一、抖音冷启动场景测试用例

1.UI：（系统状态栏沉浸式检查）

2.功能：

新老用户：新用户引导冲突、老用户覆盖安装

登录态判断：登录态收藏点赞等状态的同步、登录态操作点赞等交互、未登录下调用登录接口、

启动：push跳转冷启动、首次启动广告、视频、直播、图片视频、接口拉取时机、下拉刷新

首页框架：tab加载和切换、reco推荐首页tab定位

上下滑容器：通用组件加载是否正常、上下滑

操作交互：滑动进入P页、点赞收藏关注、负反馈、进度条拖动、长按屏幕、跳转页面后操作返回状态同步

视频相关：不同编码方式的视频解码、清晰度、倍速播放、音频视频同步、耳机或外音播放、播放卡顿、播放暂停、上下滑续播or重播）

Ab实验：

3.兼容（安卓iOS不同系统、不同分辨率、中英文、暗黑模式）

4.网络（无网下冷启动、弱网加载、无网有网状态切换、错误重试、播放过程中断网播放缓冲的视频、插件下载失败）

5.中断（来电话、音乐软件后台播放、切后台、系统通知栏遮挡使activity不可见、锁屏）

6.性能（启动时长、首屏时长、cpu、内存消耗、切后台性能

二、视频卡顿原因：

1.网络：

用户侧网络不稳定

Cdn 质量抖动导致下载缓慢

cdn和用户之间的链路出现问题导致下载缓慢

推流端网络不稳定

2.视频源

视频源录制本身存在丢帧问题

3.视频编码

视频和音频时间戳不同步，导致渲染出现问题解析错乱出现花屏或卡顿等现象

视频流在采集编码的时候编码方式与传输服务器不兼容或码率帧率分辨率等参数设置不合理，就会导致在解码的时候出现错误，发生卡顿

编码帧率设置过低

编码码率设置过高

4.视频解码，渲染丢帧出现渲染卡顿

设备性能低，或者设备后台进程较多cpu占用过高超过80%，或者播放器软件原因，导致解码速度降低，低性能设备播放高清视频、低性能设备解码高清晰度视频出现丢帧

因为CPU或GPU的原因，帧数据还没有更新到帧缓冲区，就会继续读取上一个帧数据，在一个VSync时间内显示了两次帧数据，**就会造成卡顿现象**

**码率、帧率、清晰度、解码错误率、预下载、缓冲大小、错误重试、播放器开播失败率、用户主动退出率**

**、网络、首屏时长、起播卡顿、开播失败、播放卡顿次数**

**卡顿原因：影响卡顿的两个变量为视频本身的码率和用户网络的带宽。**

**服务器cdn线路问题、设备问题（设备硬件对清晰度兼容，或者不支持软编码形式）、视频源问题（视频源本身存在丢帧问题）、视频流（音视频对应的时间戳不同步、或者视频流在采集编码的时候编码方式与传输服务器不兼容或码率帧率分辨率等参数设置不合理，就会导致在解码的时候出现错误，发生卡顿）**

**测试点：**

**弱网无网切换网络进度条清晰度切换，其他资源播放、视频最大容量**

**快进暂停横屏耳机投屏权限悬浮窗默认加载图、错误重试**

**音视频不同步**

**渲染丢帧：渲染耗时长，导致实际渲染帧数变少，切前后台**

**解码丢帧：解码耗时长导致解码丢帧**

**根据不同的编码类型去做硬解或软解，手机性能分辨率上限，**

**硬解：性能损耗小，目标提升硬解覆盖率**

**软解：硬解失败会转到软解，硬解会设置最大同时创建数，个数占满会转到软解**

**编码格式：hevc(h265） h264、kvc**

**开播失败：没有到用户首帧解码失败或网络原因导致**

**开播后失败：播放器内部出现各种错误码，网络原因，解码原因，播放器内部错误等等**

**丢帧：包括渲染丢帧和解码丢帧**

5.设备

设备后台进程较多cpu占用过高超过80%

低性能设备播放高清视频、低性能设备解码高清晰度视频出现丢帧

卡顿监控：

码率、帧率、清晰度、解码错误率、预下载、缓冲大小、错误重试、播放器开播失败率、用户主动退出率

、网络、首屏时长、起播卡顿、开播失败、播放卡顿次数

二、卡顿排查总结：

1.查播放日志（缓冲时间、播放时间）、下载日志、客户端日志、cdn日志

2.通过日志整理时间线根据反馈的问题在数据中找到对应的体现

3.查看其他有异常的数据，如：某个模块耗时异常、出现错误码

三、准出问题排查流程

版本回溯，上个版本有问题，查找对应的mr分析，上个版本没有问题，从开关或放量分析

四、线上问题排查流程

1.确认影响范围及时止损：版本、安卓iOS、用户量

2.影响范围比较大，确认客户端问题还是服务端问题，确认开关放量情况，确认客户端发版窗口，是否能够热更新

3.影响范围比较小，具体问题具体分析

排查手段：

1.尝试本地复现：明确用户出现场景、安卓or iOS、版本、uid、手机型号

2.捞取客户端日志、在日志平台查询操作日志、查询用户命中实验情况、

3.查询用户接口返回内容，服务端日志、GRPC日志

3.查询sql

上线前合入代码，没有最后一道准出检查机制，当研发在上线前临时改动未周知QA回测时，问题直接遗漏道线上，可规范pipline流程，增加上线前接口自动化用例检查卡点。

**线上问题排查痛点：**

**1.客服反馈的部分问题重复：需要研发和qa排查，需要提供工具给客服自查**

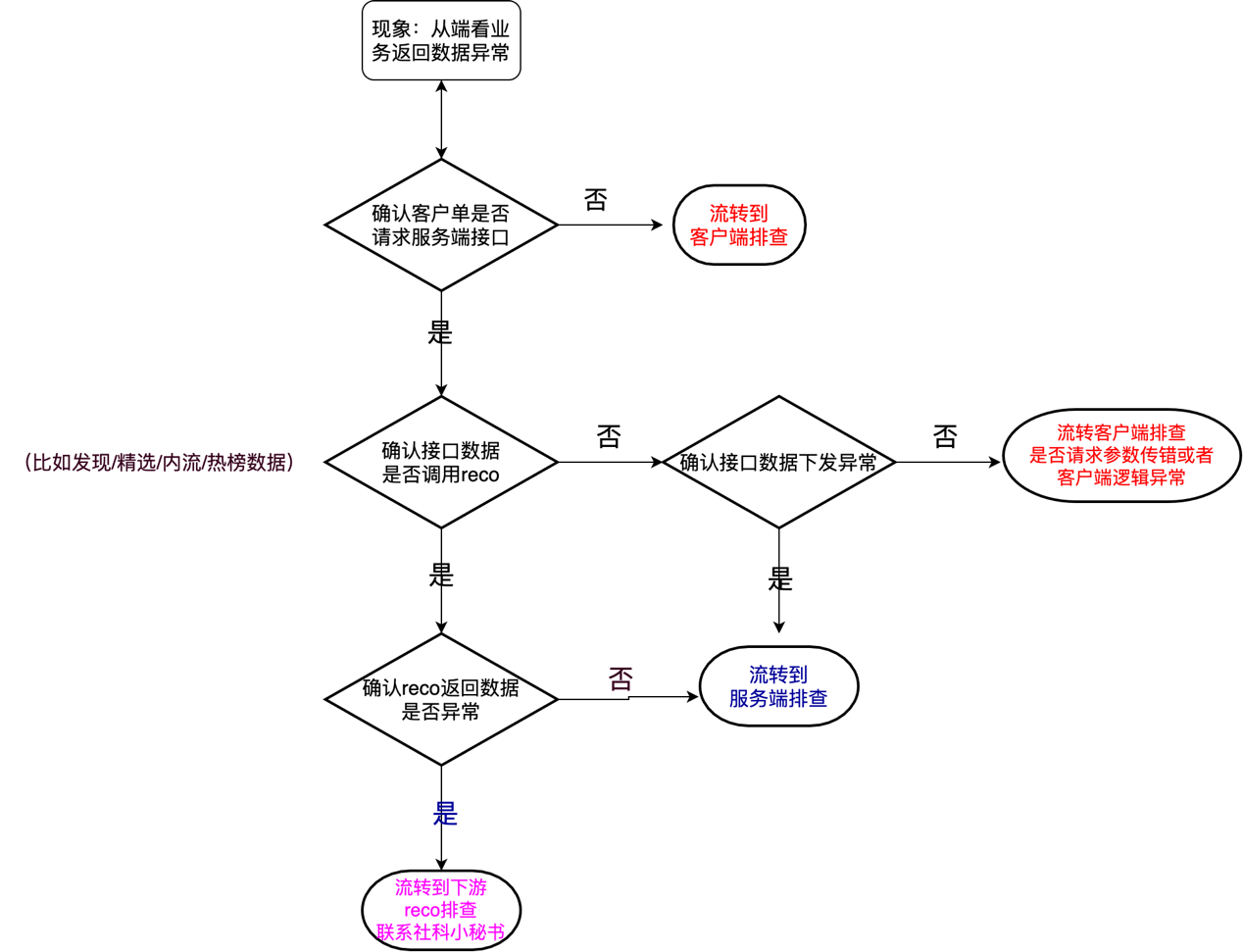
**2.问题难复现：没有API或rpc日志来还原问题现场后分析日志**

**3.问题快速定位难：业务数据经过多个业务的服务，很难区分那一方出现的问题，也很难区分是服务端问题还是推荐策略问题**

**4.耗时长：服务端和客户端策略一起介入，耗用人力成本**

**5.不能实时查询日志，只能查询t+1**

分析过程：



流程建设：

1.埋点走查流程

2.埋点回测自动化检测

3.发板前核心数据确认

4.API字段速查索引

举例：

1.用户刷到的视频有重复的视频

查视频ID，查询reco RPC日志，两次返回两个相同ID，第一次客户端未曝光

2.批量下载埋点

埋点评审人员不齐，需求理解出现偏差—需求评审和交接流程

埋点用例缺乏评审—用例设计和评审，核心埋点范围确认、埋点回测

埋点测试未走检查流程—埋点验收流程

灰度阶段数据验收流程----核心数据报表

线上阶段放量规则不严谨---开关审核流程

需求交付质量低—提测标准未通过，代码review流程等

过程中需求多次变更—工作量未重新评估

**自我介绍：**

我先简单快速自我介绍下，面试官您好，我叫xxx，下面我从工作经历、项目节奏、项目经验、主要工作内容进行介绍吧，目前在快手工作，工作时间是4年，上一家是猎豹移动，目前主要负责快手app客户端的测试工作，我们的项目节奏是从需求到代码合入一周，灰度到发版一周，测试过的项目有很多业务模块，比如分享、feed流测试、粉丝团、此外还会应对紧急项目和大型项目的测试，因此项目变更的适应能力和对项目的风险把控能力有一定的成长，同时在跟进业务的同时也会通过过程中存在的问题去分析并探索测试在质量提升上可以做哪些事情，比如流程规范的改进，测试方案的总结，线上问题前置等

**你目前负责的业务重点是什么？难点是什么？**

重点是对需求的质量交付，产品需求的功能上线

难点：在倒排的项目中，怎么在既定的时间里保证测试质量，并思考项目的合入会不会给用户带来体验问题：包括性能、服务器压力、网络预加载、对框架的依赖，对交叉实验的依赖）

1.在完成业务基本功能的同时，需要思考产品形态和实现方案是否对用户体验、性能、服务器压力的请求（比如：首屏播放时长、滑动加载卡顿、接口多请求）

2.从技术层面上考虑：基于对技术方案的理解保证测试效率的提高和测试范围的准确性，

不仅仅是保证功能的实现，还要确认框架侧对业务体验带来的影响（如弹窗优先级、气泡优先级、开关生效时机、插件下载时机），基于对业务和技术方案的理解去构造异常场景测试用例（比如：弱网、预加载、客户端兜底策略、服务端兜底策略、中断测试、和系统的交互等）

3.问题前置：预知问题出现的可能性，需要做线下的性能筛查和竞品评测

**如何从体验的角度去指导测试工作？**

围绕目的：提升产品体验

可以从多角度进行：线上bug、性能、产品形态、线下体验、线上数据的变化都会影响用户体验

所以可以从多个方向去开展测试的工作：bug测试场景的延伸、线下业务交叉的体验、竞品对比发现的性能or体验问题、对反馈问题做聚类处理、对产品差异的整理

**推动开发去做的技术专项举例：**

双列feed保证4排小卡，一排2个，插一个大卡，线上会出现小卡奇数个出现空白占位的情况，推动开发去做技术优化，空白格占比下降50%

通过对弱网测试的对比和体验，分享失败率下降3%

**为提升用户体验做了哪些事情：**

提升用户体验除了提升业务质量，最快解决用户体验问题，还要提升产品体验

**提升线下业务质量：**就需要在线下、灰度、线上阶段根据问题做出对应的改进，将问题左移到线下阶段

**缩短用户线上bug持续时间：**线上问题的排查效率（驱动多个团队实现日志服务平台）、数据监控的完善、UI异常的检测、kmonkey的运行、自动化的运行

**提升产品体验：**和运营合作、驱动开发去做技术专项（秒开率、空白格）

取得了什么收益

**如何衡量你负责的业务体验好还是不好？**

业务质量：线上用户bug反馈量、线上bug量下降趋势、崩溃趋势下降

性能体验数据：内存cpu卡顿数据、卡顿次数和竞品对比

用户体验数据：页面加载耗时、首屏耗时、完播率、丢帧率、卡顿时长和竞品对比

产品形态和策略：产品功能上线指标正向pv 、uv、ctr、使用时长等、功能的可用性、易用性和竞品对比

运营平台上用户的建议：功能的改进建议、功能体验的负向问题

**你是如何保证测试质量的？**

质量好还是坏并没有一个精确的数字去量化，减少影响用户体验的bug，同时不断优化用户体验

**对于测试来说保证测试质量其实是将测试左移**

**将问题最大限度的在线下和灰度阶段发现**

**发现并解决更多的线上的留存问题**

通过对线下、灰度、线上几个阶段的数据分析，线下过程进行拆解找到我们可以改进的地方，找到问题的解决方案，在每一个阶段做出的改进都可以在帮助我们去提升产品质量

**线上：**

分析：漏测问题、线上故障、versionfix、用户体验问题、用户反馈问题的分析

改进项：对用户反馈和内部反馈的流程改进最快解决问题，线上接口的自动化巡检、线上数据看版的建立和报警、上线后放量规则的统一、异常弹窗异常UI的检测、kmonkey、例行自动化

**线下：**从需求到代码合入到灰度到线上发布，在不同的阶段对不同的角色会有一定的要求，数数据做归因分析每个环节上存在什么样的问题

测试和研发不同指标的分析：回测占比、回测归因、分阶段bug数、P0P1占比、打回数、reopen数、问题类型分析

改进项：case设计、用例执行、提测、环境准备、测试范围和技术方案评估、风险评估、埋点测试流程改进、自动化介入pipeline、kmonkey、覆盖率

**灰度：**

做了哪些事：数据监控、性能准出、竞品分析，改进灰度放量、性能优化、每周进行业务交叉体验、大回测

如何保障测试质量：

质量不只是测试的事情，包括整个团队，产品、运营、研发、设计

1.人员质量意识的提升：

2.全流程的把控：每个阶段做好严格质量把控，需求预评审、评审、方案设计、case设计、case评审、提测、测试、灰度、上线

规范的流程更有助于项目的顺利进行，也可避免出现低级问题

评审：方案评估、产品调研

研发：流水线检测、准入准出、自测推进、

线下：测试阶段提测质量、测试流程改进、免测机制、代码覆盖率、数据准确性保证、风险意识、产品设计走查环节

灰度：灰度监控、稳定性观察、灰度放量、

线上：问题复盘、改进落地、监控

其他：代码解藕、技术优化

3.保证产品全面质量：不仅仅是业务功能，还包括产品的易用性、兼容、稳定性、性能（用户反馈、性能准出

、竞品对比、交叉体验）

**你做了哪些提效的事情？**

在工作中和的任何阶段做出的使我们的工作更加流畅的事情，都是可以提高效率的

**不限于工具的产出，工具的使用，数据监控看板的建立，以及工作流程的改进，对线下和线上不同阶段提升效率**

比如:

**1.流程上的改进：**

**通过对质量的分析和项目复盘，提出对测试前、测试中、测试后各个阶段进行流程优化**

比如:

测试前：免测机制、code review，数据准备，用例完善，测试分工，自测

测试中：走查

线上：运营合作、回测流程、线上问题处理流程、研发产品定期的访谈

a.和研发产品运营的合作模式（提测、反馈、走查）、、满意度调查；

b.测试回测工作量的评估（回测case的梳理优先级标签、确认技术方案准确评估回测范围和方案可行性）

埋点自动化

c.需求与评审、评审、case评审、数据准备、前后端测试配合、代码review、范围评估、免测评估

**2.推进工具的开发和使用：**

线上问题定位：解决线上问题的快速定位（日志服务平台开发和问题筛查的指引）

性能测试效率：性能置信工具的开发、自动创建测试报告的工具开发

竞品耗时测试效率：irs AI检测平台

回测效率：对测试用例的重构、

问题解决效率：对team任务的实时提醒、超期提醒、kdev的使用推进自测流程

**3.数据监控的价值最大化**

**线上数据看板的建立解决排查问题耗时长的问题：**

实时监控，增加监控维度和粒度

举例来说：可以在测试阶段和线上问题排查方面提高效率，

线上阶段：难复现的bug解决率10-20%，问题处理时效3人日—0.5人日

线下测试阶段：冒烟case执行率和通过率上升、P0P1bug的减少，需求打回比例下降，测试人日下降x%

灰度阶段性能测试：2人日-0.2工作日

数据问题：增加监控维度和实时监控，分享问题排查经验，2人日-0.3人日

测试用例怎么培训

大的规范、大的标准、大的框架套用、review机制、查漏补缺

倒排的需求：

优先级、技术方案可行性评估、前期准备卡点、人力、过程中风险把控、开关放量策略的风险控制

测试难点：

怎么保证线上问题左移

趣头条

1、有什么好的方法提高测试效率的吗

2、linux查询实时日志、查询100条日志、查询非实时日志

3、期望薪资，和现在相比涨幅大概多少

4、抓包工具charles代理修改了头文件

5、工作上规划

抖音北京：

项目效率如何提升

如何保证项目质量

说说发布一条的数据流程

story简单介绍一下

story的发布时长是多少为什么是11s

登陆图片验证码是用来干啥的

比组内其他人的优势是什么

一瓶水怎么测试

问我为什么换工作

社交对于快手来说起到了什么作用：提供消费和生产的原动力，促销费促生产

b站

sql内连接外连接区别

快手关注页码率多少

快手自动化怎么处理allert

关注页怎么测

adb命令怎么抓log

对性能了解的少什么码率啊之类的

为什么换工作

腾讯

python装饰器应用场景

接口自动化有做过吗，没有

ui自动化失败的原因

java的快排以及时间复杂度

一个数组顺序是乱的，比如1-100中间少了一个数，怎么判断少的是哪个，直接遍历怎么找

多线程造成死锁的原因

自己是否做过一些小工具

http请求头和请求体都有啥

soul

怎么测试微信

测试一支笔的异常情况

给你一个关注的case怎么分析问题

项目流程

头条-上海

怎么提升效率

最近测试的一个需求

说一下自动化测试框架以及流程

给一个问题怎么定位—发送消息，这边显示发送成功，另一边没收到

服务迁移怎么测试

服务迁移时，服务器在苏州，我们在北京压测准吗

一个url从输入开始到请求结束的整个过程

tcp的三次握手

对于提升项目效率你做过什么

英语流利说

http错误码

charles原理

遇到bug怎么分析问题

平时测试工具

adb装包命令

后面规划

**介绍快手的业务：**

快手业务框架

分为两种消费形式：视频播放、直播，视频播放和直播都使用上下滑框架，承载信息展示需要用首页框架，首页框架由多个容器构成，容器可以套容器，使tab可以动态分配，同时使业务层也能实现个性化功能

视频浏览是上下滑框架，包含通用组件，上下滑组件，业务层加工、播放组件

业务层调用播放器sdk、编辑sdk、生产sdk、埋点sdk、私信sdk、分享sdk等

对于中视频feed客户端采用mvp模型结构，以pageview的形式加载feed，点播时调用播放器sdk进行视频播放

**目前团队中的痛点是什么，怎么解决的？**

**1.产品运营 2.开发  3.外部团队 4.测试内部 5.流程**

**1.产运：**

运营提出的问题超过50%为非问题，投入产出效率低，发起非问题总结汇总

产品：需求变更、需求插入，增加需求预评审，随后排优先级

**2.开发：**

bugreopen率、提测打回，提测前环境ready、自测执行流程建立、bug reopen原因分析（通用规范解决头部问题、尾部问题个性化针对性解决）

**3.外部团队**

代码耦合对业务带来的影响：合入流水线增加code review机制，建立回测流程，边界case持续更新，4.对回测问题归因

**4.测试内部**

对外包的管理：case的设计（和后端文档的结合、边界值、异常情况的设计、实验交叉的总结和覆盖）、case评审调起主动性、测试范围的主动分析或找开发评估确认

数据监控的敏感度：需要机器算法去提升报警准确性、需要增加监控粒度来辅助排查问题，但是资源经常不足

性能测试的深入度：流程化还是很完善的

学习分享的参与度：运用到实际工作中

**5.流程：**

**推出流程改进后落地速度比较慢：找leader达成相同认知**

**回测时间占据40%：梳理回测范围、免测机制、**