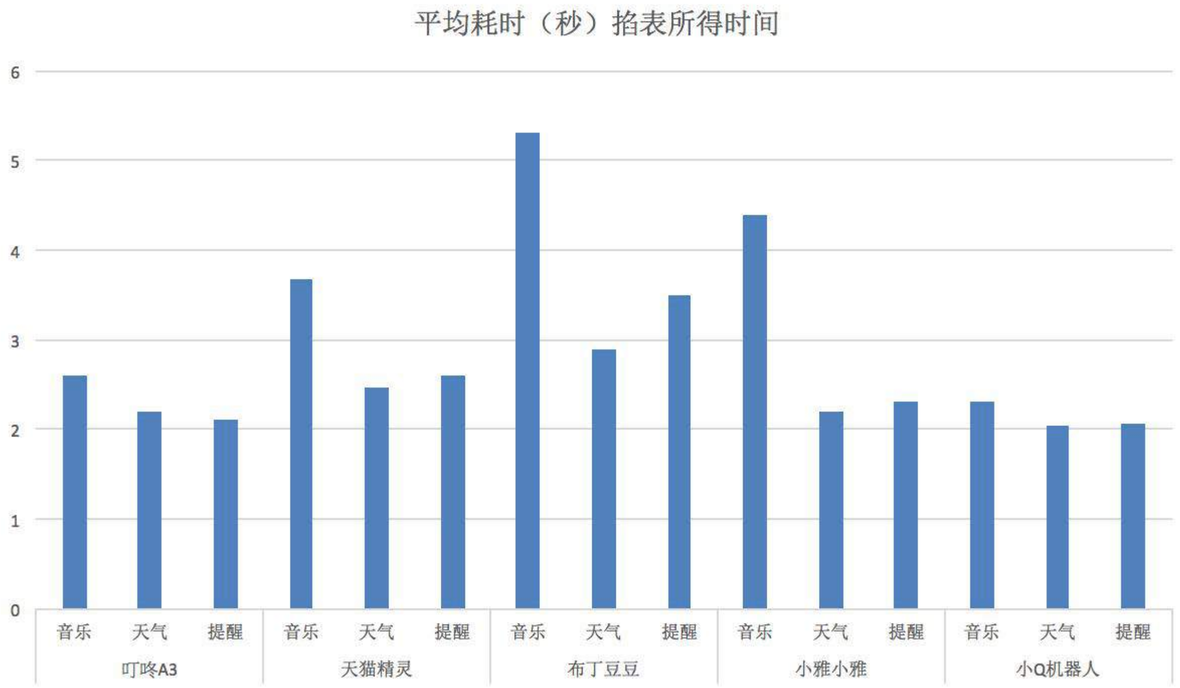
响应速度优化

优化之后，应答速度在主流硬件里处于领先地位，并且后续还有优化空间。



测试数据来源：测试同学

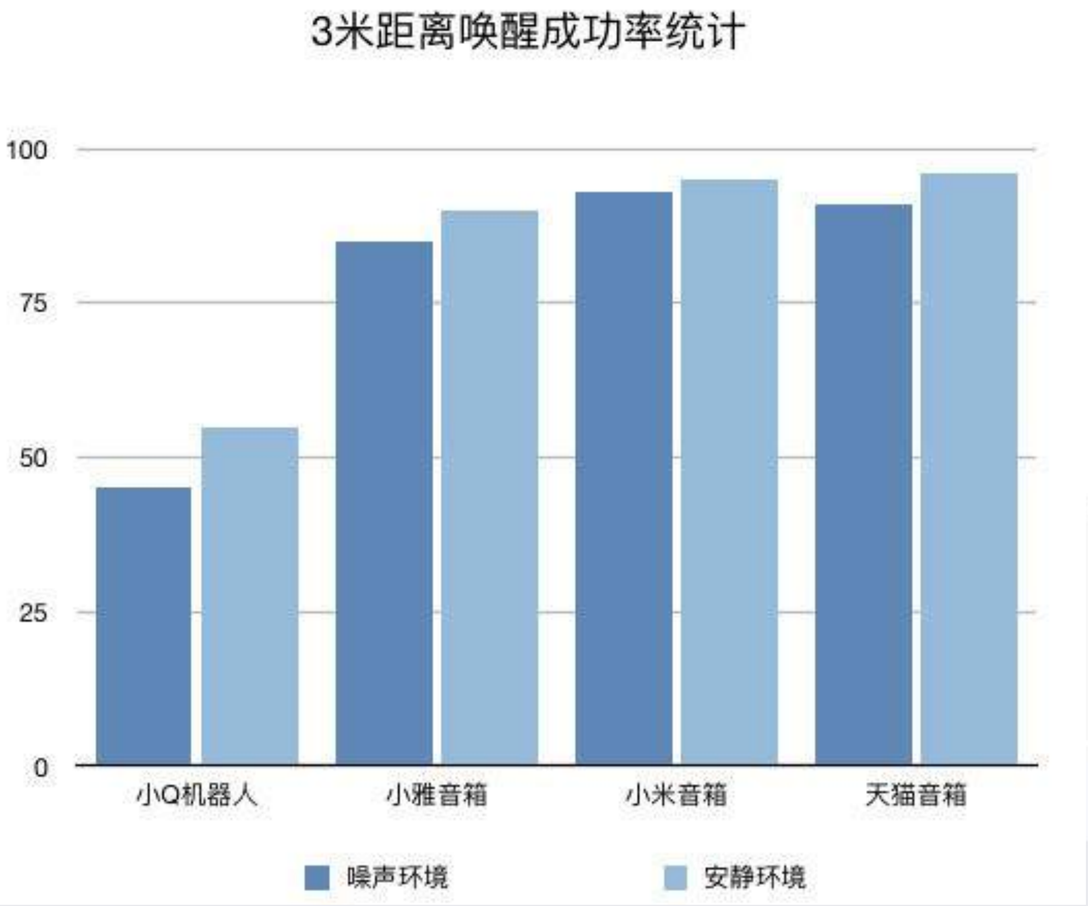
关键事件TOP

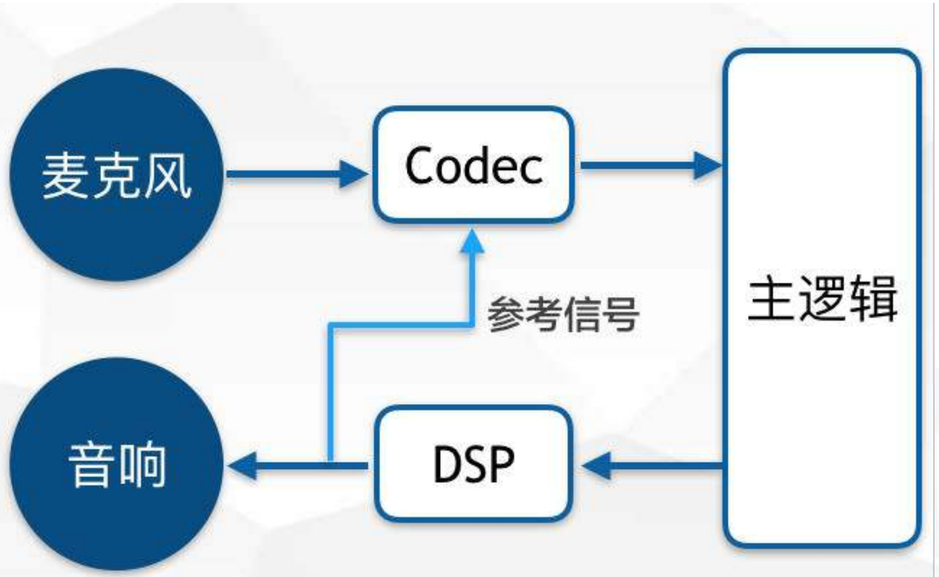
事件	成功率	事件量	耗时
client_OnRspBegin	95.401%		314.309
client_PlayerPreparedT...	100.000%		344.408
client_SilenceTTSDelay...	100.000%		1060.329
client_SilenceVoiceDel...	100.000%		1275.294
client_TTSRspDelayTime	100.000%		324.591
Server_Rsp_Client	100.000%		1.034

测试数据来源：link.oa.com

唤醒成功率低

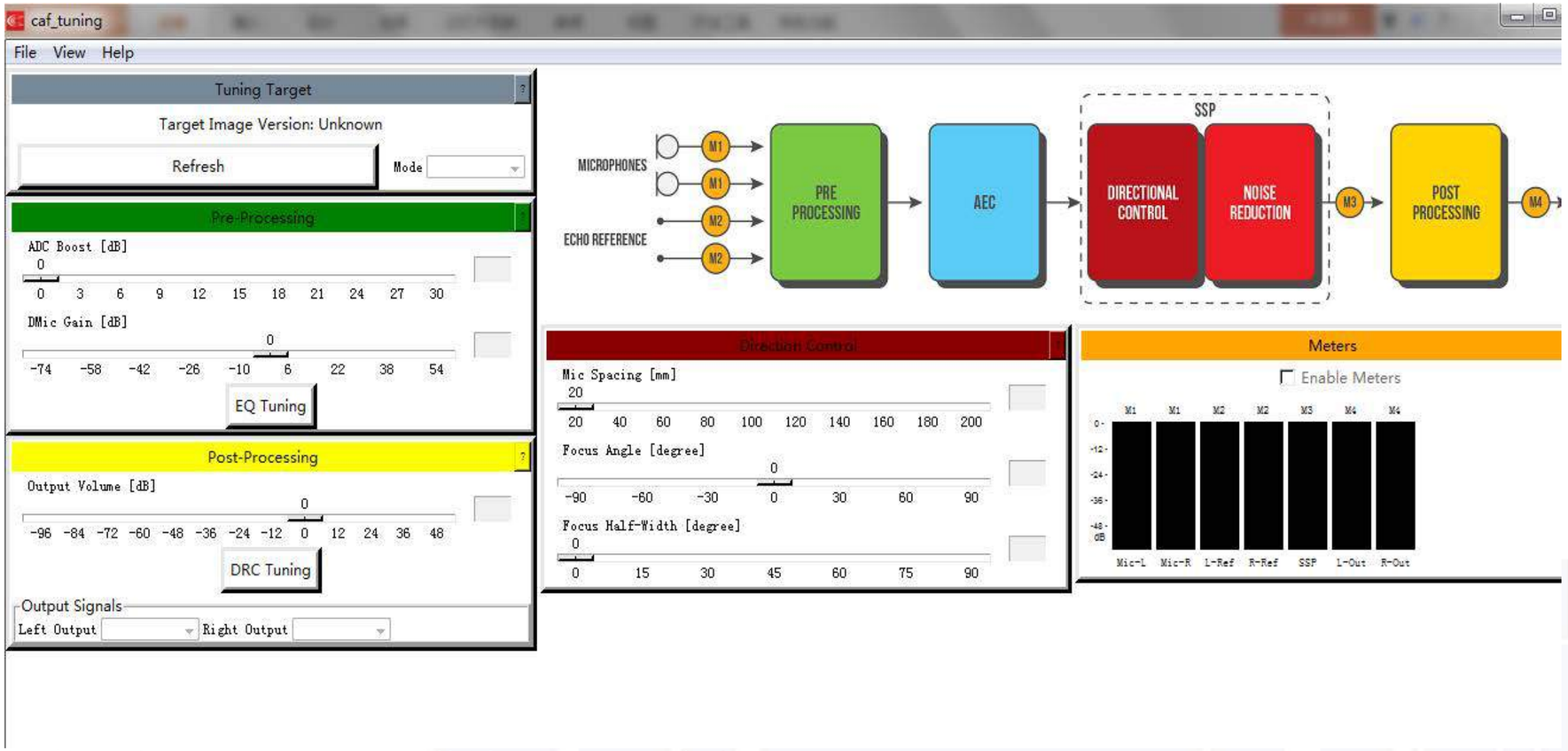
第一版本工程机器出来以后，遇到一个比较大的问题是唤醒成功率低，跟市面上的竞品做了对比测试：





唤醒成功率低

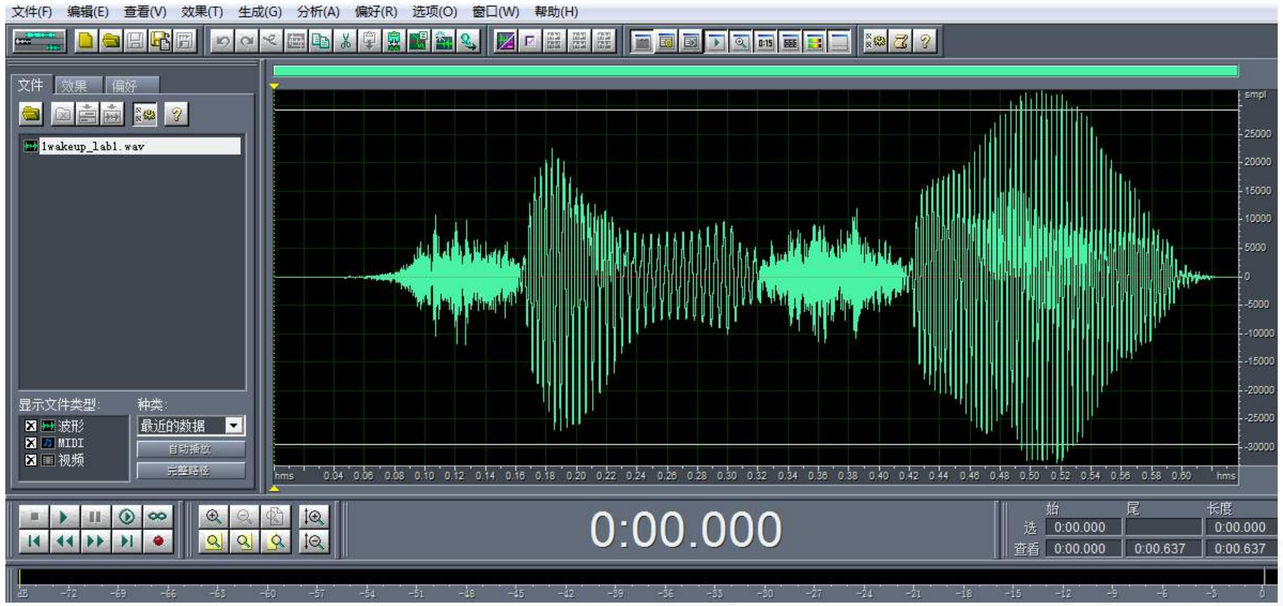
PC调试工具连接前端IC，dump工作时各寄存器的设置，确认IC配置是否正确。

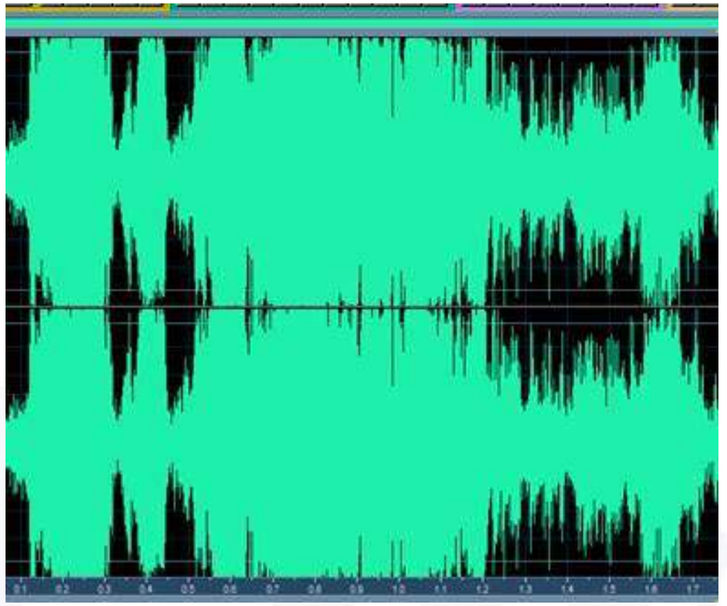


发现问题：codec模块AEC没有打开。

唤醒成功率低

前端adc、codec抓取原始音频pcm数据，分析前端信号完整性。



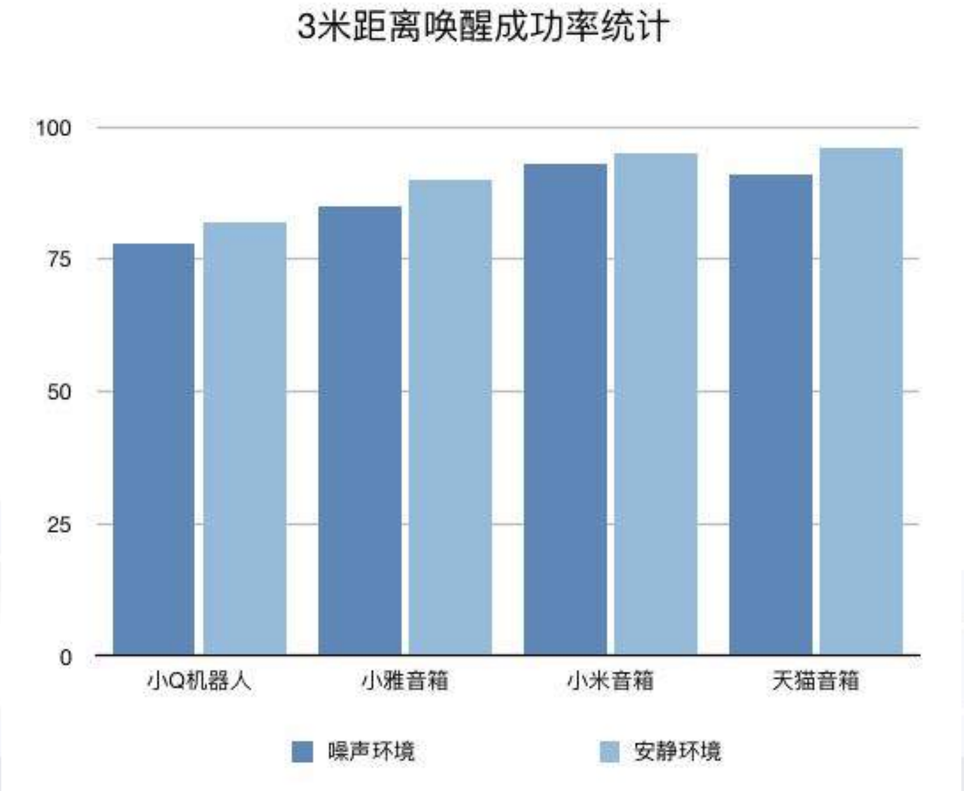


找到问题原因：

1. mic输入信号饱和，信号被截幅。
2. AEC参考信号饱和，信号被截幅。

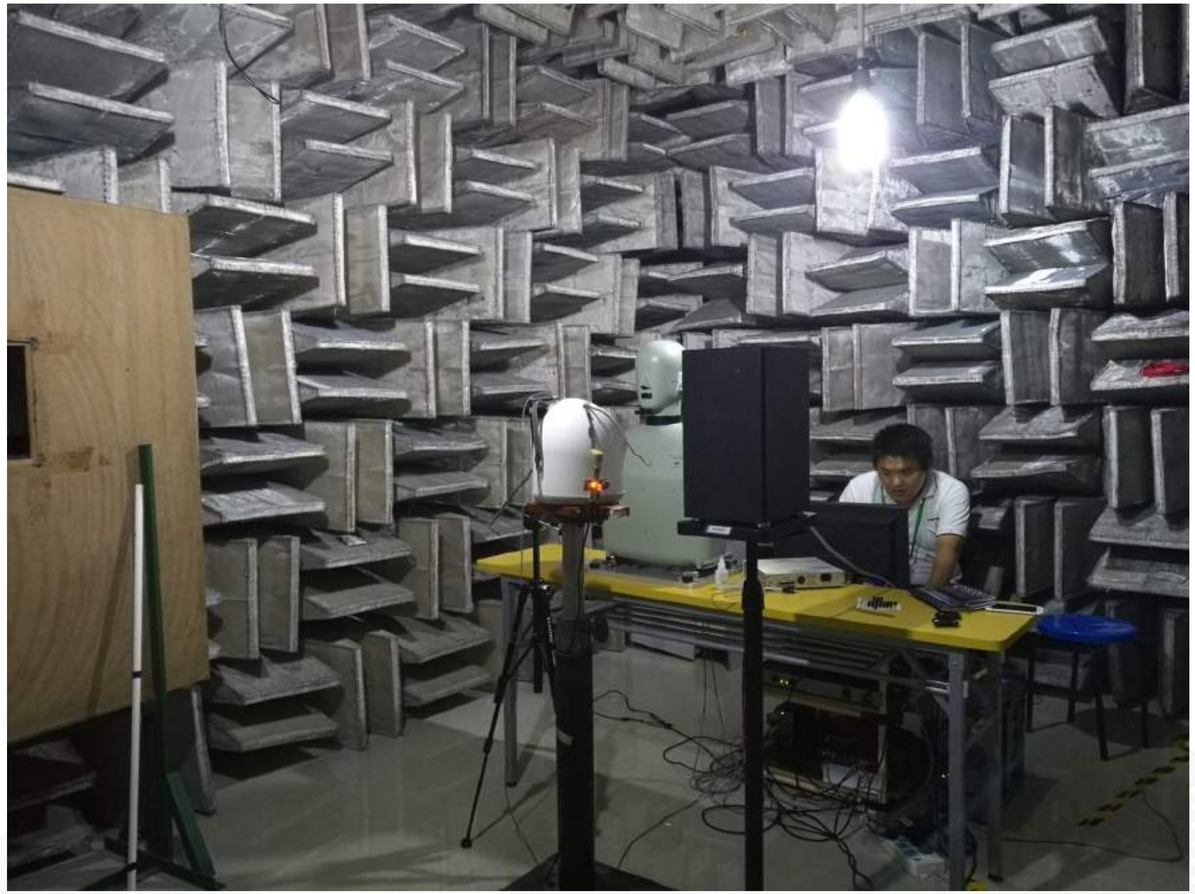
唤醒成功率低

针对前端信号幅度被截的问题，调整了codec的参数，开启硬件回声消除；
调整mic输入的放大增益参数以及调整AEC 信号的放大增益参数，确保信号幅度的完整性。
测试的成功率在3m安静环境测试，由原来的55%提升到82%，有明显的提升，但是跟竞品对比还有明显差距。



唤醒成功率低-优化

专业音频实验室测试，验证Mic频响和结构密封性。



唤醒成功率低

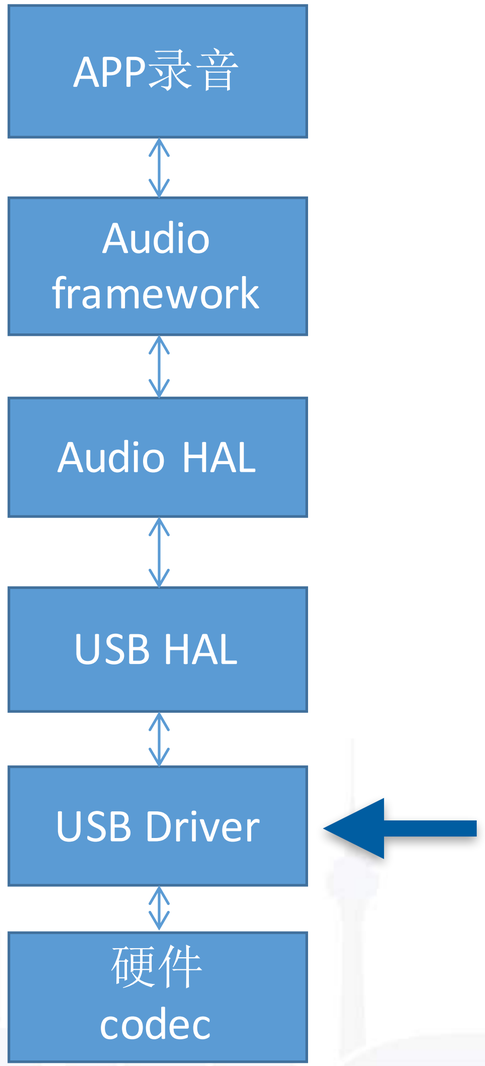
软件分析，应用层、Audio framework 、USB声卡驱动代码分析，USB眼图信号分析

问题原因：

USB driver中多核多线程并发情形下有一个变量变成了异常值，引起USB传输异常，导致APP录音一直获取不到数据卡住。

解决措施：

修改USB驱动bug，推动解决了MTK底层USB Driver的bug。



录音流程

唤醒成功率低

优化以后，小Q机器人的唤醒成功率有明显的提升，3m距离安静唤醒成功率提升到97%，跟竞品对比，我们的唤醒成功率略为超过它们。

测试数据汇总：

1. 6.180.4.25测试数据（带DRC版本更新）

唤醒情况

din:14411519239426201 3	安静环境		
	1m	3m	5m
唤醒总数	100	100	100
唤醒成功数	98	97	99
唤醒成功率	98%	97%	99%

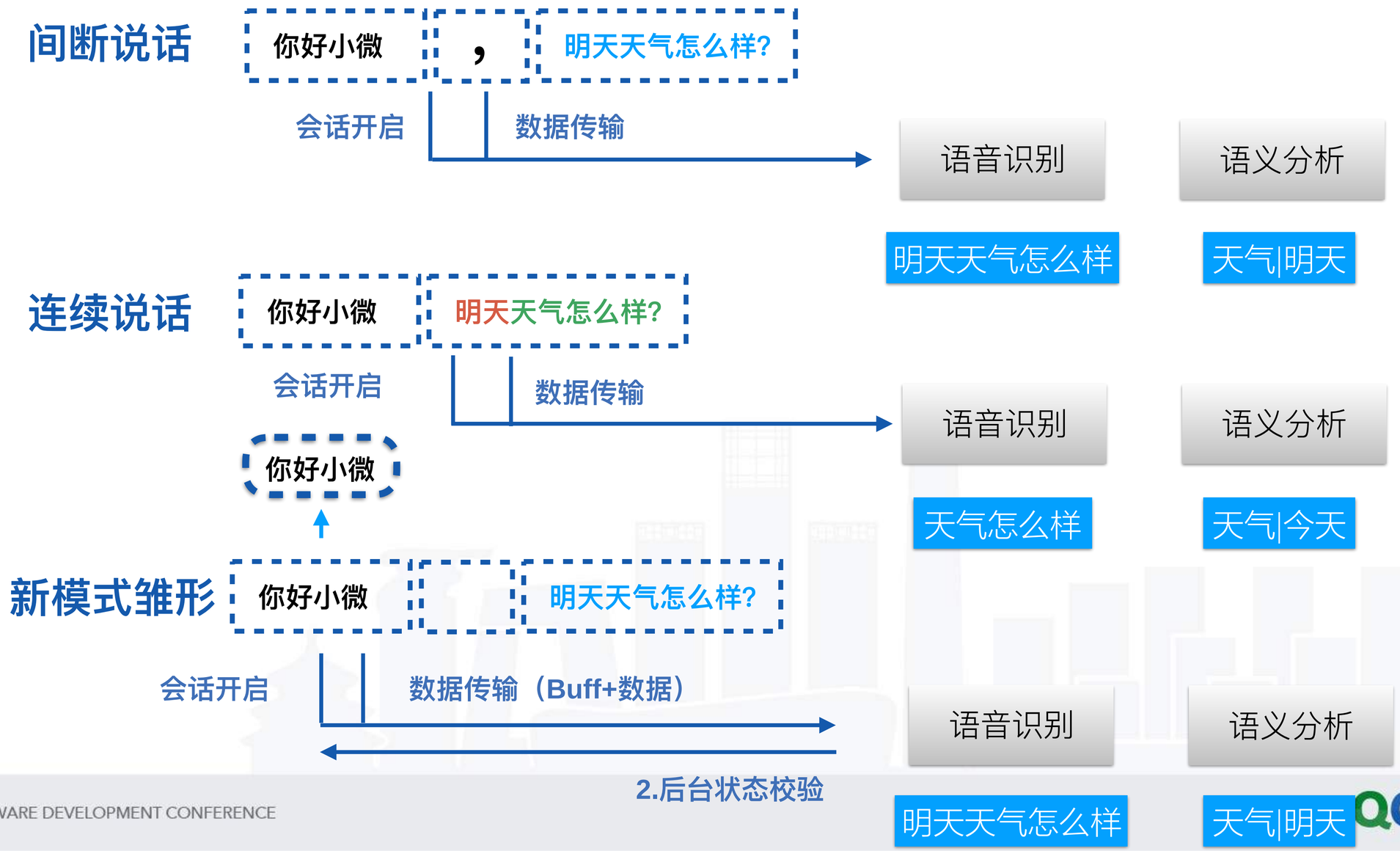
唤醒体验优化

现有模式痛点：

1.唤醒纯客户端识别，内置识别库为一次训练结果，唤醒率与误唤醒率固定，动态升级困难，9月份第一版数据 为 误唤醒为 10小时8次，唤醒率为 60%；

2.只有当唤醒后，开启会话后，用户的语音内容才会被传输至后台，在用户群体差异化情况下，不同习惯容易造成本身语音内容丢失，进而语义分析不准确，导致内容异常。9月份第一版语义的成功率为 70%

用户唤醒方式对比分析


间断说话
你好小微
，
明天天气怎么样?
会话开启
数据传输
语音识别
语义分析
明天天气怎么样
天气|明天
连续说话
你好小微
明天天气怎么样?
会话开启
数据传输
语音识别
语义分析
你好小微
天气怎么样
天气|今天
新模式雏形
你好小微
明天天气怎么样?
会话开启
数据传输（Buff+数据）
语音识别
语义分析
2.后台状态校验
明天天气怎么样
天气|明天

高可用唤醒方案

[唤醒拆分方案]

提高唤醒率，降低误唤醒率

【分析】：**拆分本地唤醒及后台唤醒**，本地唤醒提高唤醒率，后台唤醒降低误唤醒率，后台部分可以实时动态调整

无缝支持连续或断续语音指令

【分析】：唤醒词拆分，提前开启会话，可以达到连续发布语音，保证内容完整性

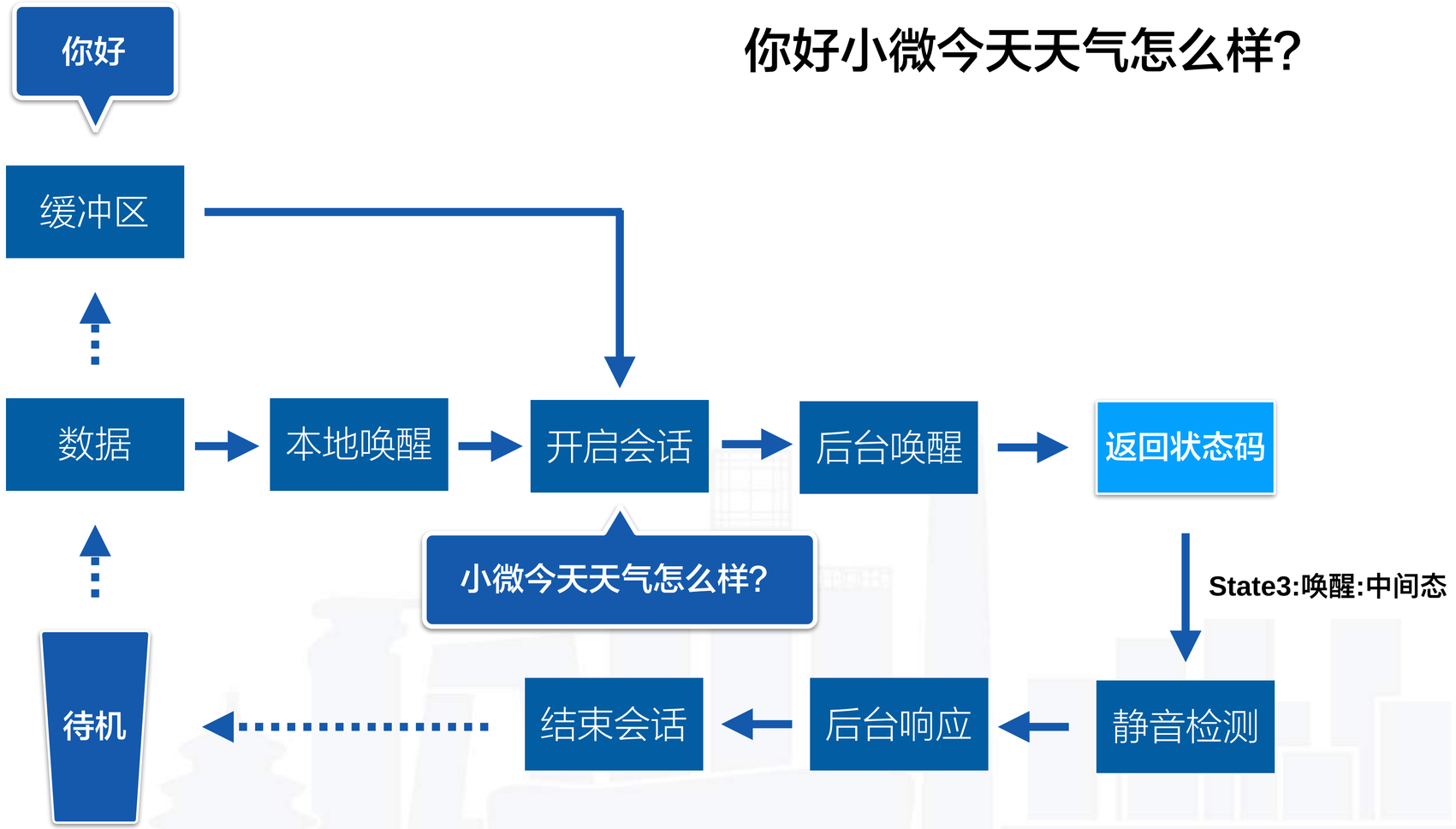
+

[后台状态机管理方案]

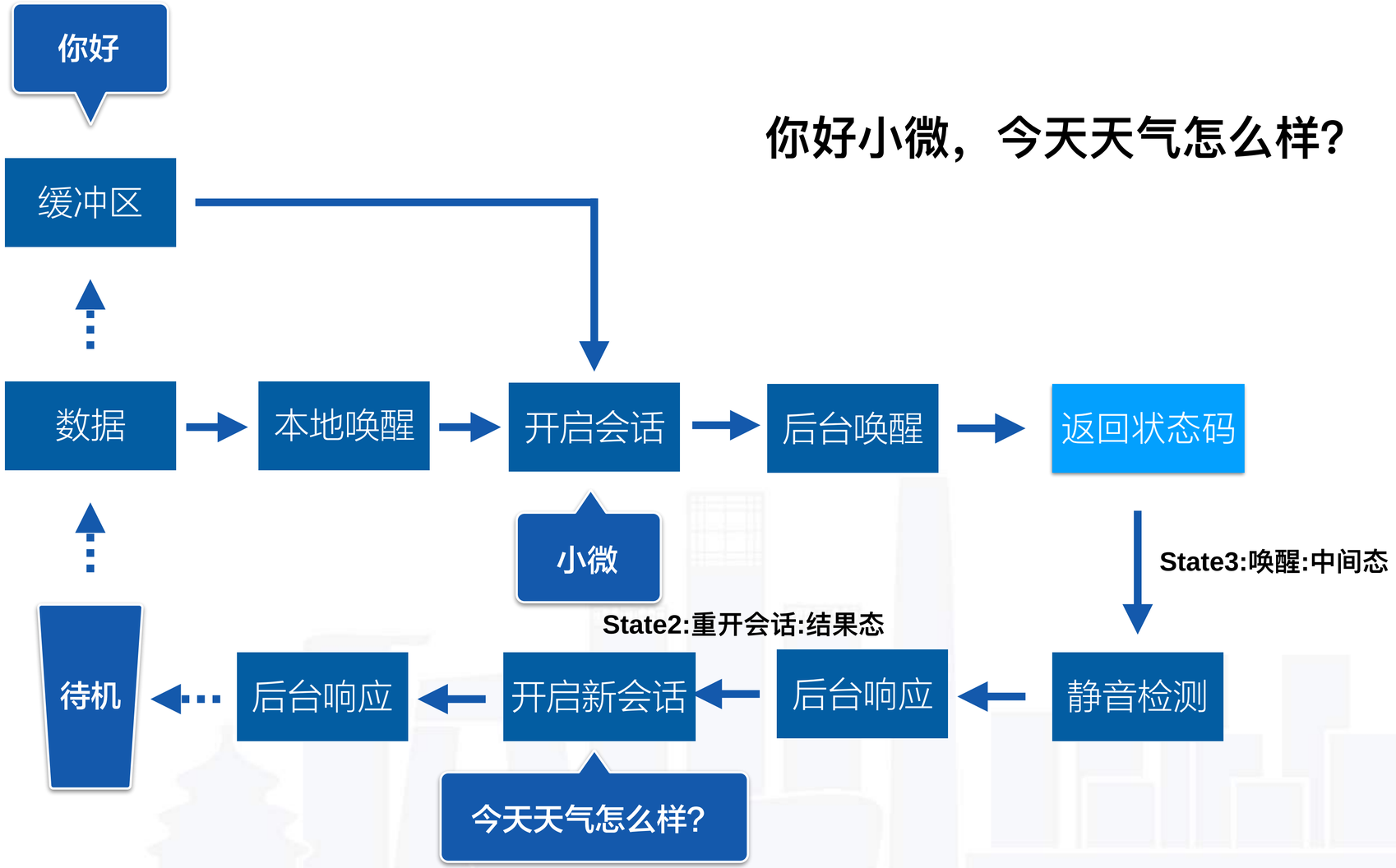
针对用户发出语音指令后，可能产生三种最终状态的交互状态，由后台返回3个不同状态的标志位供客户端处理。

State1.未唤醒。(结果态)
State2.静音检测有停顿，需要重开会话。(结果态)
State3.唤醒(中间态)。

高可用唤醒方案—连续唤醒



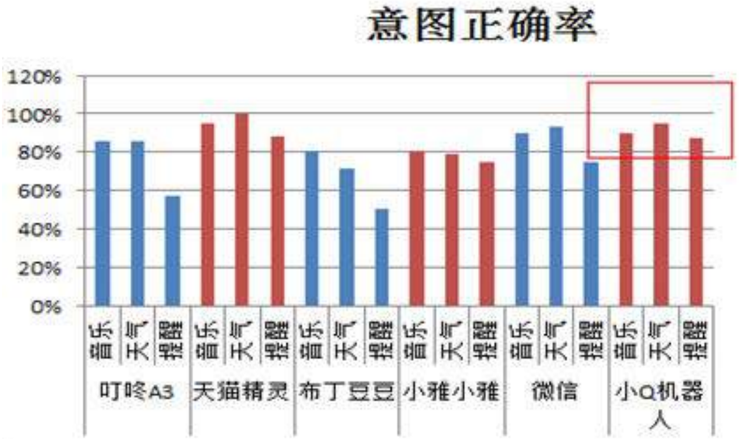
高可用唤醒方案—间断唤醒

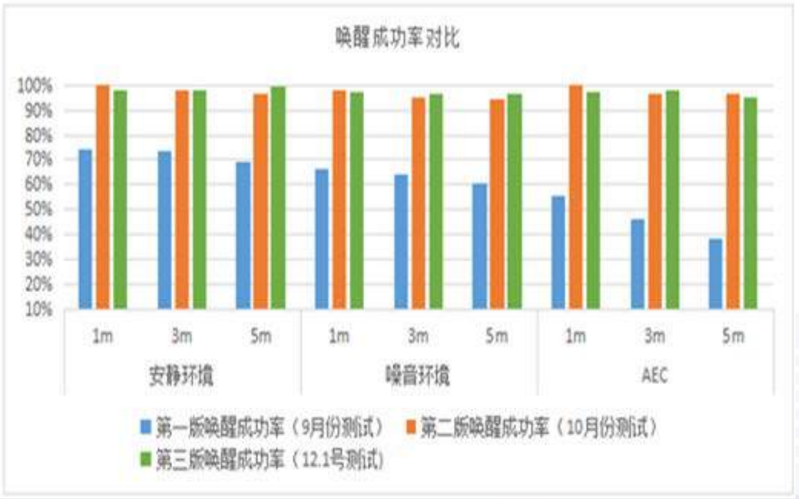


你好小微，今天天气怎么样?

高可用唤醒方案—数据对比

- 保证用户内容完整性，提高语义和意图准确率
- 唤醒拆分，动态调整，有效提高唤醒率，并降低误唤醒率，误唤醒率（8/10h -> 1/10h）





硬件项目之坑

产品定义

外观设计

功能手板

概念设计

外观手板
结构手板

开模

一试

二试

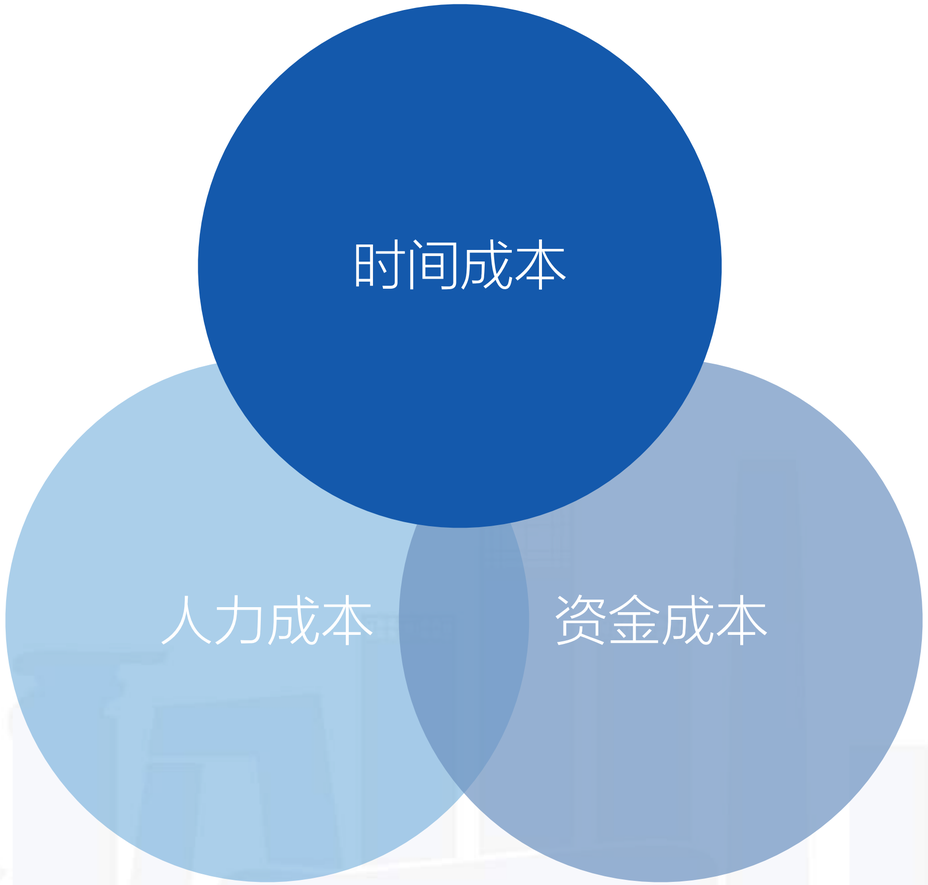
首量

批量

硬件项目之坑

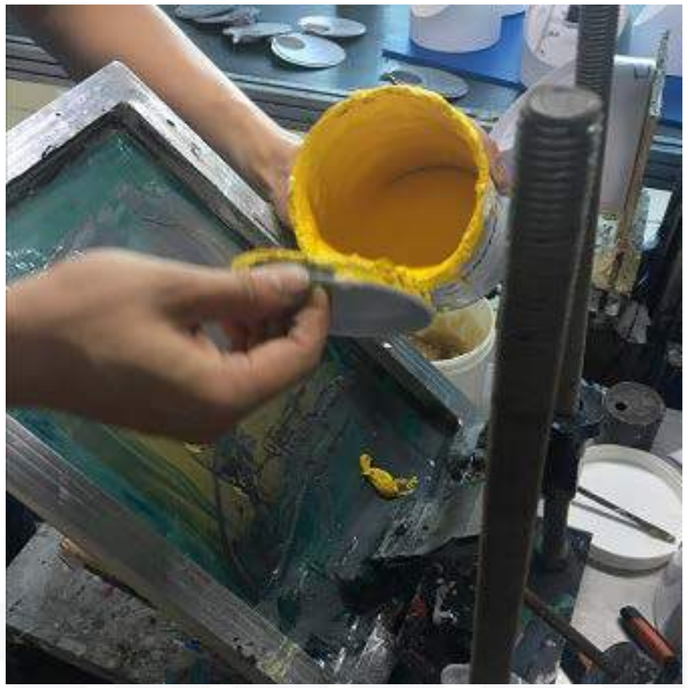
1. 周期长
2. 供应链波动
3. 成本高
4. 前期充分准备

变更成本高



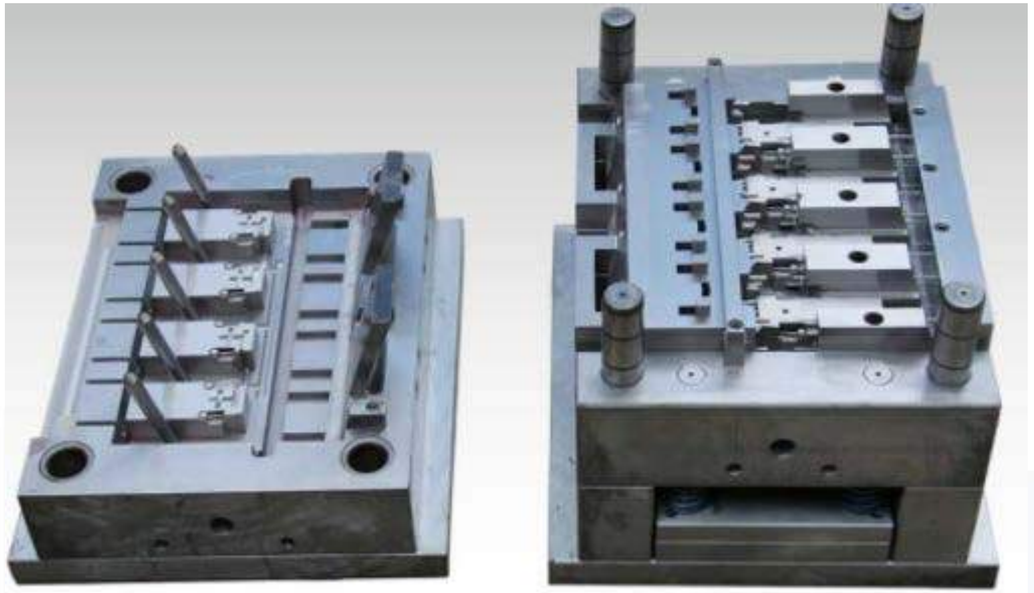
变更成本高

这个颜色不够满意



变更成本高

这个按键的位置改一下?



供应链波动风险

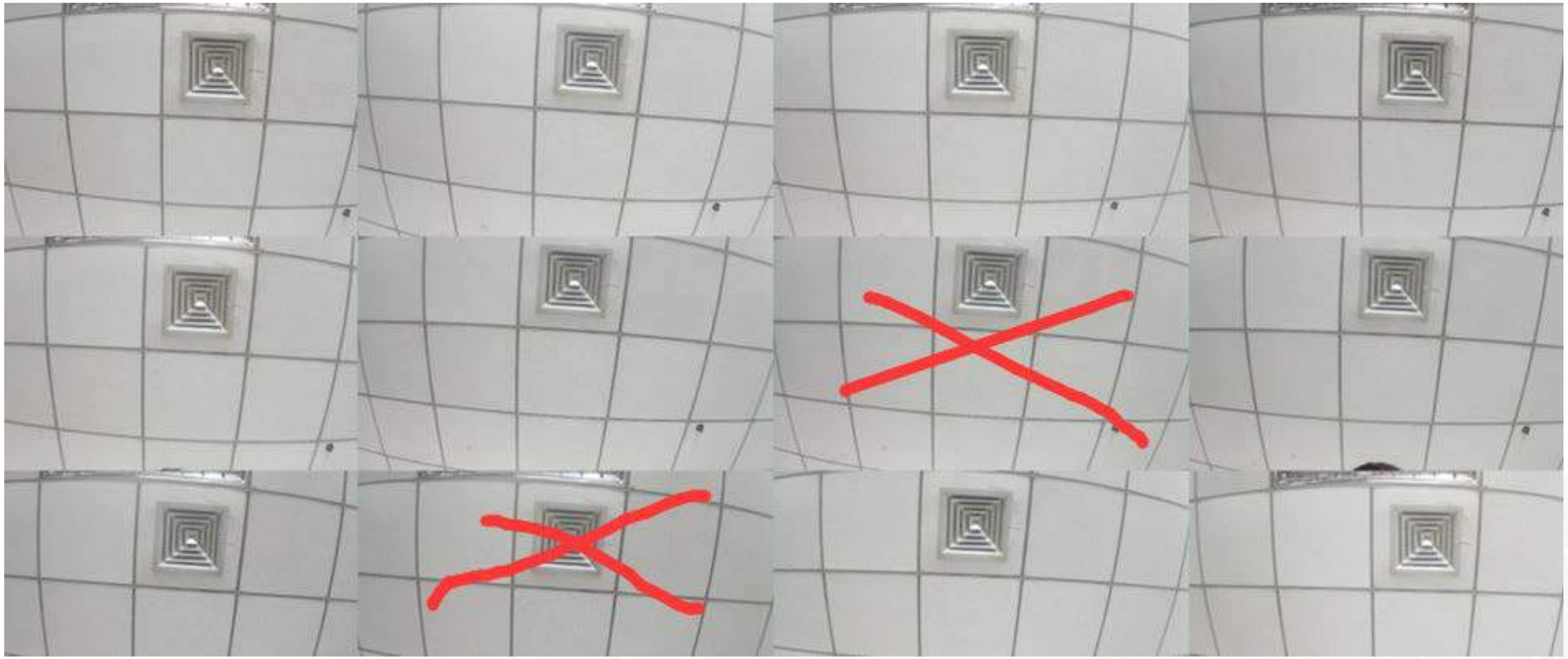

整机供应商
模具
摄像头
屏幕
塑胶件
电路板
屏
油漆
镜片
玻璃盖板
喷涂
传感器
触控
备份
备份
备份
审厂
商务谈判
合同制约

供应链波动风险

交期可以提前么？排期

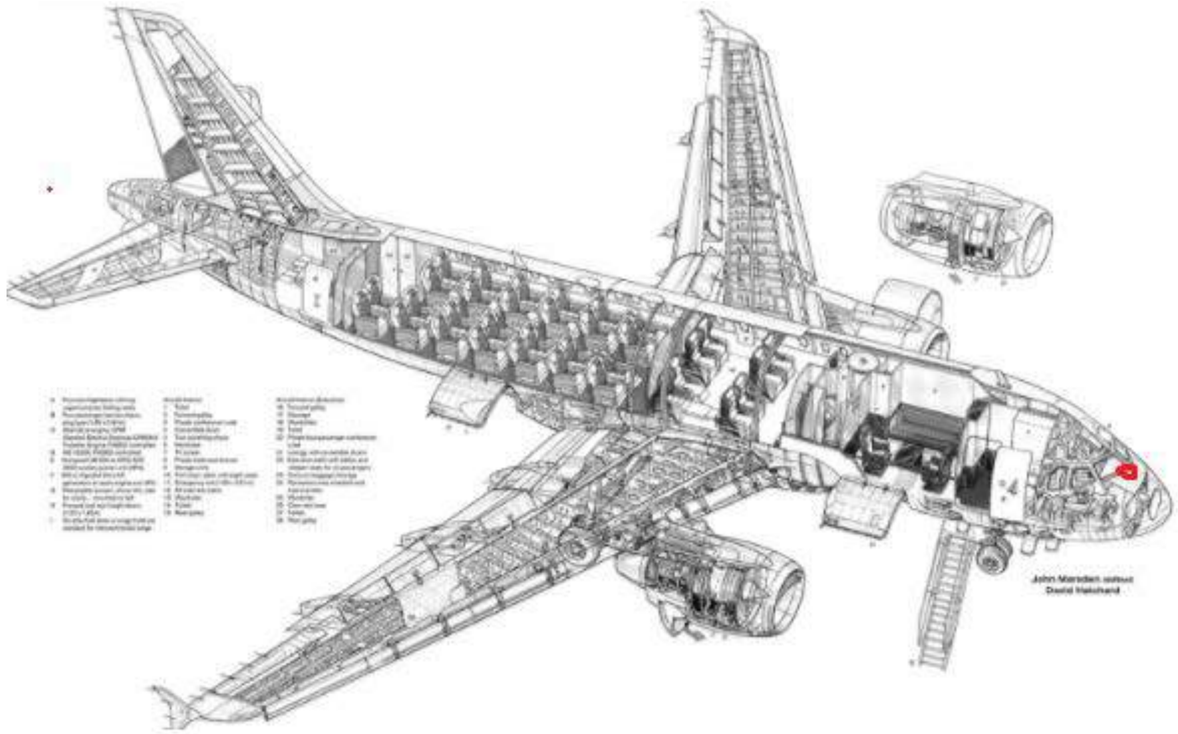
质量风险

良品率：产线上，最终通过测试的良品数量占投入材料理论生产出的数量的比例。

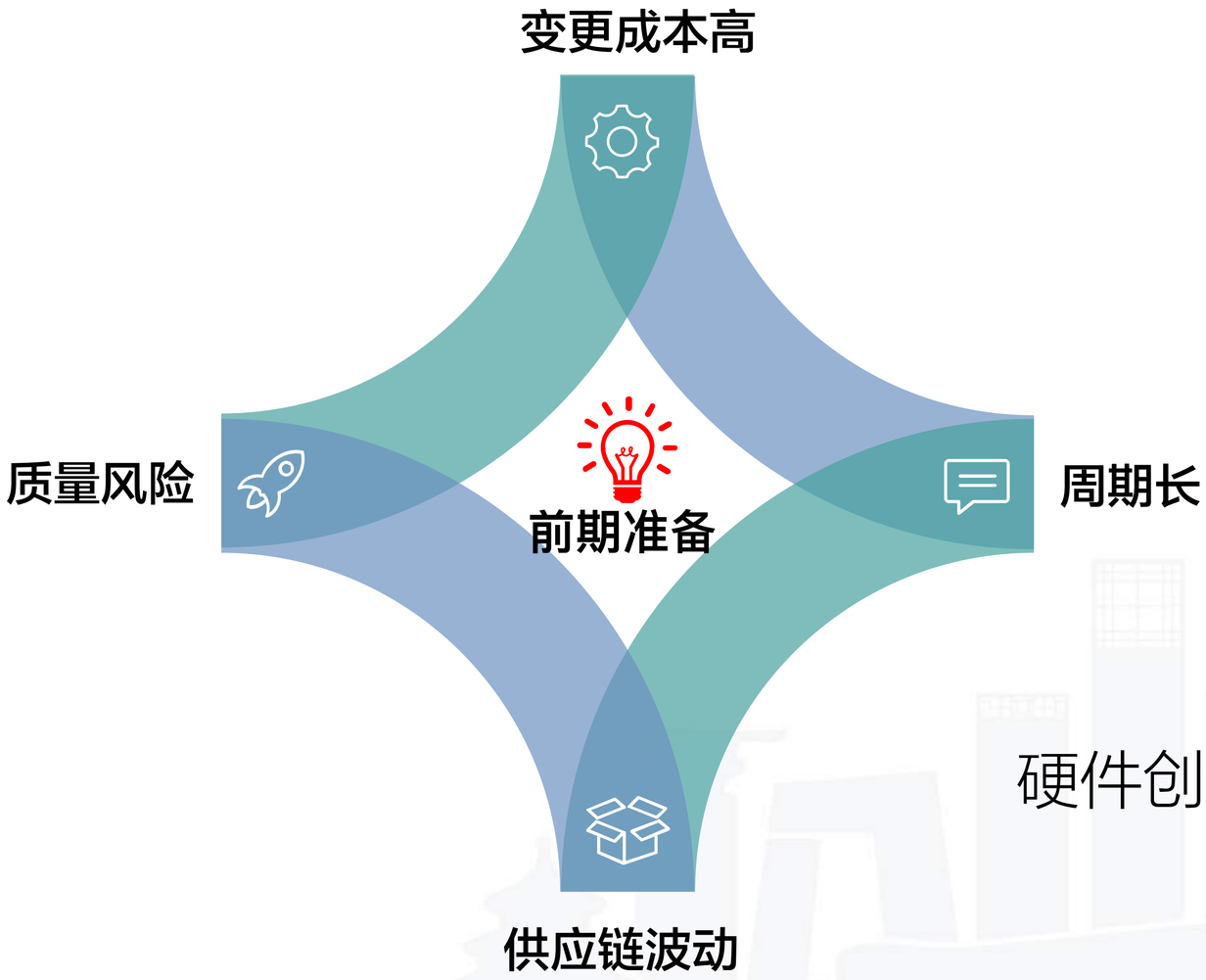


质量风险

毁灭性的bug



前期充分准备的重要性



硬件创新必须一步一个脚印的按部就班。

总结

总结

1 聊天机器人

起源

- 海量语料
- 数据检索
- 独家配方数据Rank

2 腾讯云小微

发展

- 能力整合
- 任务对话支持

3 小Q机器人

落地

- 落地方案
- 系统架构
- 硬件填坑之旅
  - 响应速度优化
  - 唤醒成功率优化
  - 唤醒方案优化
  - 语音识别优化
- 硬件项目之坑

关注QCon微信公众号，
获得更多干货！

Thanks!

INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

主办方 Geekbang. 极客邦科技 InfoQ