

项目流程与规范

王先明 2013.02

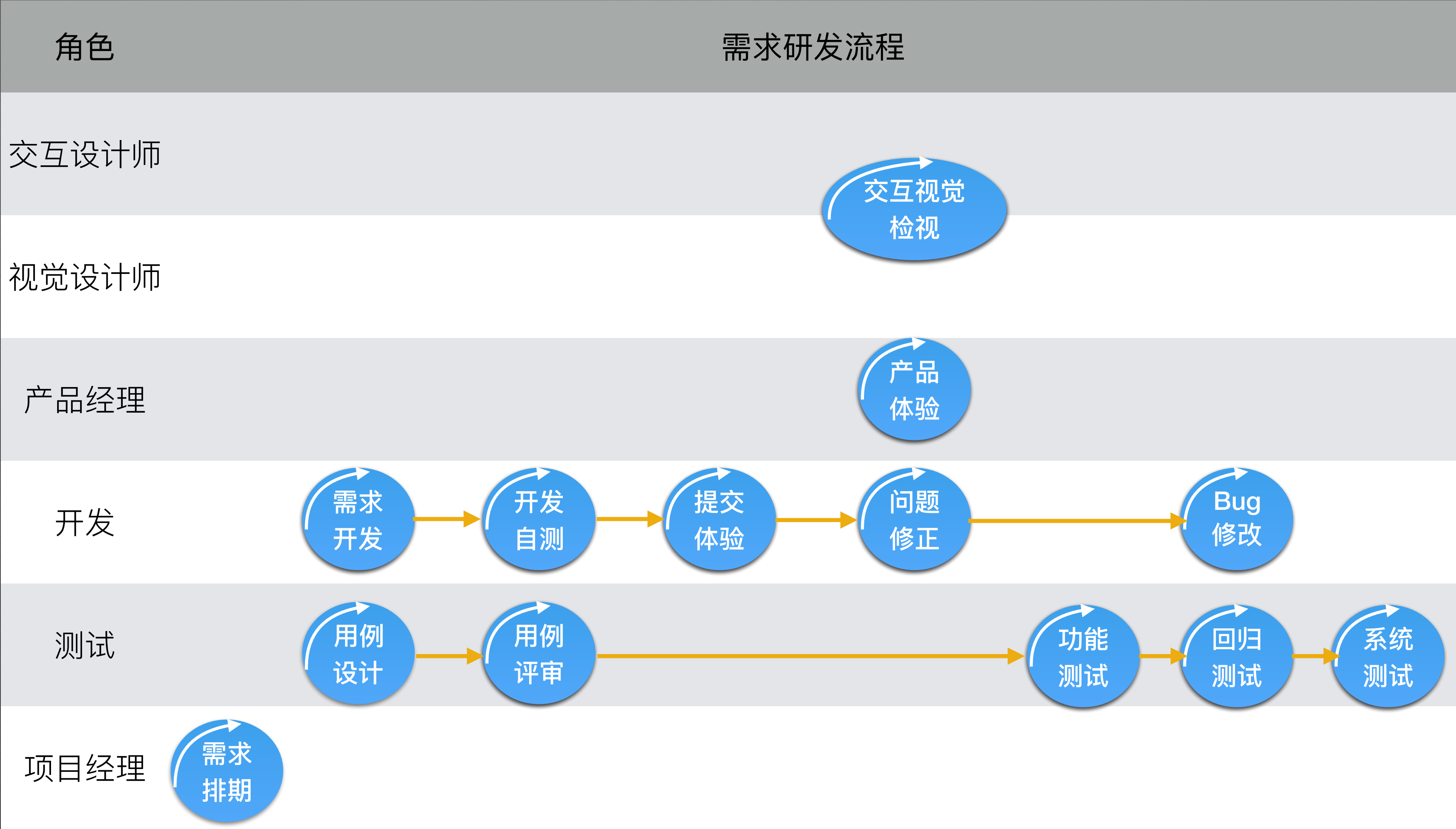




1 立项与启动

关键是让团队成员清楚：

- 1.** 版本目标是什么？（ KPI？过程中，不断的回顾和对齐目标 ）
- 2.** 我们要做什么？（ 哪些需求？ ）
- 3.** 怎么做？（ 分几个迭代实现？ ）
- 4.** 哪些人在一起做事？
- 5.** 预计什么时间做完？



迭代流程

	当前迭代	产品，交互，视觉和开发在当前迭代需要为下个迭代准备的工作				
	1/2周	1/2周	1/2周	1/2周	1/2周	1/2周
产品	1) 启动迭代，需求讲解与评估； 2) 细化开发过程中的迭代的需求更新与细化 3) 跟进迭代	1) 细化开发过程中的迭代的需求更新与细化 2) 跟进迭代	1) 确定迭代主题，目标和需求范围 2) involve相关产品经理讨论	1) 与交互设计一起规划下个迭代范围内的需求，与开发沟通技术可行性。 (产出：交互设计稿，需求文档) 2) 跟进上个迭代的产品运营与问题	1) 与交互设计一起规划下个迭代范围内的需求，与开发沟通技术可行性。 (产出：交互设计稿，需求文档)	1) 产品组同行评审下个迭代的需求文档 (产出：可在迭代启动会上讲解的需求文档与交互稿)
			跟进当前迭代	跟进当前迭代 产品体验	跟进当前迭代 产品体验	1) 跟进当前迭代 2) 发布 3) 迭代总结会
交互	按需更新	按需更新	按需更新	下个迭代需求交互设计	下个迭代需求交互设计	更新下迭代需求的交互设计，出交互设计稿
视觉	视觉设计，交付设计稿	按需更新	按需更新	按需更新	下个迭代需求的视觉设计 视觉走查	下个迭代需求的视觉设计，陆续交付设计稿
					视觉Bug 走查	视觉Bug 走查
开发	1) 参加迭代启动的需求讲解与评估 2) 开发，技术预研，调研，依赖梳理	开发	开发	开发	deBug	deBug
				与产品沟通下迭代需求的技术可行性	与产品沟通下迭代需求的技术可行性	与产品沟通下迭代需求的技术可行性 (如需)
测试	用例编写与反馈	用例编写与反馈	daily build 功能测试	daily build 功能测试	系统测试	系统测试
所有角色	参与迭代启动会					全民体验 参加迭代总结会

2 沟通与协作

文档 gitlab

代码 gitlab

设计文档

开发设计文档

测试用例

流程规范说明文档

积累的外部依赖联系方式

...

邮件&邮件组（重要的结论，邮件覆盖范围超过项目团队之外）

面对面沟通，企业微信，电话，会议（内部成员适用）

QQ，微信，WhatsApp，电话（跨部门/跨地域适用）

1. 需求（ 状态跟进，变更记录 ）
2. 任务
3. 版本迭代规划
4. 缺陷

状态流转 → 流程图

状态：
☒ 保持为已规划 ☐ 规划中 ☐ 评审中 ☐ 已归档 ☐ 实现中 ☐ 已实现
☐ 已实现并转产品体验 ☐ 已挂起

当前处理人 ...

处理意见 : 评论/处理意见...

邀请讨论 ▾

邀请: 请输入RTX号或者邮件组 [添加用户组]

通过: ☐ RTX ☐ Email

提交

状态流转 → 流程图

状态：
☒ 保持为已实现并转产品体验 ☐ 需求变更中 ☐ 实现中 ☐ 体验通过转测试 ☐ 已挂起

处理意见 : 评论/处理意见...

邀请讨论 ▾

邀请: 请输入RTX号或者邮件组 [添加用户组]

通过: ☐ RTX ☐ Email

提交

需求状态流转说明：

开发启动该需求的任务时，需要将需求的状态改为“实现中”

最后一个完成该需求下任务的开发人员，需要将需求的状态，由“实现中”改为“已实现”或者“已实现并转产品体验”，并在处理意见栏里备注体验环境配置和账号等注意事项

“已实现”和“已实现并转产品体验”两个状态的区别在于，“已实现”的需求和其他需求有关联，无法单独体验，待相关需求ready之后，再一同改成“已实现并转产品体验”的状态

产品体验如果通过则转测试，不通过，则将状态转回到“实现中”，由开发继续处理

当某需求的第一轮功能测试结束，则该需求的状态改为“第一轮功能测试结束”，版本上线之前，需要将需求的所有状态改为最终状态“测试通过”

1. 操作

项目迭代中按照需求状态流转来跟进，任务建立不作为强制要求

但建议新同事尝试使用几次，通过新建任务和跟踪任务来管理自己的工作

2. 好处

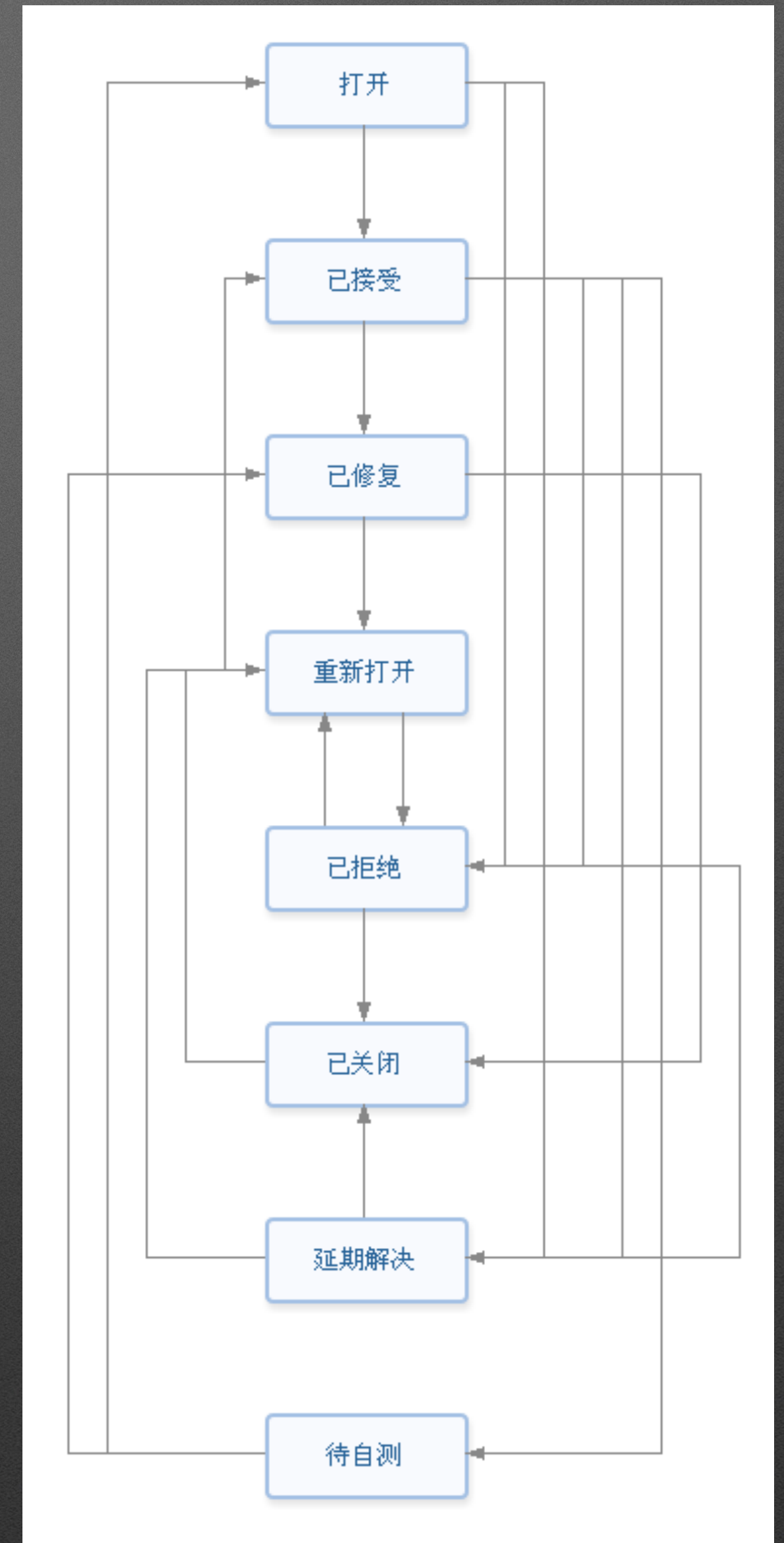
通过新建和评估任务，了解自己对任务和时间评估，帮助自己梳理工作，及时配合同事一起完成

熟悉业务之后，对需求的工作量评估逐渐准确起来之后，可以不再使用

按实际及时填写花费（不论提前或者滞后）

完成后立即关闭任务

1. 开发收到Bug之后，请尽快确认是否是自己的Bug：
若是，请将Bug改为“已接受”
若否，请尽快将Bug处理人指派给其他的开发人员
2. Bug修复完毕之后，请尽快将Bug流转到“已修复”
3. 请不要在没有沟通的情况下，直接将Bug状态改为“已拒绝”，
涉及到业务层面的Bug认知问题，请务必卷入PM 和 PDM讨论
4. 延期解决的Bug，需要卷入PM 和 PDM讨论 review 后决定



1. 识别有上下游依赖的需求，特别是外部依赖，优先处理，提前共同商定方案
2. 尽早告知上游你需要支持的时间点，根据复杂程度和经验判断，预留一定BUFFER
3. 定期询问对方完成情况，确保对方知道你对该需求的紧张程度
4. 联调阶段，优先配合外部依赖方，不要让对方等待
5. 完成之后，别忘记致谢，同时总结经验并存档

标题： 【项目名-<模块名>】 <版本号> <日期> XX需求/功能提交测试

【提测人】	
【测试人】	
【提测时间】	
【所属版本信息】	1.0 迭代
【需求文件】	http://gitlab.*.com/需求文档/管理需求规格说明书.doc
【访问入口与配置信息】	访问入口： 需要配置的HOST信息： 配置的HOST信息：
【账户信息说明】（非必填）	帐号：2355199808 密码：123@qq
【提测功能点】 注：如果对之前的功能点有更新，也需要写入并备注	
【遗留问题】	
【测试建议】 注：1）建议测试完成的时间 2）需要重点测试的范围	
【自测结果】注：描述P0用例自测结果	

3 测试与体验

1. 功能提测

按TAPD上的需求，开发完成之后，自测通过则转产品体验（备注环境配置方法）
产品体验通过之后，转“体验通过转测试”，测试取daily build 进行功能测试

2. 提交系统测试

功能测试的Bug达到进入系统测试的标准之后，由某一个开发同事统一出提测邮件，邮件模板详见PPT

3. 提交灰度前冒烟测试

Bug达到进入灰度的标准之后，由某一个开发同事统一出提测邮件，测试同事进行灰度前冒烟测试，邮件模板详见PPT

注：整个测试阶段的Bug回归，都取头一天晚上的daily build 来验证已修复的Bug

1. “致命”、“严重”的必现Bug修复100%（非必现的严重Bug不在此列）
2. 全部Bug的修复率达到85%
3. 未修复Bug，经过测试评审（如有异议，请与测试、开发和项目经理一起评审）同意，可放入系统测试阶段修复

1. “致命” , “严重” 的必现Bug 100%修复
2. 非必现的 “致命” , “严重” Bug修复率95%
3. 全部Bug修复率90%

1. “致命”、“严重”的必现Bug 100%修复
2. 非必现的“致命”、“严重” Bug修复率95% (非必现Bug的特殊情况情况需要项目组review后决定)
3. 全部Bug修复率为90%

Android版本

在功能测试阶段，采用debug 版本进行测试，以便获得更多的日志信息

在性能测试阶段，以及在进入系统测试之后，都用Release 版本进行测试

iOS版本

都用Release 版本进行测试

1. 系统测试开始，就要开始准备了

2. 重点灰度功能有哪些

3. 根据灰度重点选择灰度名单，邀请+报名的方式，哪里报名，哪里邀请

4. 灰度安装包发放方式

- 安卓用户和越狱的iphone 用户可以采用 企业微信 群发，或者广播通知的方式推送下载安装包
- 非越狱的iphone用户（正在尝试，稍后归档）

5. 灰度安装包App ID和正式安装包的App ID要不一样，确保正式上线后能强制升级
6. 如何运营灰度用户，更高效率获得灰度反馈？
7. 如何吸引用户下载？吸引和推动用户使用并积极反馈（可以考虑奖励机制）
8. 对灰度用户的行为数据进行观察，对日志进行监控
9. 灰度版本在启动页要加上“尝鲜版”的水印字样

- 1.** 灰度的时候，由产品经理和运营人员对外收集用户的反馈，开发工程师通过监控日志和数据发现问题
- 2.** 每天下班前统一反馈给测试进行提单（紧急问题需要及时反馈并提单，不必等统一时间反馈）
- 3.** 开发解决Bug单后，生成新的灰度安装包，如有需要进行下一次灰度验证（也可以请发生问题的灰度用户配合验证，并给到一定的奖励）

用户反馈监控：微博、论坛、用户等

版本数据监控：整体规模数据、关键特性数据

配置和及时升级灰度用户到正式版

版本普及运营，活跃度与功能运营

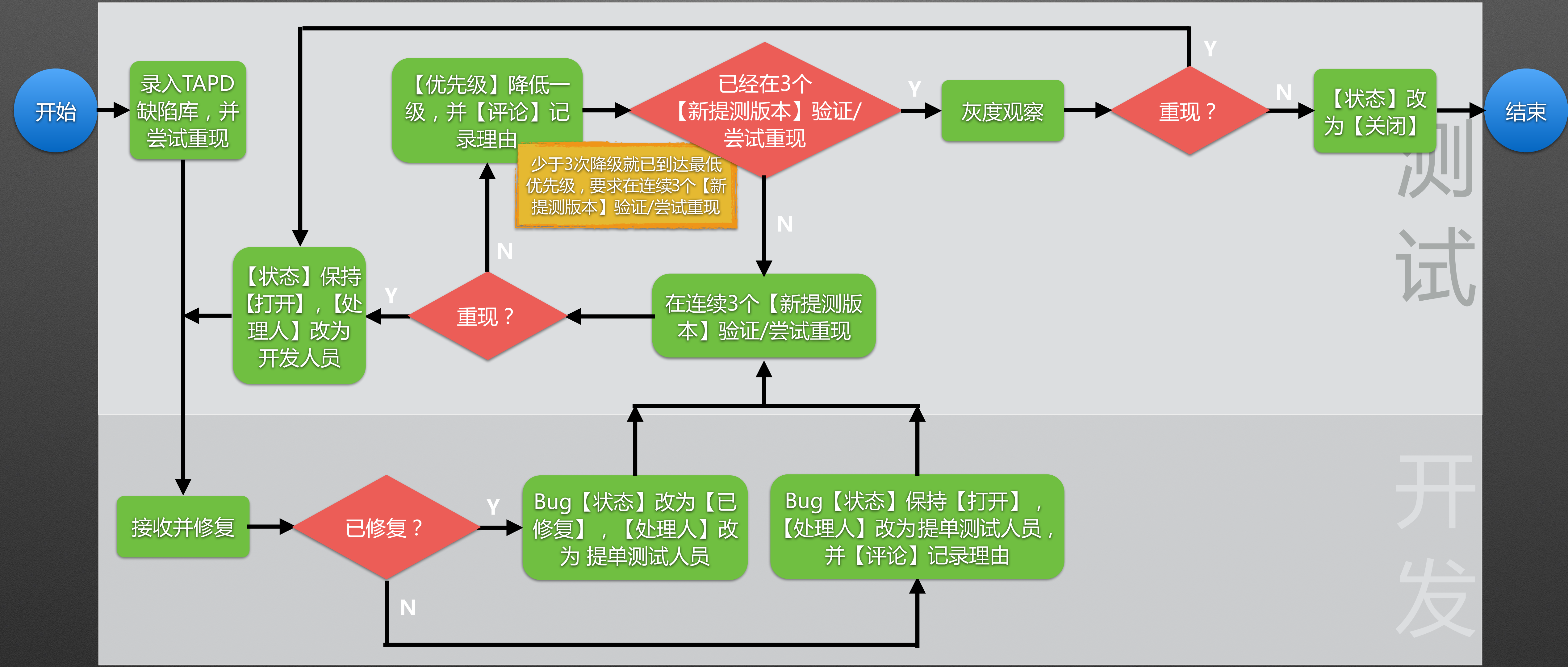
及时向老板和团队发送发布后运营数据报告

线上Bug处理流程

优先级	响应时间	解决时间	沟通方式	备注
严重	<=10分钟	<=30分钟	当面、电话、TAPD	如果开发在30分钟内无法修复，则给出对应的解决期限
一般	<=24小时	根据实际情况	TAPD	如果开发在24小时内无法修复，则给出对应的解决期限

角色	任务
测试Leader	每天上班前检查Bug是否得到及时反馈和解决，没有反馈的上升级别
开发Leader	每天下班前查看自己团队的Bug，保证及时反馈、及时解决
测试人员	提Bug时注明优先级，分配给开发人员，严重的Bug通知开发人员（当面、电话）
开发人员	及时反馈和解决自己名下的Bug，每天下班前检查一遍

无法重现的Bug处理流程



Thanks

To Be Continued...