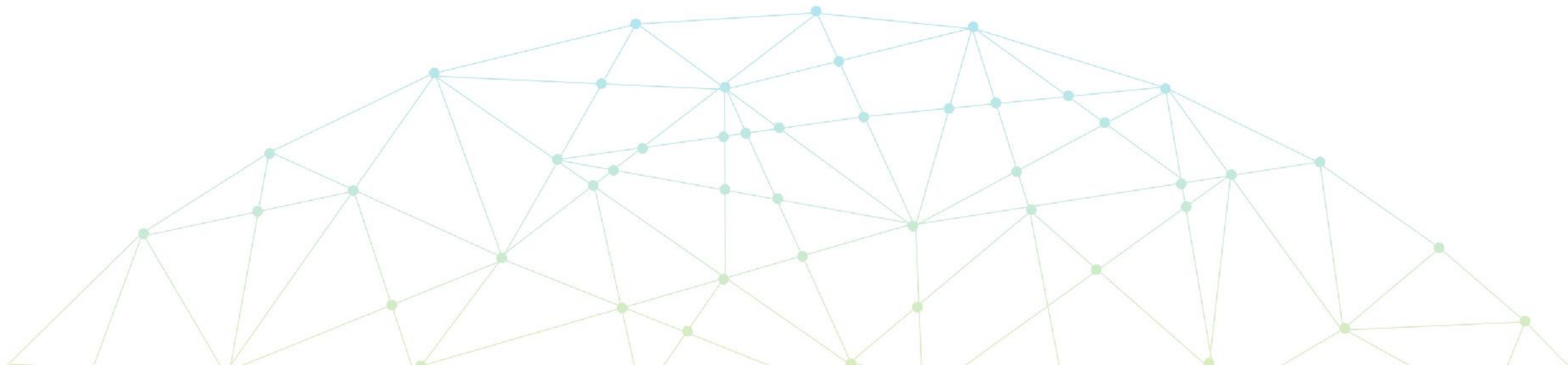




内涵段子  
哥从此不再寂寞

# 内涵段子的稳定性建设

今日头条 曹建栋





# About Me

2011年本科毕业于东北大学

2011年7月加入 Baidu. 历任百科、贴吧资深研发工程师

负责目前百度内最大的私有云的研发管理工作

2014年2月加入今日头条，工作至今

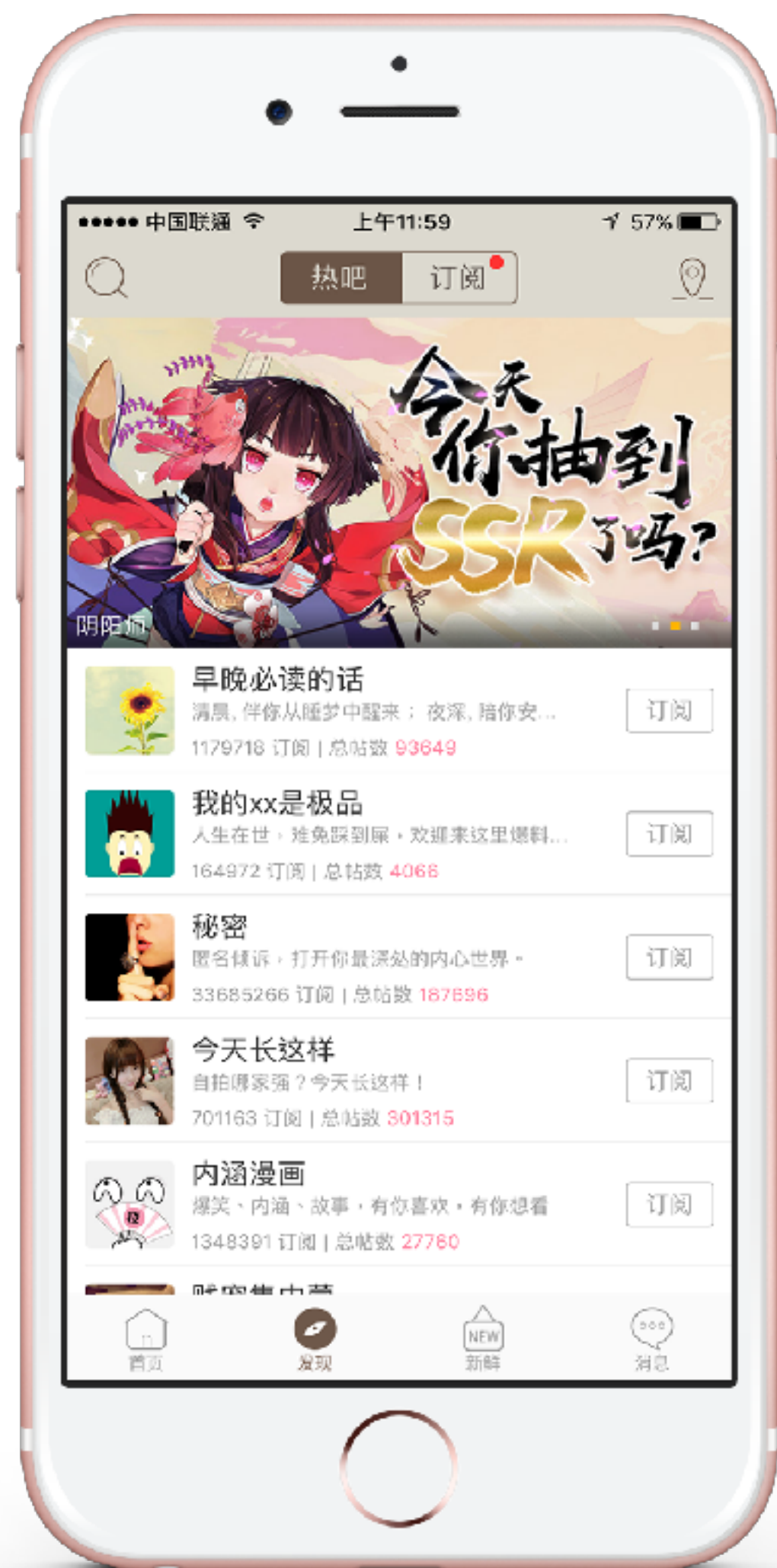
负责新业务方向的探索

**GIAC** | BEIJING  
Dec.12.16-17



[thegiac.com](http://thegiac.com)





## 多主题

200+的“兴趣吧”，18种兴趣纬度，360度娱乐视角！

## 原创内容

短视频、图片、段子、神评论.....

## 个性化推荐

最适合你的，才是“好段子”！

# 段子哥的成长

- 2012.4  
第一版  
内涵段子上线



- 2012-2013  
DAU:41万  
在线时长:18min

- 2013年  
发布14个版本  
全新UI、视频、私信、关注等



- 2013-2014  
DAU:170万  
在线时长:27min

- 2014年  
发布8个版本  
多主题内容、个性化推荐



- 2016.10  
DAU:640万  
在线时长:55min



## 关键词

- 绿色发布
- 自动演练平台
- 文化机制

**GIAC** | BEIJING  
Dec.12.16-17

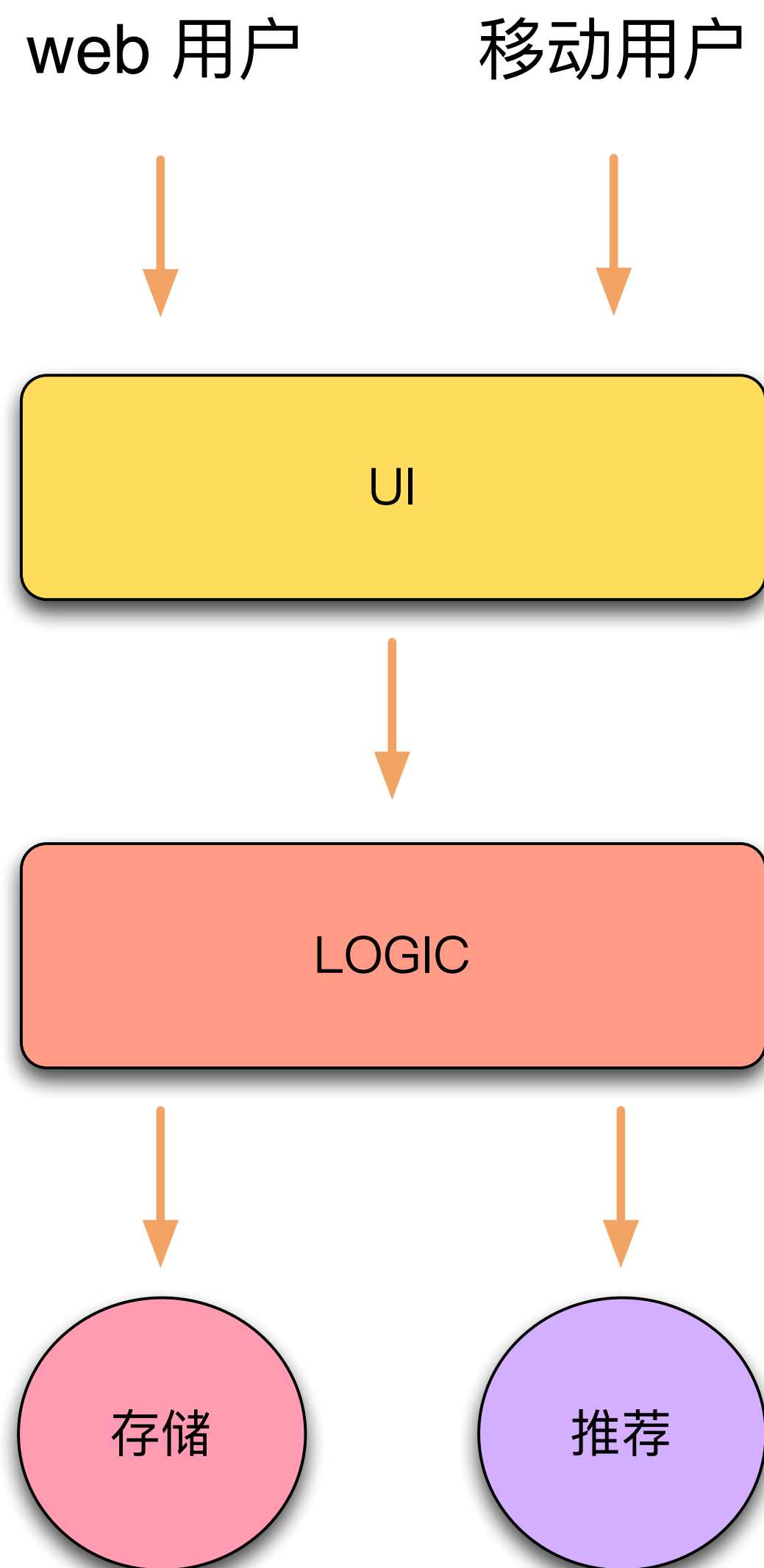


[thegiac.com](http://thegiac.com)



# 历史架构

- 典型的三层架构
- 开发部署非常便捷
- 问题
  - 耦合严重（开发/部署）



# 历史架构

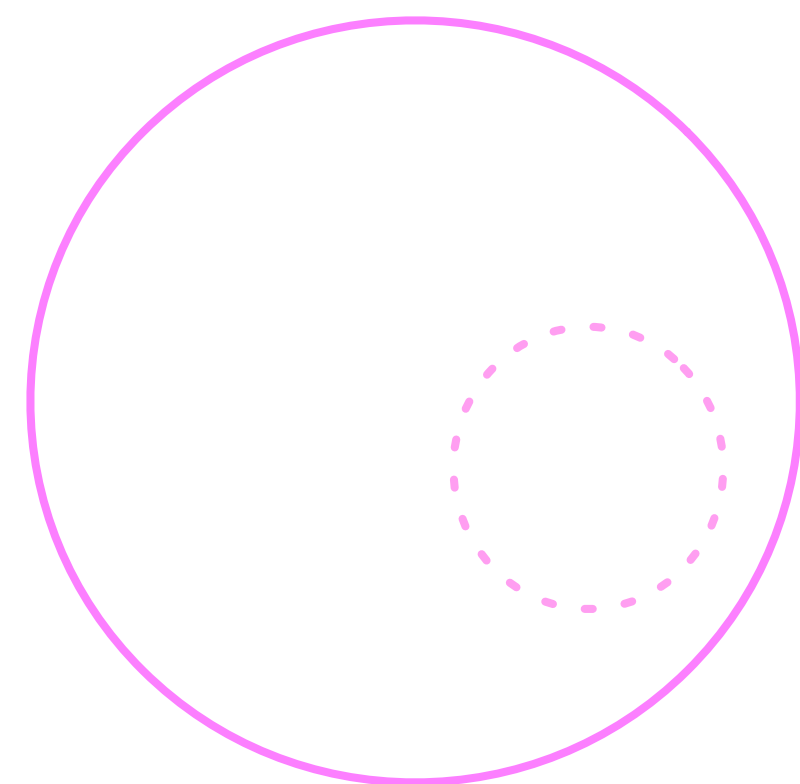
- 后果-各种挂站
  - 非核心服务挂了（上线/依赖）
  - 上线带上了别人的代码
  - etc.....





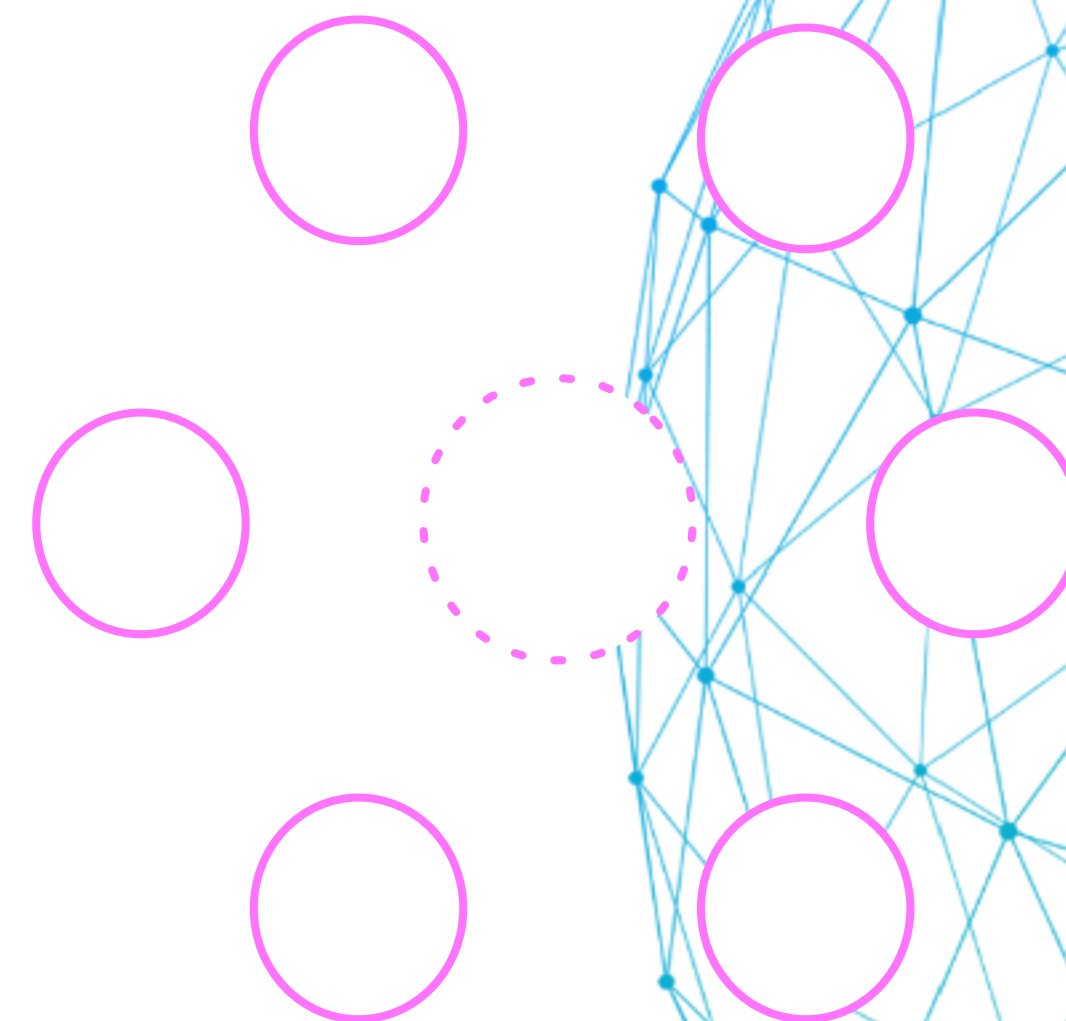
# 传统思路

- 架构拆分
- 非常必要，但是：
  - 周期长，见效慢
  - 目标不聚焦



大脑容量有限

无法关注



关注自己的业务

做精做透



# 稳定性评估

- 传统思路

- 200比例 or (1-5xx 比例)

- 错误log比例

- 各种监控图表

- 问题

- 系统视角而非用户视角

- 目标不聚焦

# 稳定性评估

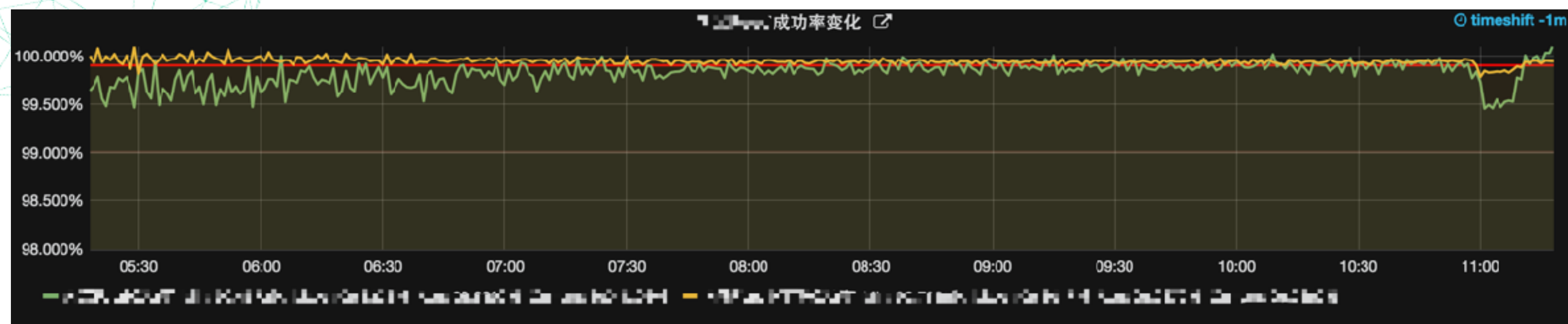
- 思路

- 聚焦于对于用户的核心价值
- 客户端评估
  - 解析成功量 / 总流量
- 服务端评估
  - 业务打点请求成功量 / 总流量



# 稳定性评估

- 实时稳定性图表



- 双月稳定性汇总

## 双月稳定性

双月	产品线 / 服务				
	稳定性	稳定性	稳定性	稳定性	稳定性
	稳定性: 99.98%	稳定性: 99.97%	稳定性: 99.99%	稳定性: 99.90%	稳定性: 99.97%
	不可用时间: 26.53s	不可用时间: 22.64s	不可用时间: 8.82s	不可用时间: 84.89s	不可用时间: 27.43s
	稳定性: 99.96%	稳定性: 99.96%	稳定性: 99.98%	稳定性: 99.84%	稳定性: 99.97%
	不可用时间: 1,443.56s	不可用时间: 1,421.78s	不可用时间: 736.16s	不可用时间: 6,030.94s	不可用时间: 1,302.41s



# 定目标

- **ROI**
- 不是越高越好
- **权衡点**
  - 用户体验极点的服务水平是如何的
  - 与收入的关系如何
  - 竞品的服务水平如何



# 数据分析

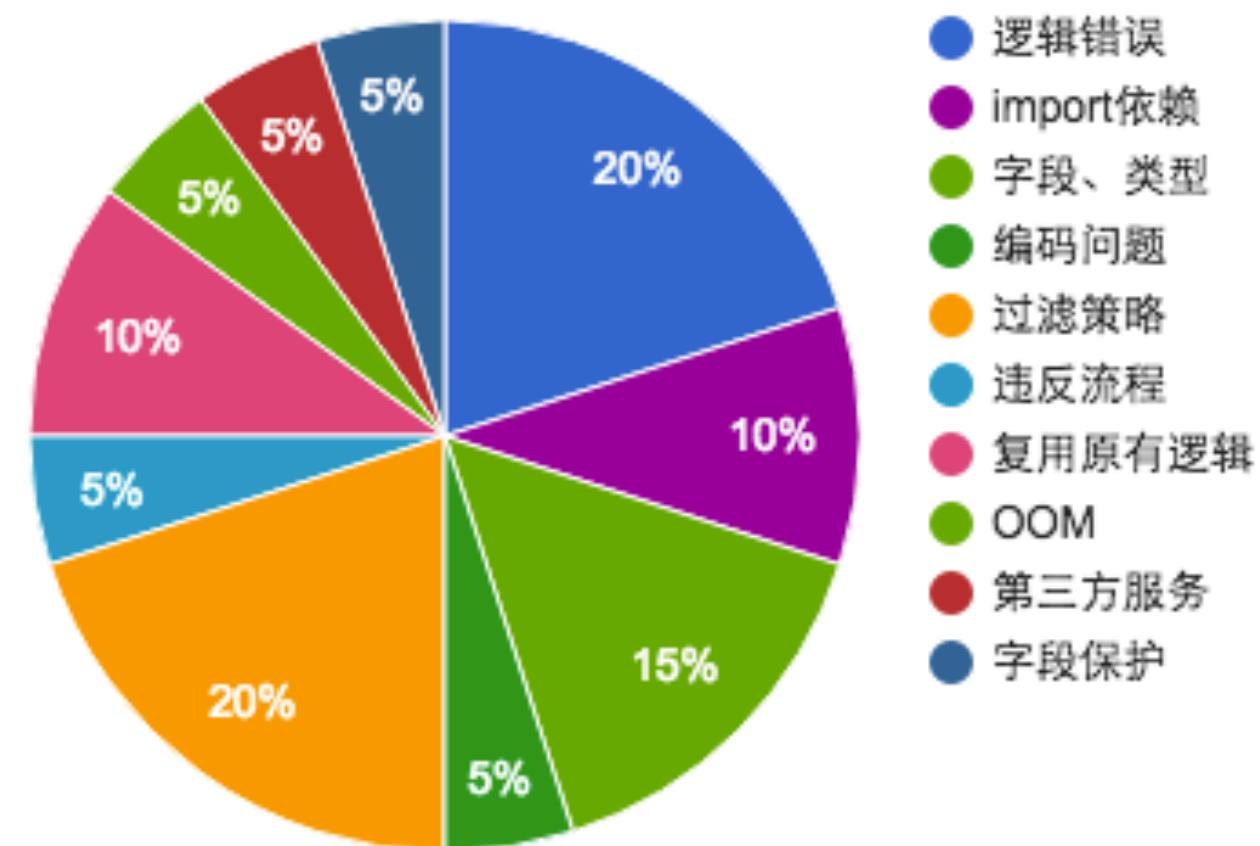
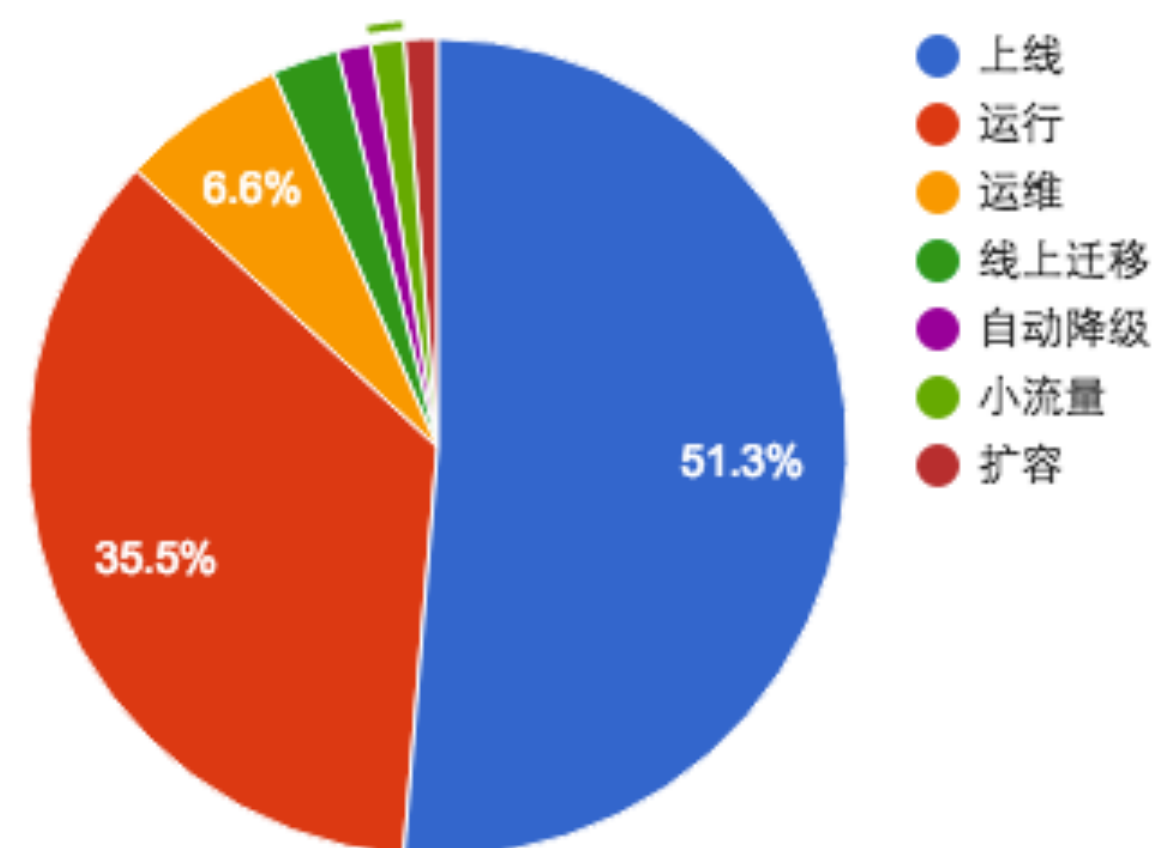
## 产生阶段

- 50%来自于代码变更
- 35%来自于依赖服务不稳定

## 变更细分

- 80%的错误可以事先避免

产生阶段分类汇总





# 方向划分

发布前稳定性 + 发布后稳定性



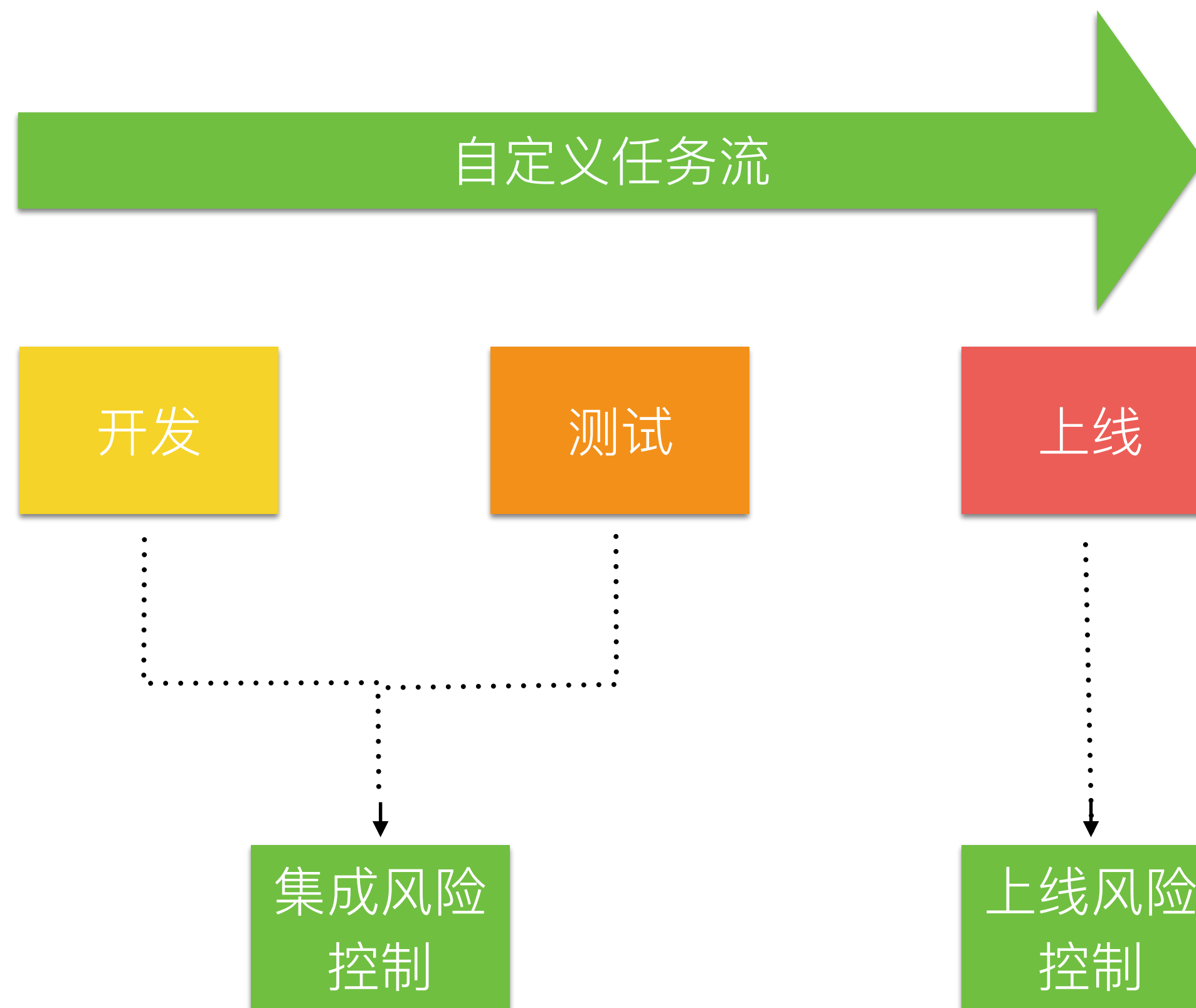
# 发布前稳定性

- 集成风险控制

- cd 平台

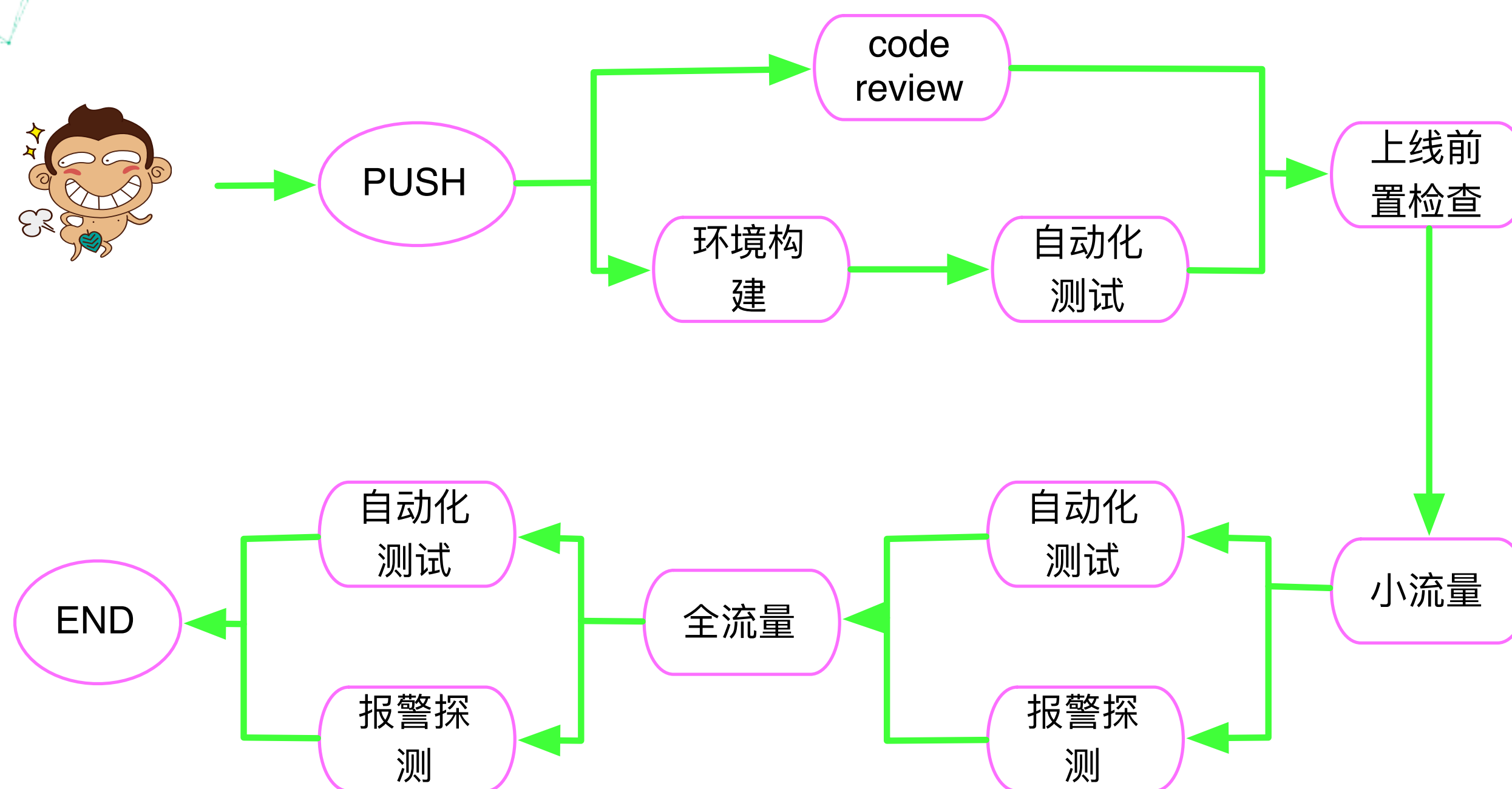
- 上线风险控制

- noops



# 发布前稳定性

- 绿色发布





# 发布前稳定性

- cd 平台

The screenshot displays a CI/CD platform interface with a list of recent builds and a detailed view of a specific build.

**Build History:**

- Build 1:** Status: Success (green bar). Stages: build (green), test (blue). Triggered by: P master. Time: 4 minutes ago.
- Build 2:** Status: Success (green bar). Stages: build (green), test (green). Triggered by: P master. Time: 5 minutes ago.
- Build 3:** Status: Success (green bar). Stages: build (green), test (green). Triggered by: P master. Time: 10 minutes ago.
- Build 4:** Status: Failure (yellow bar). Stages: build (green), test (yellow with error icon). Triggered by: P master. Time: 15 minutes ago.

**Build Details (CR:197178/2 add test stage Change-Id: I34c5...):**

- Commit ID:** 181ad15d
- 提交人:** wangling
- 耗时:** 分钟
- 开始时间:** 2016-12-14 16:15:21
- Build ID:** #1403
- Buttons:** Restart build

**Build Progress:**

- 构建依赖和打包 (Building dependencies and packaging) - In progress (blue bar)
- 测试 (Test) - Pending (grey bar)
- 部署小流量 (Deploy small traffic) - Pending (grey bar)
- 部署线上可观测 (Deploy to production for observability) - Pending (grey bar)

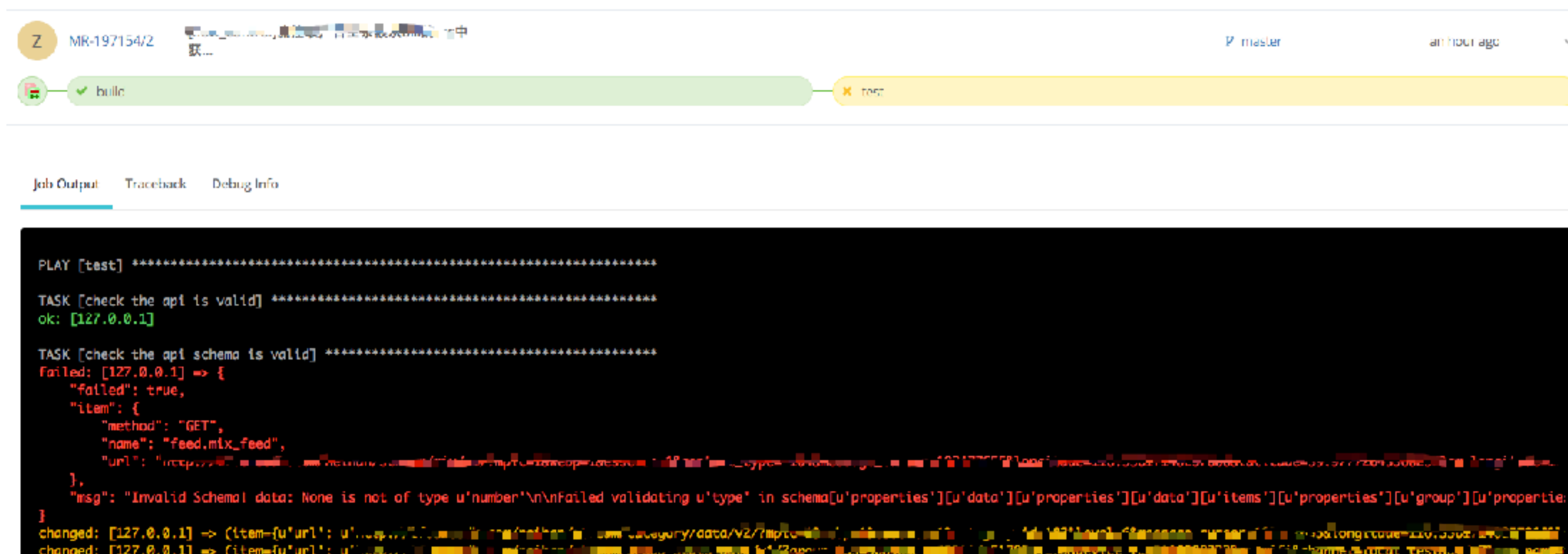
**Job Output:**

```
PLAY [构建依赖和打包] *****
TASK [install the npm dependencies] *****
```



# 发布前稳定性

- cd 平台



The screenshot shows a CI/CD pipeline interface. At the top, there's a header with a user icon, the username 'MK-197154/2', and a dropdown menu. Below this, a progress bar shows a green segment for 'build' and a yellow segment for 'test'. The 'test' segment is currently active. Below the progress bar, there are tabs for 'Job Output', 'Traceback', and 'Debug Info'. The 'Job Output' tab is selected, displaying a terminal log. The log shows a 'PLAY [test]' section followed by two 'TASK' sections. The first task, 'check the api is valid', passed with 'ok: [127.0.0.1]'. The second task, 'check the api schema is valid', failed. The failure message is: 'Invalid Schema! data: None is not of type u'number'\n\nfailed validating u'type' in schema[u'properties'][u'data'][u'properties'][u'data'][u'items'][u'properties'][u'group'][u'property:'. The log also shows some debug output for the failed task.

```
PLAY [test] *****

TASK [check the api is valid] *****
ok: [127.0.0.1]

TASK [check the api schema is valid] *****
failed: [127.0.0.1] => {
  "failed": true,
  "item": {
    "method": "GET",
    "name": "feed.mix_feed",
    "url": "http://127.0.0.1:8080/feed/mix_feed"
  },
  "msg": "Invalid Schema! data: None is not of type u'number'\n\nfailed validating u'type' in schema[u'properties'][u'data'][u'properties'][u'data'][u'items'][u'properties'][u'group'][u'property:"
changed: [127.0.0.1] => (item={u'url': u'http://127.0.0.1:8080/feed/mix_feed', u'method': u'GET', u'name': u'feed.mix_feed', u'category': u'data/V2/7mptc-0', u'longitude': u'116.5502', u'latitude': u'39.9062'})
changed: [127.0.0.1] => (item={u'url': u'http://127.0.0.1:8080/feed/mix_feed', u'method': u'GET', u'name': u'feed.mix_feed', u'category': u'data/V2/7mptc-0', u'longitude': u'116.5502', u'latitude': u'39.9062'})
```



# 发布前稳定性

- testing 平台

PSM - web.api

Case名称 *	<input type="text" value="user/profile"/>
Case描述	<input type="text" value="user/profile"/>
Diff_type *	<input type="text" value="json_validate"/>
case的URL地址 *	<input type="text" value="http://..."/>
Method *	<input type="text" value="GET"/>
测试数据(JSON) *	<input type="text" value="null"/>
期望结果(Schema) *	<div><div><pre>"message" ], "type": "object", "properties": {   "message": {</pre></div><div>自动生成</div></div>



# 发布前稳定性

- 自定义任务流

```
name: build # 任务名称
infra: python # 使用兼容标准的线上部署工具
stage: build # 构建步骤

tasks:
- name: update project dependency (only available when using python)
  git: dest={{item.location|quote}} repo={{item.scm_url|quote}} version={{item.scm_revision|quote}} force=y
  with_items: "{{ deps }}"
- name: load the service
  shell: |
    cd /opt/tango/site && ./deploy/service deploy
- name: check the neihan api service is alive
  systemd: user={{ TANGO_NON_ROOT }} name={{ service_name }} state=restarted
  register: result
  failed_when: "result.status.ActiveState != 'active' or result.status.SubState != 'running'"

name: test
stage: test
gather_facts: no
tasks:
- name: check the api is valid
  uri:
    url: http://{{ host_ip }}:{{ port }}/api/v1/profile?uid={{ uid }}&os_ver={{ os_ver }}&api=10.0.3&api_name=
    method: GET
    status_code: 200
    return_content: yes
  failed_when: "http result.json.message != 'success'"
```



# 发布前稳定性

## • 效果

- 月均拦住有问题的上线**上百次**

## • 问题

- 项目逐步完善, case 还不全
- 目标: 将因发布导致的风险降低到整体事故10%以内

# 发布后稳定性

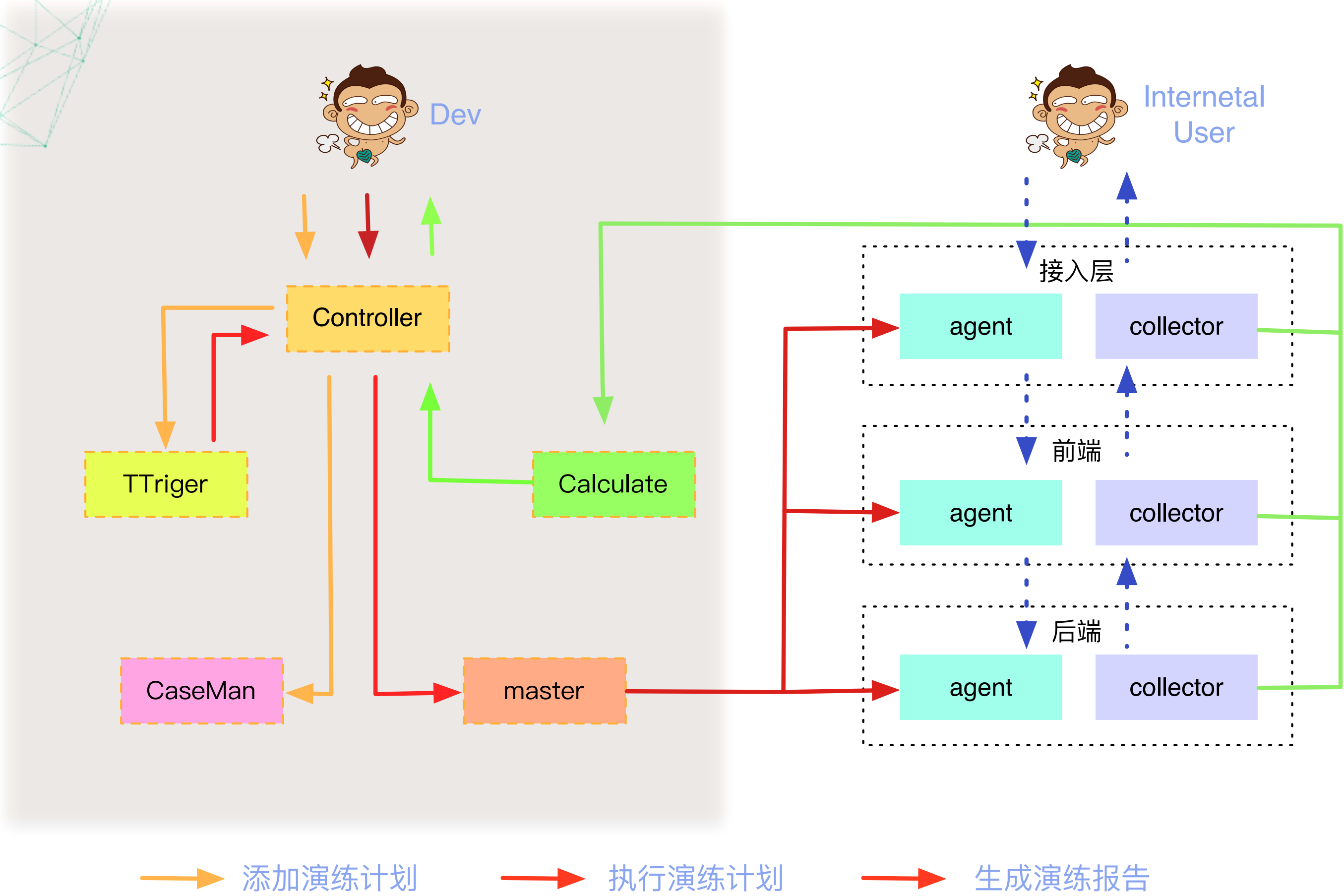
- 解除不必要的依赖
- 垂直业务逻辑/部署独立
- 非核心依赖自动降级
- 问题
- 思路
- 自动化演练



# 自动演练平台

- 背后的哲学
  - 拥抱风险
  - 风险无限，认知有限
  - 你不是系统
- 目的
  - 确保系统符合预知
  - 寻找不符合预期的点

# 自动演练平台





# 自动演练平台

- 演练类型
  - 服务不可用
  - 服务耗时增长
- 生产环境演练
- 周级别演练
- 单指标不合格自动中断
- 演练完成生成健康报告



# 自动演练平台

内涵Feed稳定性演练报告			
演练项	预期	结果	说明
Cont服务不可用	Feed正常返回	通过	
	服务成功率保持稳定	基本通过	延迟增加到 7.5 ms
	存在监控报警	通过	
MB服务不可用	Feed正常返回	-	服务本身处于大量失败状态，未进行测试
	服务成功率保持稳定	-	
	存在监控报警	-	暂无
Service 服务不可用	Feed正常返回	通过	
	服务成功率保持稳定	失败	延迟增加到 1.5 ms，稳定性下降
	存在监控报警	-	暂无，通过延迟报警可发现
Service 服务不可用	Feed正常返回	通过	
	服务成功率保持稳定	通过	
	存在监控报警	-	暂无
Service 服务不可用	Feed正常返回	-	流量太少，不确定状态
	服务成功率保持稳定	-	流量太少，不确定状态
	存在监控报警	-	
Service 服务不可用	Feed正常返回	通过	
	服务成功率保持稳定	失败	延迟增加到 1.5 ms，稳定性下降
	存在监控报警	-	暂无，通过延迟报警可发现
impression 服务不可用	Feed正常返回	通过	消重效果变差
	服务成功率保持稳定	失败	延迟增加到 1.5 ms，稳定性下降
	存在监控报警	-	
Service 不可用	Feed正常返回	-	流量太少，不确定状态
	服务成功率保持稳定	-	流量太少，不确定状态
	存在监控报警	-	
g_stats db 不可用	Feed正常返回	-	流量太少，不确定状态
	服务成功率保持稳定	-	流量太少，不确定状态
	存在监控报警	-	
w_baidu_memc 不可用	Feed正常返回	失败	返回异常结果，客户端不断重试
	服务成功率保持稳定	失败	未能发现这个错误，稳定性监控缺失
	存在监控报警	-	暂无
	Feed正常返回	通过	





# 文化机制

- oncall
- case study
  - 向事故学习
- 运维手册
- 最佳工程实践
- 有意识提升质量底线
- 持续强调安全意识

GIAC | BEIJING  
Dec.12.16-17



thegiac.com

# 稳定性体系

评估 (user-facing)

演练体系

容灾演练

性能演练

绿色发布

开发

测试

上线

容灾体系

流量调度

降级

过载保护

良好的架构

文化机制

oncall

case  
study

best  
practice





谢谢观赏

对架构或直播、视频感兴趣  
欢迎加入我们：  
[caojiandong@bytedance.com](mailto:caojiandong@bytedance.com)