

个案排查能力建设

背景

SSC Android 目前在线上问题排查方面还是有一些问题

- 发现问题
 - NOC 监控：这块主要是监控接口成功率，和后端一样
 - 用户主动反馈，工单驱动，这部分可能会漏掉不少日常中的个案问题
- 排查问题：主要是通过看服务端日志，看客户端代码逻辑的方式去排查

从 SSC Android 过去的经验看，归属客户端问题为 0，大多 1~2 小时可解决，最多 0.5 天，这里有几个原因

- 用户本身容忍性较高，一些小问题就算了，不反馈了
- 目前记录在案的问题，100% 是服务端的问题，可以直接看服务端接口的数据异常就可以定位到

还有一些案例未记录在案的，且不能通过服务端接口日志定位到的

- 有定位时间长 0.5 天的、1 天的
- 也有些不了了之的，需要去联系用户去询问的（了解操作方式、网络之类的），这种在以前的工作经历中也是经常遇到的

从个人经验上看，客户端排查问题往往是最难的。一方面很难接触到用户，另一方面用户所处的网络环境很复杂，机型也比较多，而且用户的操作不一定是我们预期中的操作流程，靠看代码去定位问题还是不太靠谱，需要进一步提升问题排查的能力。

目标

- 提升问题定位效率：90% 问题可在 24 小时内定位到原因

思路

- 方式：主动 + 被动
- 手段：日志 + 业务关键信息监控

主动排查：重要业务场景业务监控，完善调用链信息和关键信息收集，主动上报，便于主动发现问题

被动排查：如果用户反馈问题，可提供用户 ID、手机号等，根据这些信息捞取用户操作记录等关键信息来进行分析

不管主动还是被动，客户端都需要记录好较为完整的程序运行过程和关键信息，尽量能够在服务端还原用户的实际场景

- 程序运行日志：[0.日志能力建设](#)
 - 业务调用过程
 - 异常分支
- 业务关键信息监控：[1.核心场景主动监控](#)
 - 出问题时候的网络诊断报告（如果是与请求服务端有关）
 - 具体业务场景的关键数据
 - 如扫码场景导致长时间识别不成功的原图片，可分析二维码是否模糊、褶皱、亮度不够等，便于后续的优化操作
 - 如 RN 白屏场景的截图、View Tree 的位置信息、父子关系等
 - ...

关键信息这里就是要具体看业务场景，任何有助于分析该场景问题的数据都可以。

子项目