迭代开发流程规范

迭代开发过程指研发团队将产品需求转化为最终可交付物的过程.

本流程规范主要是为了解决目前开发过程中遇到的问题，提高开发效率，各开发团队可以根据实际情况选择执行。

**活动：**

1. **PRD & 技术解决方案沟通**

在每次Sprint Plan会（原PDR评审会）前，即PRD（包括UE\UI）阶段，对于某个功能点,如果对技术实现上有任何疑问（包括UE\UI上的细节），项目经理负责协调产品经理、开发经理（架构师）共同商讨技术解决方案。

同时，对于需要其他开发团队协助的任务，需要在此阶段及时识别并提需求给相应的开发团队

PO Team: 包括业务人员、产品经理、开发经理（架构师）、项目经理，这个虚拟团队的目标是解决做什么的问题。

**输入：**

**-** PRD、UE、UI

**输出：**

* 技术解决方案（可选）

**解决问题：**

* PRD评审过程中，出现功能不能实现的情况，导致PRD、UE、UI返工
* Plan会上，无法评估工作量
* 避免无法评估有依赖的工作
* 增加工作量评估的可靠性

1. **Sprint Plan会议：**

**输入：**

* PRD文档(包括UE/UI)
* 技术改造类任务
* 技术解决方案（可选）

**输出：**

* 开发&测试任务及估算
* 开发对产品两周交付承诺

1. 需求答疑
   * 产品团队对于已提交的PRD进行一次集中统一的答疑，明确产品需要做什么
   * 对于某个未明确的产品功能点，不推荐进入到本次迭代周期中
   * 开发经理(架构师)需要确定技术实现方案
2. 任务分解
   * 开发人员将需求功能点进行拆分并做估算，拆分完的任务记录在JIRA看板
   * 测试人员需要评估测试用例和测试时间的时间，拆分完的任务记录在JIRA看板
   * 根据优先级和人员安排给出各个功能点预计交付时间
   * 开发团队只评估未来两周能够完成的功能点，其他功能点全部放入产品Backlog

**解决问题：**

* 明确需求（做什么），包括业务类和技术类
* 明确未来两周DEV & QA的具体任务
* DEV & QA对于可交付产物的承诺（可交付指完成测试）

1. **迭代周期（功能开发 & 测试）：**

**输入：**

* 本次迭代需要完成功能点的PRD文档
* 本次迭代需要完成的技术改造类任务
* 技术实现方案（可选）
* 本次迭代开发任务
* 本次迭代测试任务
* 产品Backlog（产品团队在迭代周期中可随时调整下次迭代的Backlog）

**输出：**

* 可交付产物
* 测试用例文档
* Release note
* CR清单

1. 编码 & 测试用例编写及评审

* 开发人员编码阶段，测试人员需要编写该功能的测试用例。当测试用例编写完成后，双方可线下进行测试用例评审。开发及测试人员任务以看板上任务为准，任何的变动第一时间通知项目经理
* 解决问题：
* 开发、测试任务透明，任务追踪以看板为准，团队所有成员清楚任务的进展
* 开发、测试任务同步进行，避免串行工作带来的问题

1. Code review（可选）

* 解决问题：
* 为保证高质量的产品，开发经理决定是否需要进行code review。

1. 变更

变更包括以下几种：

1. 需求变更：产品经理对原有需求进行修改，召集团队所有成员，紧急商讨技术解决方案、评估工作量、决定是加班还是删减相应工作量的原任务，相当于开一次plan会，所有紧急需求，项目经理以CR的形式记录
2. 紧急需求：由业务、产品决定紧急需求是否要加入本次迭代，若需加入则召集团队所有成员，紧急商讨技术解决方案、评估工作量、决定是加班还是删减相应工作量的原任务，相当于开一次plan会，所有紧急需求，项目经理以CR的形式记录
3. 突发问题：如网络无法使用、员工请假等，项目经理进行协调，并及时调整原计划
4. 线上故障：线上发现的故障，以bug类型记录在JIRA上，由产品经理决定是否要加入本次迭代中修改，若需要，做法参照紧急需求
5. 生产事件：第一时间优先解决生产事件，事件解决后，项目经理根据影响的工作量，试情况决定是否要召集团队成员重新评估两周的承诺

所有技术支持事件，全部归为变更处理，由项目经理（技术支持）跟进。

* 解决问题：
* 尽量避免任何可能影响DEV & QA团队两周开发测试任务的情况。
* 非P0需求or线上问题不应加入到本次已经开始的迭代中

1. 功能Pre-review

对于编码已完成，进入测试阶段的功能，产品团队应尽早进行pre-review，并提出修改意见，对于工作量大的修改点，参照紧急需求处理。

* 解决问题：
* 产品经理、UI、UE可以根据实际效果，尽早修改
* DEV未按照PRD需求的实现，尽早发现
* 产品经理作为第一道测试进行把关

1. 每日站会（可选）

* 团队每日同步更新各自JIRA看板上的任务状态，每位组员只讲以下三点：
* 昨天我做了哪些任务
* 今天我要做哪些任务
* 开发|测试中是否有问题（产品需求问题，需要产品团队介入）
* 解决问题：
* 避免DEV & QA任务进度不透明化。
* 任务进度及时更新，避免问题积压。
* 保证团队所有成员信息同步。

对于不愿意召开站立会议的团队，项目经理负责跟团队成员沟通并每天更新进度

1. 集成测试 & UAT测试

* 测试团队在发布前进行统一的集成测试，产品团队进行UAT测试。
* 解决问题：
* 避免上线后出现严重问题

1. **发布计划**

**输入：**

* 可交付产物

**输出：**

* 发布计划

1. 针对app在每次迭代前，项目经理排出发布计划。
2. 针对非周期性发布应用（api、service、web、H5等），提前3天做好发布计划，并通知相关团队
3. 发布计划应当至少包括两部分：线上监控&回滚方案
4. 发布计划要通知到所有相关团队

**解决问题：**

* 所有发布透明化，特别是会有跨团队影响的发布，及时通知并评估影响
* 系统发布后，相关人员进行监控，监控项需在发布计划中明确
* 系统发布出现问题，第一时间根据回滚方案进行回滚。

1. **迭代周期回顾会议（可选）：**

**输入：**

* 本次迭代可交付产物
* 测试报告

**输出：**

* 下个迭代中可改进的措施 & 方案
* 本次迭代中出现的问题

**解决问题：**

* 持续改进，提高团队协作、技能及产品质量。

**流程图：**

