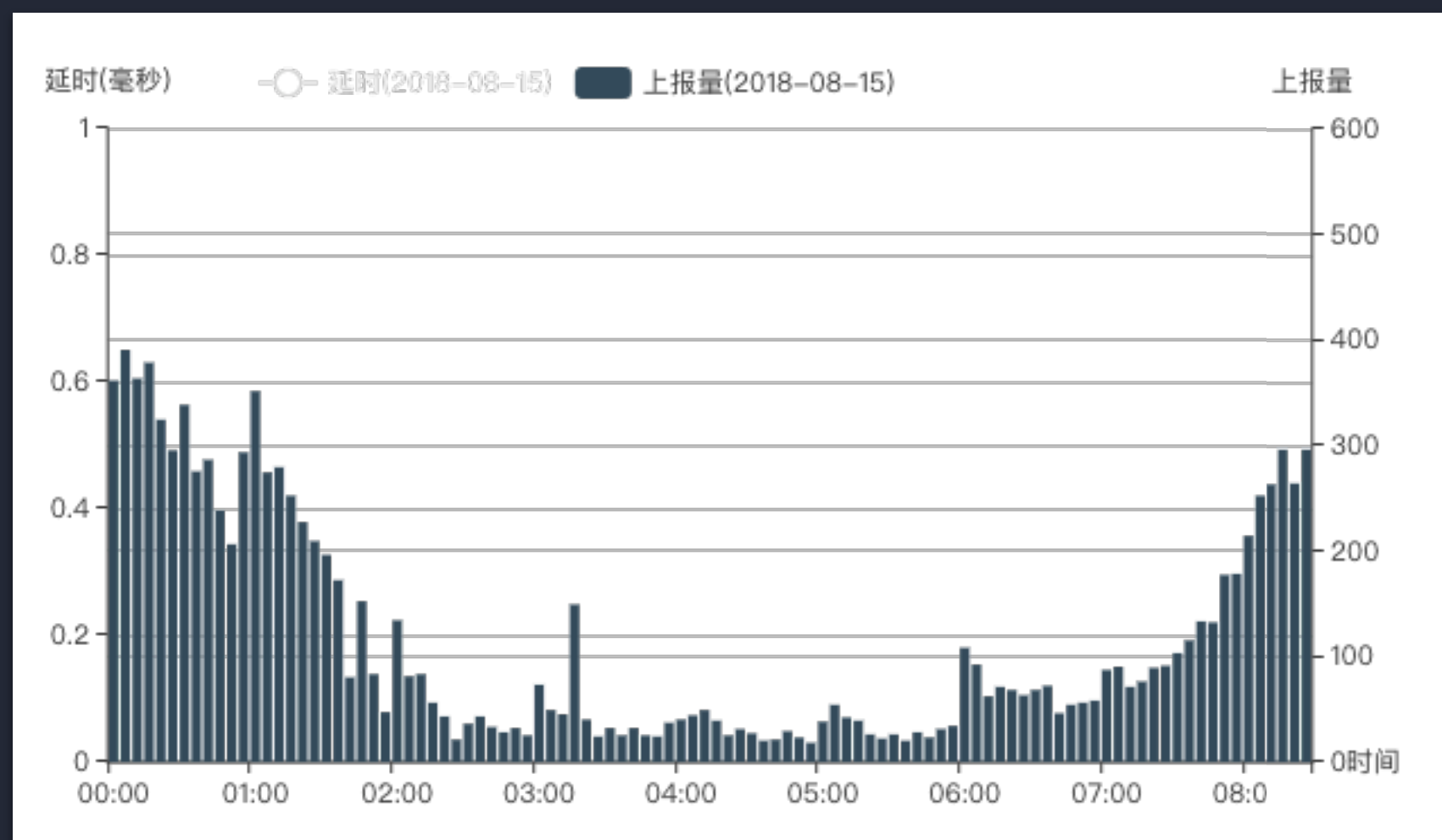
监控素材尺寸与体积  
确保页面 性能

素材监控

增加监控

素材名称	选择器	事件类型	操作次数	特定区域	特定规则	操作
轮播图	.J_slider_control_next	click	7	.J_slider_item_img	790x340*250	 
头部导航	.J_tab_head_item	mouseover	(无)	(无)	(无)	 
Logo	.J_pt_logo_next	click	1	(无)	(无)	 
楼层封面大图	.pt_cover	click	(无)	.pt_cover_img	100x100*100	 
十二宫格	.service	mouseover	4	.service_item	40x40*20	 
商品图	.item	(无)	(无)	.item_img	120x120*80	 

## 24h监控告警服务



请求失败

模块隐藏

接口超时

2018/8/4 00:45

### 告警通知

8月4日

京东首页2017版

时间：2018-08-04 00:45

内容：兜底请求：今日推荐超过阈值0.03至0.04(13544)

Athena监控

2018/8/4 01:15

### 告警通知

8月4日

京东首页2017版

时间：2018-08-04 01:15

内容：兜底请求：今日推荐超过阈值0.03至0.03(10025)

Athena监控



























似乎完美了

# DOM监控

## 监控页面DOM结构

### 确保页面 完整

页面规则 不懂规则怎么写？[去看看](#)»

规则名称	匹配元素 ?	规则类型	处理器	期望值	操作
首焦	#J_focus.slider_item	数量 (length)	等于 (=)	8	   
今日推荐	.rec_item	数量 (length)	等于 (=)	3	   
智能	#J_tech.box	数量 (length)	等于 (=)	6	   
家居	#J_house.box	数量 (length)	等于 (=)	6	   
生活	#J_life.box	数量 (length)	等于 (=)	6	   
时尚	#J_vogue.box	数量 (length)	等于 (=)	9	   
直播	#J_live.box	数量 (length)	等于 (=)	3	   

自动化测试 [增加测试](#)

用例名称	参数列表	期望结果	操作
打开url	https://www.jd.com	(无)	 
登录	(无)	(无)	 
点击元素	.service_item a:nth-child(0)	.service_frame:nth-child(0)样式符合条件	 
hover元素	.good_item:nth-child(0)	.good_item:nth-child(0) img样式符合条件	 
输入	#search ， 苹果	.search_helper .item数量符合条件	 

## 自动UI测试

用户视角的UI监控

保障页面 100%可用

低效 浪费



总计30+系统



闲置系统40%



55%项目  
覆盖不全

整合

整合

不是单纯的hao123  
而是实现工具间的**高效衔接**

## 系统化

每个人都知道该怎么做

## 自动化

不需要care怎么做

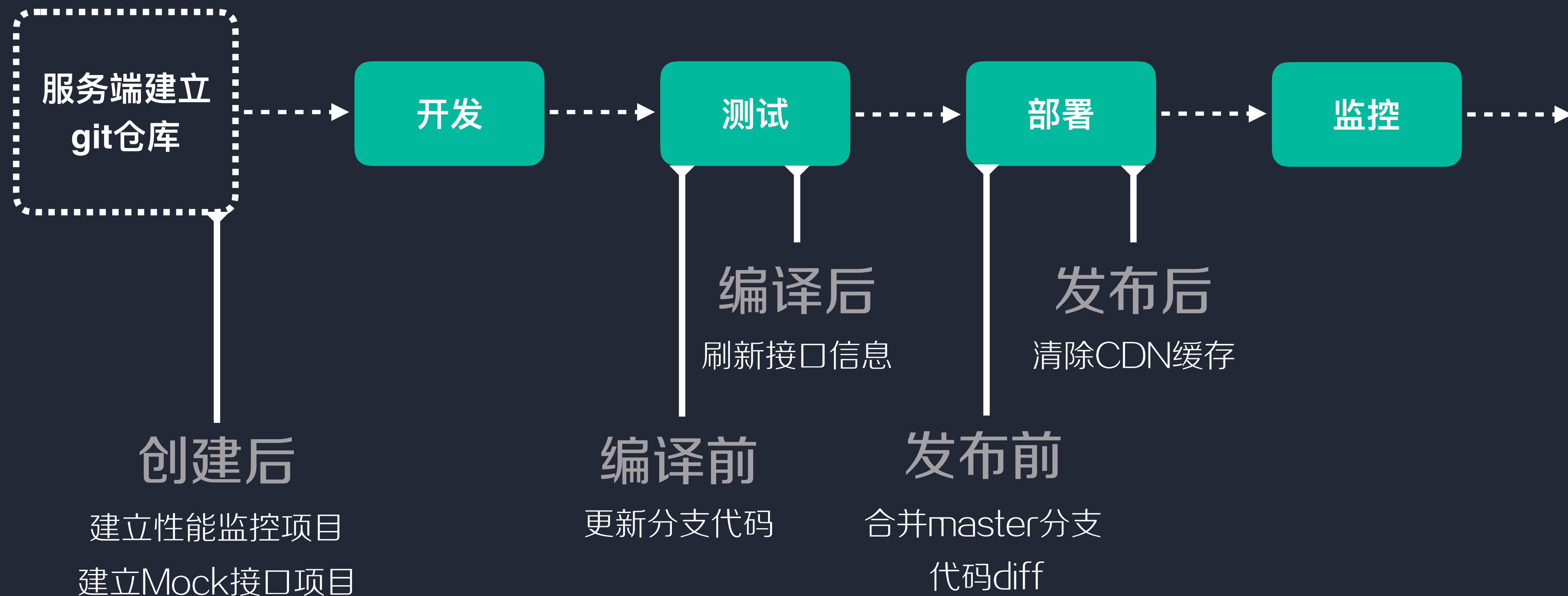
# 系统化

每个人都知道该怎么做



# 自动化

不需要care怎么做





## 基于tapable的插件化功能

```
project.hooks.lifecircle = new HookMap(() => new SyncHook(["projectInfo"]))
project.hooks.lifecircle
  .for('create').tap("create git repository", (projectInfo) => {})
project.hooks.lifecircle
  .for('create').tap("create backup project", (projectInfo) => {})
project.hooks.lifecircle
  .for('create').tap("create mock project", (projectInfo) => {})
project.hooks.lifecircle
  .for('create').call(projectInfo)
```

统一的开发工具 + 通用的开发流程

覆盖从项目创建到上线监控的完整体系





[aotu.io](http://aotu.io)