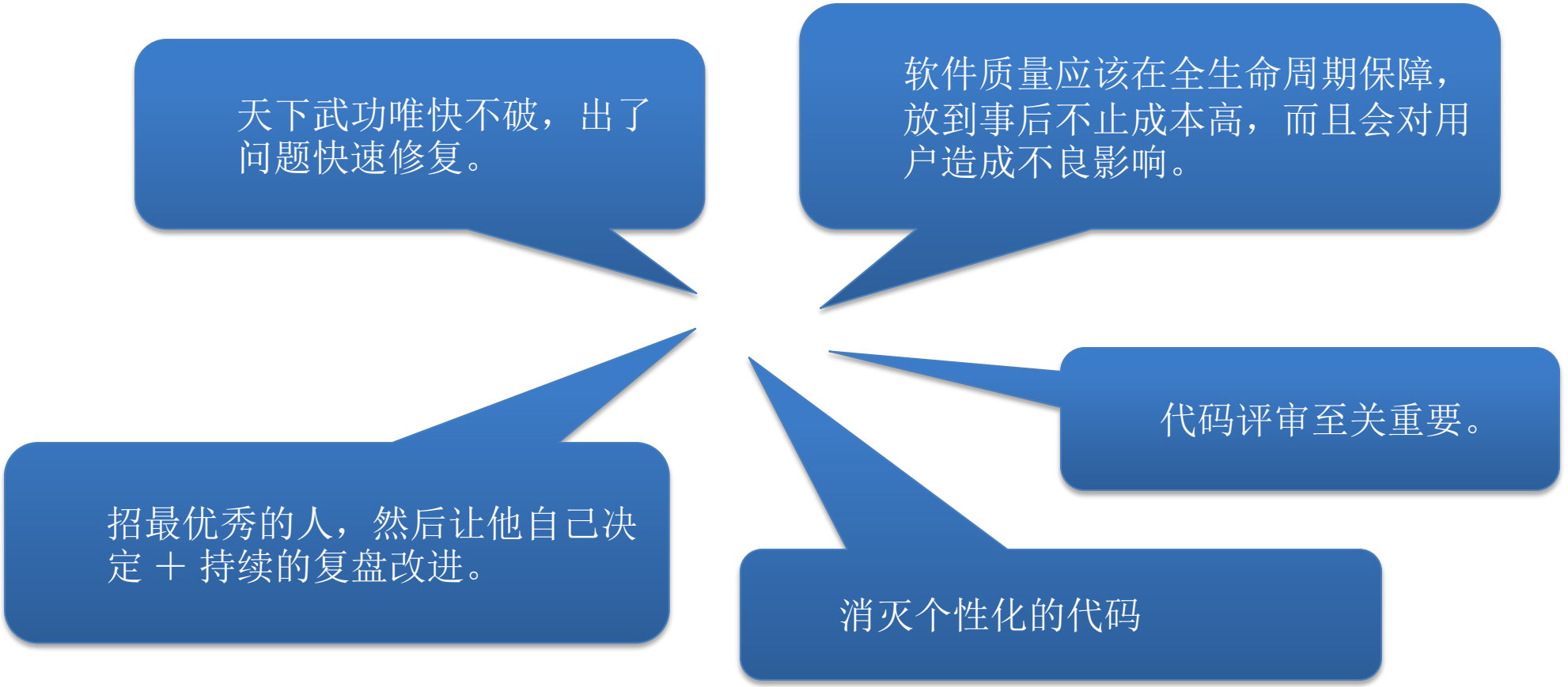
质量蜕变：STEP 0&1

讲师：宋全程

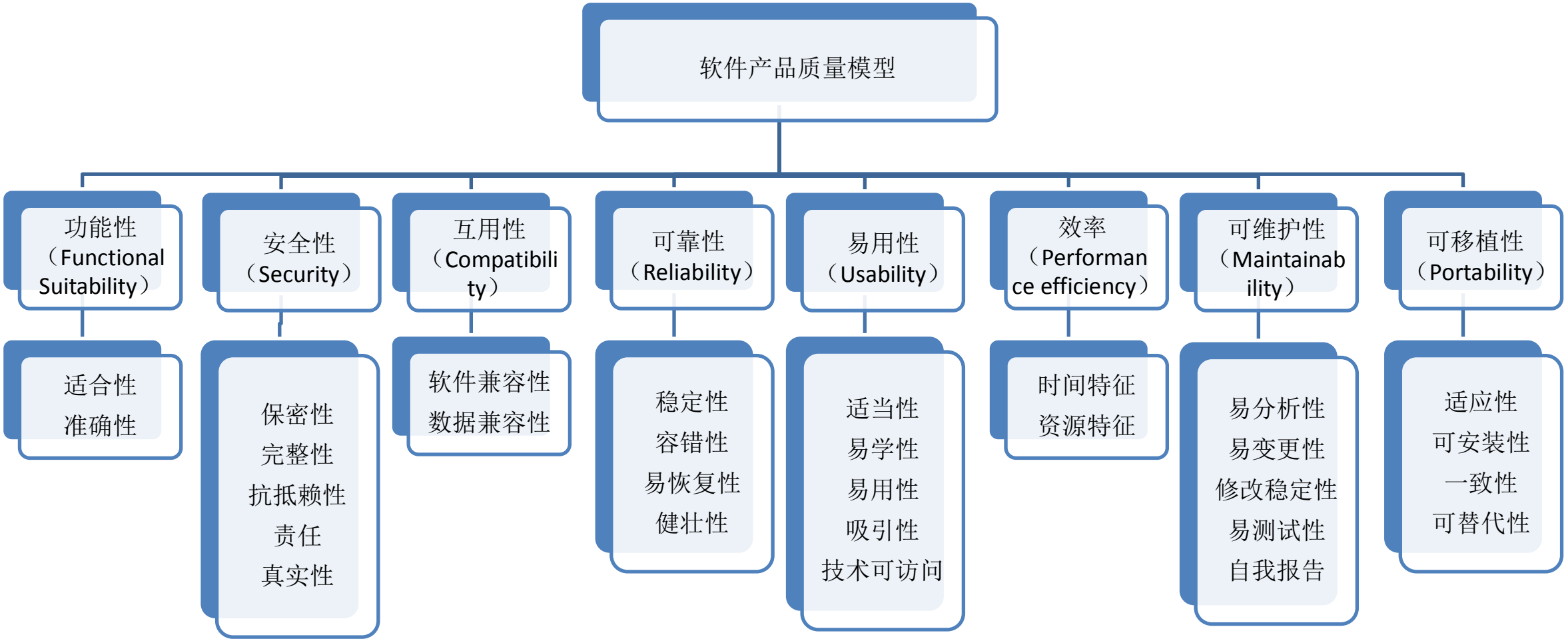
——“发展期的中型团队在控制软件质量时首先要考虑的那些事儿”

- 不同的声音


天下武功唯快不破，出了问题快速修复。
软件质量应该在全生命周期保障，放到事后不止成本高，而且会对用户造成不良影响。
代码评审至关重要。
招最优秀的人，然后让他自己决定 + 持续的复盘改进。
消灭个性化的代码

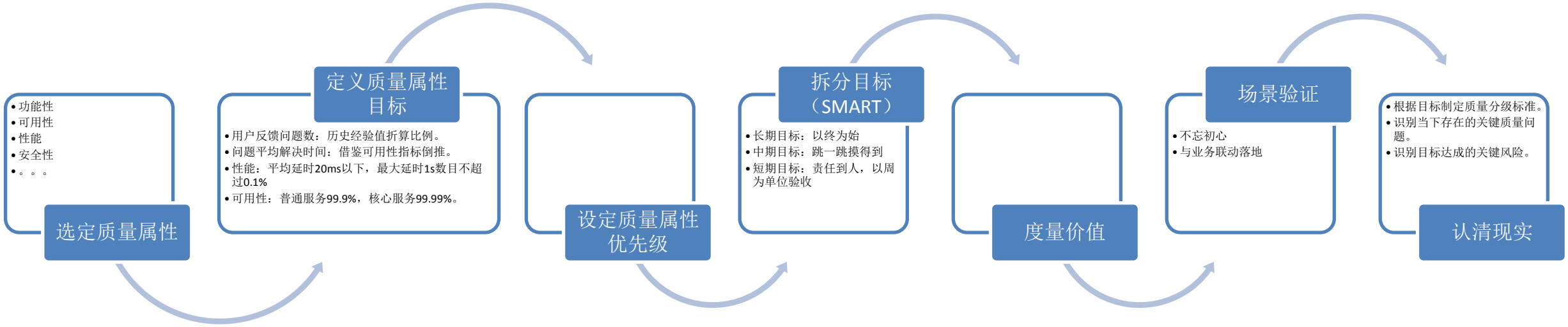

未雨绸缪
亡羊补牢
车同轨书同文
料敌机先
。。。

STEP 0：未雨绸缪

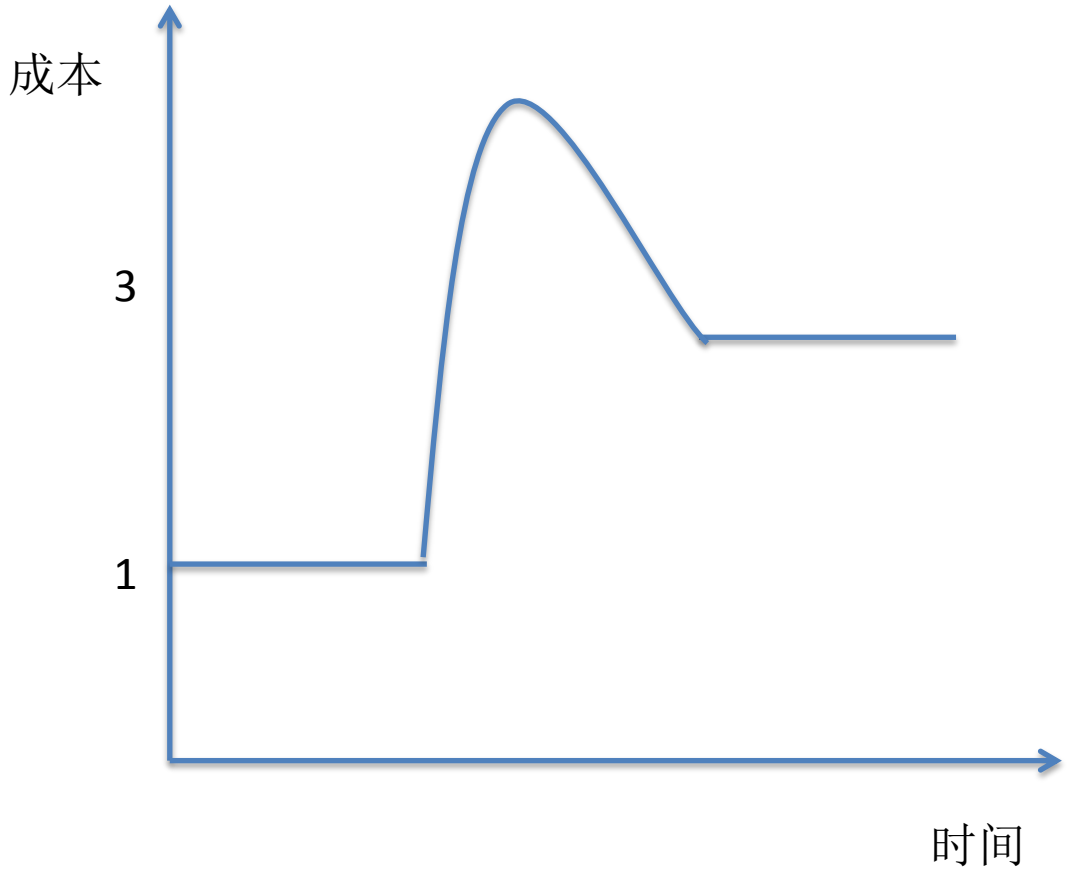

软件产品质量模型
功能性（Functional Suitability）
适合性
准确性
安全性（Security）
保密性
完整性
抗抵赖性
责任
真实性
互用性（Compatibility）
软件兼容性
数据兼容性
可靠性（Reliability）
稳定性
容错性
易恢复性
健壮性
易用性（Usability）
适当性
易学性
易用性
吸引性
技术可访问
效率（Performance efficiency）
时间特征
资源特征
可维护性（Maintainability）
易分析性
易变更性
修改稳定性
易测试性
自我报告
可移植性（Portability）
适应性
可安装性
一致性
可替代性


安全性
性能/效率
伸缩性
功能性
可靠性
。。。
可维护性
可扩展性
易用性

• 细化步骤


• 功能性
• 可用性
• 性能
• 安全性
• 。。。
选定质量属性
定义质量属性目标
• 用户反馈问题数：历史经验值折算比例。
• 问题平均解决时间：借鉴可用性指标倒推。
• 性能：平均延时20ms以下，最大延时1s数目不超过0.1%
• 可用性：普通服务99.9%，核心服务99.99%。
设定质量属性优先级
拆分目标（SMART）
• 长期目标：以终为始
• 中期目标：跳一跳摸得到
• 短期目标：责任到人，以周为单位验收
度量价值
场景验证
• 不忘初心
• 与业务联动落地
• 根据目标制定质量分级标准。
• 识别当下存在的关键质量问题。
• 识别目标达成的关键风险。
认清现实

- 成本曲线



- 跨越式发展？

1.不随意跟风，解决自己碰到的实际问题为主。不必追求屠龙术。
    1.技术要为企业创造价值，赋能需结合实际。

2.持续运营，培养业务团队的质量改进习惯并建立反馈循环。

3.他山石可攻玉，充分借鉴大公司的思路及开源工具。
    1.望远镜可以让人看得更远，但缩短不了你要走的路，他人的路，你不一定要遍历才算修成正果。

4. 一盘棋，避免烟囱式重复改进。

STEP 1：亡羊补牢

- 常见问题

1. 监控措施不完善，主动发现问题的效果不理想。
2. 用户反馈没人响应，或者没有责任人，或者被草率界定为“不予处理”。
3. 问题严重度定义太“善良”，较多致命及严重问题被胡乱降级，没有引起足够的重视。
4. 问题传播机制不完善，出了问题各级责任人或关联项目都不知情，没法从业务视角审视影响。
5. 问题处理拖拖拉拉，有的甚至一拖数月之久。
6. 很多问题需要跨团队协作排查，但甩锅现象严重。
7. 问题排查工具不完善，或者没有预埋点，很多线上问题无法复现或无从排查。
8. 问题无法根治，有些低级问题一而再再而三的发生。
9. 问题fix后发布周期太长，且缺乏应急恢复手段。
10. 问题总是扎堆出在少数模块上，但积重难返，牵一发动全身。
11. 部分人员意识淡薄，且破窗效应，有不良影响。
12. 与QA、技术部间协同解决问题效率低。

- 机制

• 绩效考核

•线上问题扣分原则：
- •问题分级
- •抓大放小
- •鼓励改进

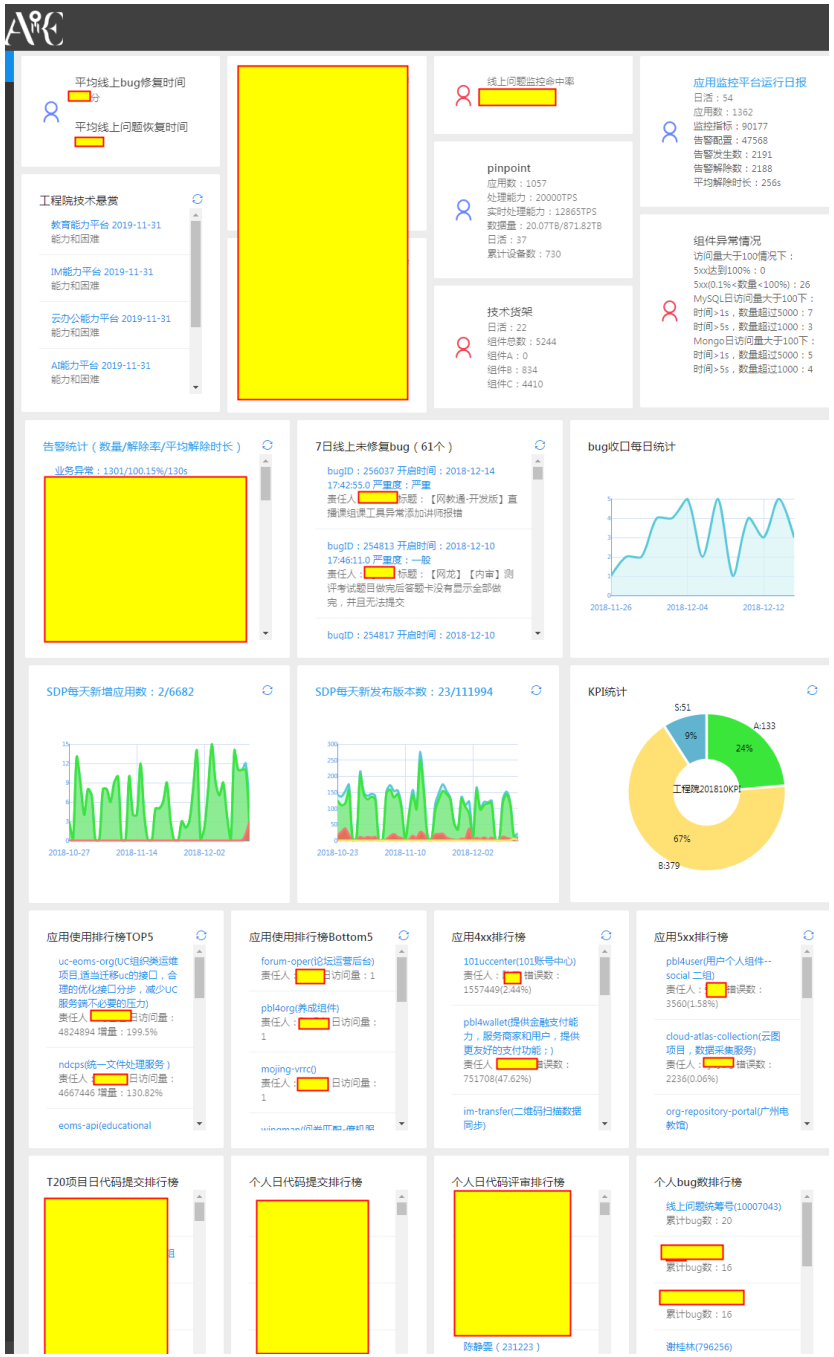
季度OKR

- 线上问题数减少比例
- 线上问题及时发现率
- 致命及严重问题平均恢复时间
- 线上问题平均修复时间

功能的严重程度(BUG性质)			+	概率		*	用户系数		*	处理时长系数			*	责任系数		*	态度系数	
等级	危害性	系数		出现概率	系数		用户量级	权重		服务端	客户端	权重		项目	权重		项目	权重
1-致命（非常严重）	**产品核心功能无法正常使用或异常。** **核心模功能：代表产品存在意义的功能** 1） 产品无法安装、升级、启动、登录，数据丢失、错乱或者损坏且无法恢复，造成系统崩溃、死机或者危及人身安全等的问题。 2） 产品核心功能无法使用或异常。	8		**必现：** 100%出现	2		1000w以上日活 或用户数≥1亿	2		48小时以上	96小时以上	2		唯一责任人	1		违规操作+指导方案未落地	3
2-严重	**产品主要功能异常。** **主要功能：QA一级测试用例覆盖到的功能** 1） 主要功能部分异常或与需求不符 2） 主要功能存在严重的性能问题、安全问题等	4		[5%, 100%)	1		[100w，1000w）日活 或用户数[1000万，1亿）	1.75		6-48小时	12-96小时	1.5		主要责任人	2/3		违规操作	2
3-一般	**次要功能未实现或存在缺陷** **次要功能：一级用例以外的功能** 1）次要功能部分异常或与需求不符 2）次要功能存在性能问题、安全问题等	2		[0，5%）	0		[10w，100w）日活 或用户数[100万，1000万）	1.5		1-6小时	2-12小时	1		次要责任人	1/3		指导方案未落地	1.5
4-轻微	**对功能无明显影响的问题** 界面或展示问题，如错别字、提示信息、格式不正确等，不影响用户理解和使用	0.5					[100，10w）日活 或用户数[1000，100万）	1		10分钟-1小时	20分钟-2小时	0.5		无责任人	0		正常操作	1
5-建议	**建议** 产品或案子的合理化建议 包括不影响实现的建议，用户友好界面，及较方便的操作方式。	0					100以下日活 或用户数[0,1000）	0.1		10分钟内	20分钟内	0.25					改进案得到嘉奖	0.5

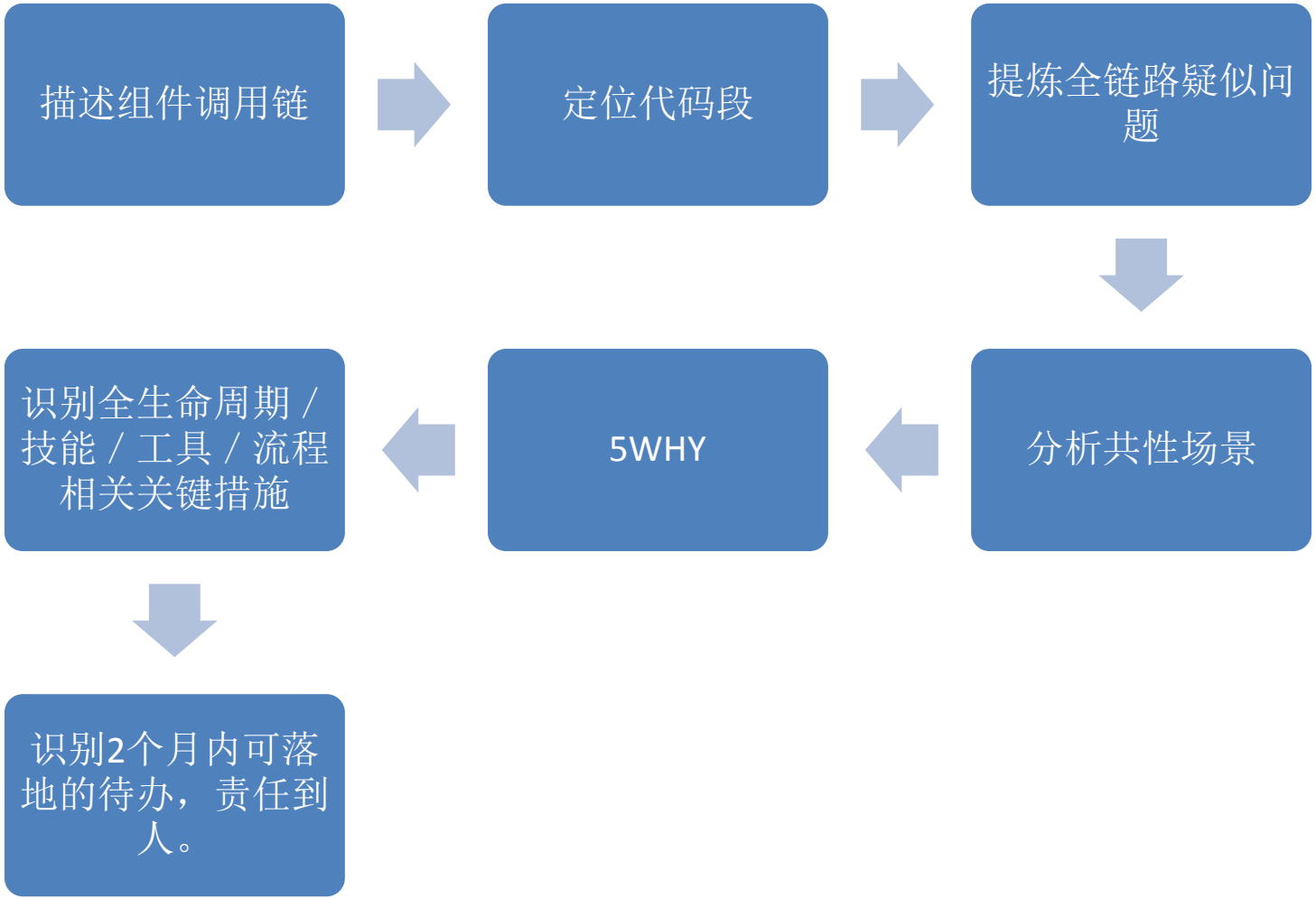
- 监督检查

质量看板： 公示在公共大屏幕。
监控系统： 监控指标：9w+， 告警配置：4.7w+， 每日告警数 2k+。
日报推送： 每日bug点名推送工程院大群 。
专项分析： 接口访问5xx、4xx，慢请求，数据库慢查询。。。
高危清单： QA每周定期给出待修复bug清单，并推进修复。
灭虫大赛： 清理遗留bug的运动。
每周一指： 每周出一份定向分析报告，并给出改进建议。
上架品控： 组件上架必须通过质量评估。



• 持续改进

根因分析：



说明：	该模板用于针对出现的线上问题进行根因分析，请严格按照表格进行填写，标*号的为必填项，请务必填写，否则报告直接被评定为不合格。如果再根因分析或解决措施或待办中有**同类问题的，请务必标出来**，认定于评委评估是否给与加分。				
BUGID *	*请填写该问题在PMS上面的ID*	异常等级 *		异常处理人 *	
责任人 *		主程 *		开发经理 *	
报告填写人 *				报告时间 *	
影响范围、危害及损失 *	*填写异常影响产品、用户量、端、功能点等信息（如，影响中物联官web端在线考试功能）* ***该项目的：使阅读人可大致了解本地故障带来的危害大小。如影响多少功能、影响多少用户、影响单端用户还是全部用户、给公司损失多少钱。***				
异常发生时间、地点/环境 *	发生时间：	*实际故障的发生时间（不仅仅是用户上报异常的时间，可从日志上查询得来）*	发生地点/环境：	□IOS端 □安卓端 □Web端 □服务端 □其他	
	定位时间：	*定位到故障原因时间*			
	恢复时间：	*实际故障恢复时间（仅恢复线上故障的时间，不一定对问题根本解决，比例如重启某tomcat暂时恢复业务）*			
	解决时间：	*真正解决本次故障的根本原因的时间*			
	总时长：	*从发生到真正解决的总时间（5天6小时）*			
异常故障描述 *	*此处填写故障用户端表现，如异常页面表现、流程中断等现象、web首页白屏、移动端考试无法提交答案提示连接超时等* ***该项的目的是填写用户可见的现象级别描述；如果是底层服务可填写技术参数（如存储空间为0、web服务器cpu打满等）***				
异常故障分析与定位 *	*请按照分析和排查的时间顺序填写处理人在接受到故障后的排查、分析过程，按照实际所做工作填写，用于记录线上过程和后期改进检查过程，如* *1、是否可以直接复现问题，直接定位问题范围、问题点* *2、网络方面的初步排查（80、443是否可连通）是包是否严重）* *3、服务端日志的排查（响应时间是否过长、是否500响应）* *4、业务日志排查（操作日志、业务调用日志）* *5、其他排查（nginx errorlog、mysql慢查询、tomcat catalina日志、zabbix监控等）* ***该项目的：记录排查、定位、分析问题过程和关键时间点，对过程、方法的好坏进行识别，好的做法进行推广，不好的做法进行识别和改进。（可以反映出问题处理的效率，是否有使用了很方便的工具，是否有很好的技术经验可分享）***				
近因解决措施	*请填写恢复业务的措施（可能并没有解决根本问题）和重要时间点如）* *1、判定为任务依赖问题的，先重启应用恢复业务* *2、业务bug，紧急hotfix* *3、底层基础服务问题（硬件、软件、网络）通知相关团队协助解决*				

原因扫描分析

该项目的：挖掘和识别根本原因，避免类似问题再次发生

1、识别原因，描述清楚什么时间（what/who（责任人的岗位/角色）/when/where/how much）+ No why/how/judgement 的要求，尽量客观、量化描述现状与标准/理想状态的差距；

2、用5WHY追问可能的最深原因，进一步问着为底，确保所有原因都列出

3、用红色字体标出每个流程你认为真正重要的原因

生命周期	**技术** （如语言、框架、组figure）	**工具/资源/设备/网络**	**流程/机制** （制度/职责/组织设置）	**人** （行为/知识/技能）	**其他**
运维方面		*如，共享平台未提供线程快照抓取能力，无法在半个小时内排查到具体问题，导致处理超过2个小时*	*如，线上BUG响应，流程需要经过共享平台、技术部2层的沟通，延长了处理问题的时间*	*如，值班人员也值班时间内，手机设置为静音，不能第一时间处理问题*	
产品发布					
测试方面					
编码方面					
详细设计	*如，设计时缺少考虑参数校验*				
架构方面					
需求方面					

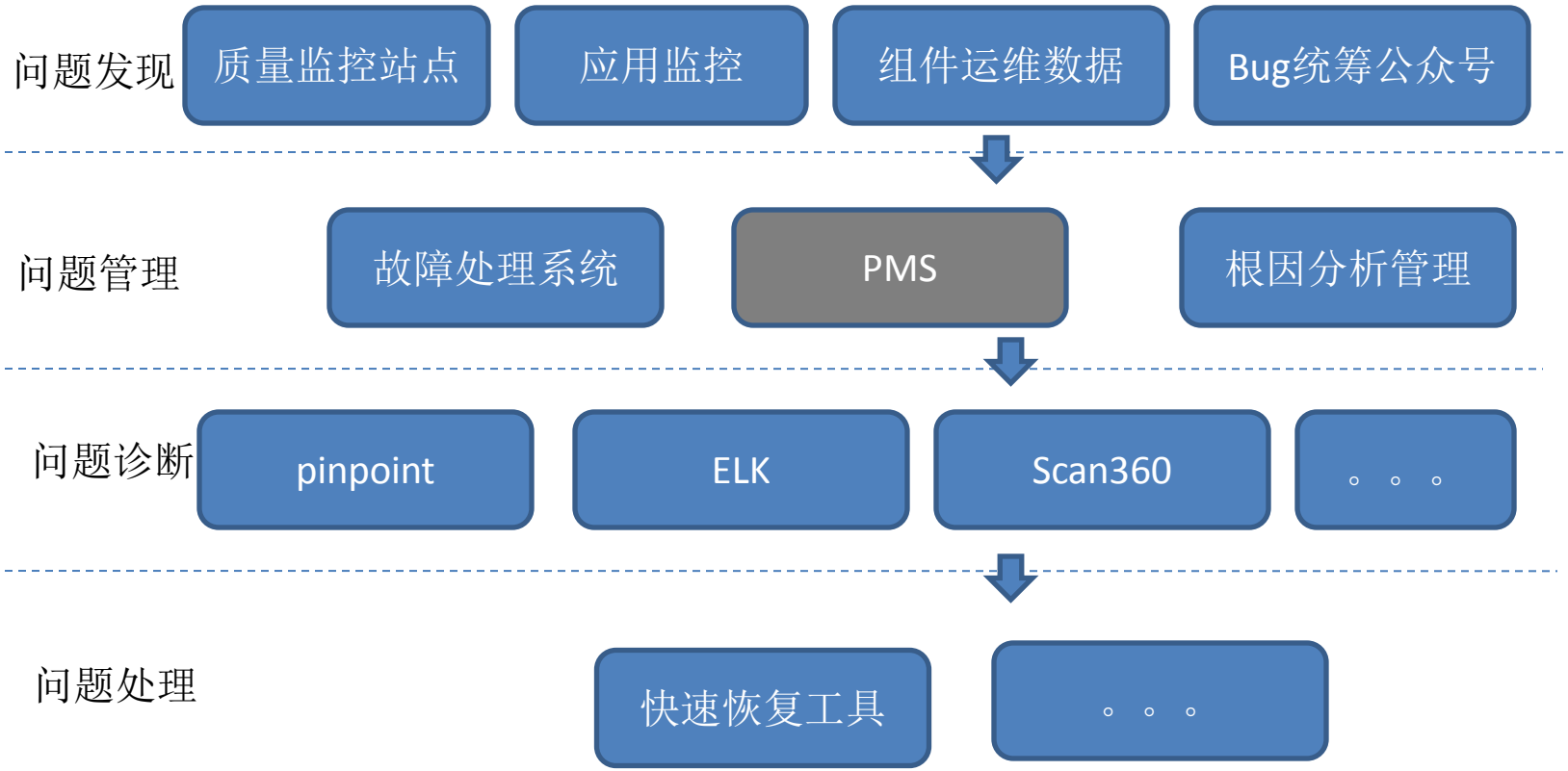
	根因总结 *	**待办清单 *** *（请思考：在保证效果的前提下，还业务方真的一定要做吗？有没有更简单有效的方法？）*	**相关负责人 ***	**截止时间 ***
根因分析 * *（请思考：如果这些根因真的得以解决，应该可以100%杜绝问题发生了吗？）*	*请按照严重程度的顺序对本次故障进行根本原因总结描述，通常在纵向上会有一个技术链条，可以自下而上的进行解释。* *应该至少有一条主线描述* *可能会涉及到很底层的技术细节，这个很有必要* *你遇到的问题，可能别人从未解决过，尊重尽可能详细的解释清楚，作为工程师的技术财富积累下来*	*用于跟进识别可落地的待办跟进，要跟主要根因和解决措施相对应，否则评审中可能会被扣分* *1、待办需要清晰具体（符合smart原则）* *2、待办尽量可验收* *3、待办需要有完成时间和可能的工作量、有明确执行人* *4、待办会定期检查验收* *5、技术评审专家可添加待办* *6、质量管控小组可以根据实际情况与被确负责人沟通后，增删待办*	*填写该项解决措施对应的负责人姓名*	*填写该项待办清单对应的完成截止时间，如超过2个月，请说明原因，如无说明视为无效*
附图	*附录必要的bug处理过程或其他信息截图（重要日志、重要监控信息、重要错误截图等）*			
其他备注	*用于填写本根因无法覆盖到的任何问题、建议，均可填写，不限主题、不限内容。*			

- 沟通协作

跨团队协作	外部门协作
• 首问责任制 • Bug日清制	• QM委员会 • devops

- 工具

• 工具


问题发现
质量监控站点
应用监控
组件运维数据
Bug统筹公众号
问题管理
故障处理系统
PMS
根因分析管理
问题诊断
pinpoint
ELK
Scan360
。 。 。
问题处理
快速恢复工具
。 。 。


DevOps工具箱
图标布局
搜索应用和组件
线上质量管控
应用监控(beta)
组件运维数据
质量监控站点(...
PinPoint
MAT内存分析...
HeapHero内...
Bugly
Scan360
故障处理系统
工程院工具
ELK统一查日...
splunk
服务治理
cube
五部一处运维...
质量管理WIKI
SDP基础服务...
QA工具
BUG统筹公众号
接口云测试
T20质量指数
线上云监控(ap...
禅道PMS
技术部发布及运维系统
Boss发布系统
服务器资产相...
技术部WIKI

• 问题发现

主动：端到端的监控

ios/android： 崩溃、卡顿、高流量、埋点。。。

web： PSI 、埋点

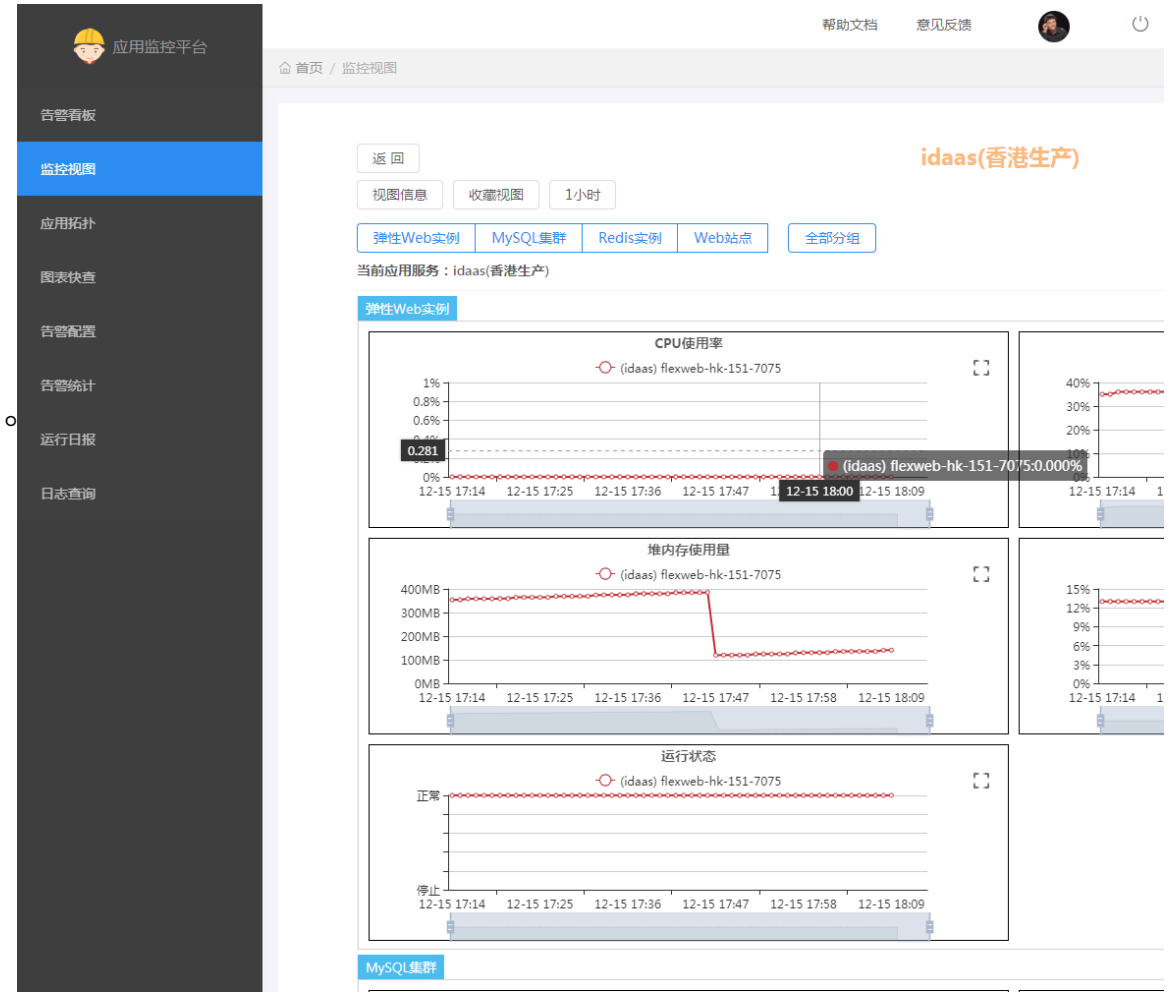
network： dns 、p2p联通性检测（开发中）。

服务： api拨测（w级）、埋点、api调用、数据库慢操作。

基础设施： 软硬件基础设施的监控项全覆盖。

被动：

bug归口：统筹公众号



- # 关于监控

数据总览:

最高接入应用数：1362	最高监控指标数：90402	最高告警配置数：47599	累计告警(发生/解除)数：3324/3323
累计告警解除率：99.97%	累计活跃用户数：54	平均每日告警(发生/解除)数：1662/1661.5	平均每日用户数：54

查看表格　查看图表

日期	接入应用数	监控指标数	告警配置数	告警数（发生/解除）	告警解除率	告警分类统计 业务/性能/资源/存活/其它	活跃用户数
2018-12-15	1362	90402	47599	1133/1135	100.18%	612/330/175/5/11	

监控指标：9w+， 告警配置：4.7w+， 每日告警数 2k+。
标准化的基础告警配置。
99U、短信、电话多路告警。
业务&可用性告警与普通告警分离。
日常工作：对监控难度切分SABCD五级，B级级以上要求做到全监控。
Aiops尝试：智能设置告警阈值、告警召回。

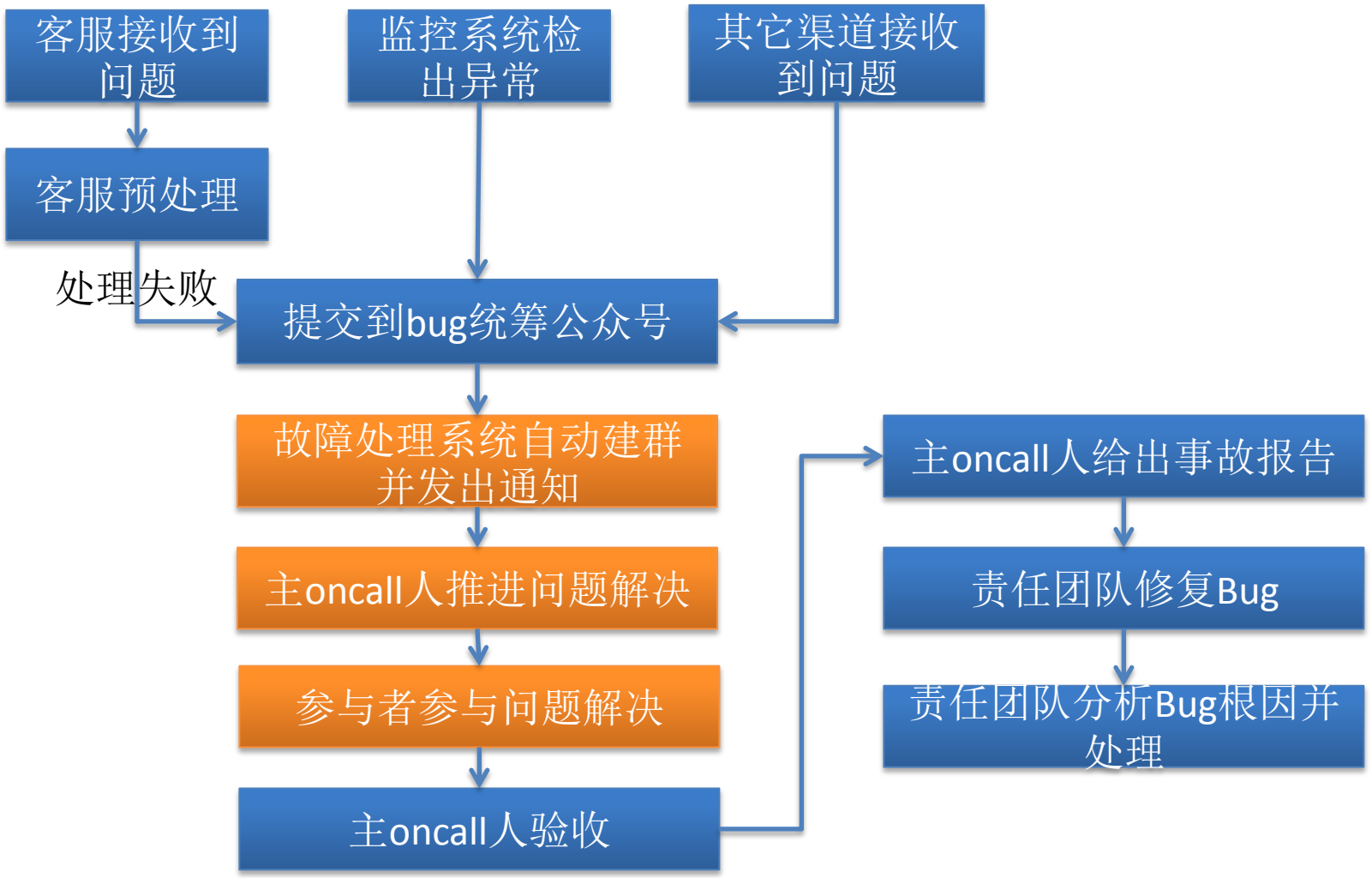
- 问题管理

应急处理 → Bug处理 → 根因管理

禅道PMS

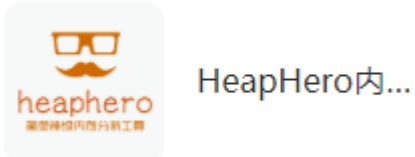
- 关于oncall

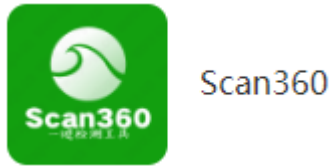
系统值守
超期升级
首问责任
自动寻人
动态调班，负载均衡。
处理期间不干扰，不追责。
故障报告。



• 问题诊断

Pinpoint：分布式追踪
ELK：业务日志分析，并与监控系统，pinpoint等打通。
DUMP分析：jvm dump可动态开关，移动端dump可定位到组件级。
scan360：一键检测工具，降低业务团队看监控视图的门槛。





• 问题处理

快速恢复措施：

版本快速回退

- 服务端一键发布旧版
- 客户端快速发布

动态配置

- 功能动态开关
- 动态参数配置

维护

- 重启
- 扩容
- 限流、降级
- 清缓存
- 清队列
- ...

谢谢