## 《并行项目推进和攻关类项目立意》

——2021年Q1小项目线分享

本季度完成的小项目主要有：X-Bond搭桥攻关项目、三方回购项目、请求报价优化项目，完成Q1基本指标。

### 一、并行项目推进

#### (一) 背景介绍

申报0.2的项目时间跨度需要3周，第一季度无法完成基本指标要求，衡量评估人员和工作分配后，能够同时再申报一个项目，于是报了同期开展的请求报价&买方机构交易项目，业务需求谈了2天后发现现有业务方案难以立项成形，于是与版本负责人沟通未继续买方的项目准备，转而到三方回购项目准备。因此，有了这次机会同时进行几个项目准备和实施，此处场景不能与大规模并行项目同日而语，主要侧重时间安排和事情优先级安排，**目标**是在此情况下能保证项目的顺利结项。

最高峰：3个项目立项和结项准备，包括请求报价、买方机构、X-Bond明日搭桥。

#### 分享内容

1. 前期准备和立项阶段

明确项目的目标和期望，优先级多维度排序，投入相应精力和资源。明确**关键时间结点**，尤其在项目起始阶段，使每位组内成员明确知道自己的任务范围和内容，以及每个时间点需要达到的目标，还有了解每位成员当前的状态，比如是否要休假、是否有困难等预测人员风险，评估业务复杂或技术复杂点预测实施风险，外部依赖沟通点预测对外沟通风险。针对本季度的实践，举例以下几方面：

1. 人员管理。针对较新的人员（不止是进新本币的时间，也包括对于接手内容的熟悉程度），需要加强小项目规则、工作内容的沟通，并确保对方已明白自己在项目内部的工作任务和使命。如遇到人员不熟的情况时可**多方补位**，例如项目中BA接手了之前从未接触过的大量需求，组内各岗位成员随时补位串联需求内容，纠正不准确的点，认知达成共识，确保各成员能力范围和时间范围内可以实现。
2. 工具使用。除了正常使用redmine、confluence等公用工具外，项目内部管理使用甘特图，在每日进度跟踪以及项目复盘时较为一目了然，每位成员也能方便清晰可知自己的工作对于整个项目的进展影响。



1. 风险管理。预测风险点，理清依赖关系，平衡各方要素，在可控范围内投入更多资源。例如在三方回购项目中，前端人员开发完成任务速度较快，当测试缺陷量增大的时候，如遇到不明显的前后端缺陷，首先分给前端人员排查推进；CDC接口定义存在歧义，需要与CDC做好沟通工作消除歧义并留存信息。
2. 需求设计阶段

把连片、效率高的时间留给处于需求和设计阶段的项目。因为设计阶段思路连贯性很重要，被打断之后再重新捡起来比较困难，另外这个阶段能够比较集中精力发现细节疑点待简化点，可以及时沟通调整和简化需求。往往部分开发接到新需求时，听第一遍讲解未能想的很细节，需要花时间整理思路，避免在后期才想到难处，与业务沟通更为被动。例如在三方回购项目里，前后经历了2天需求评审，在第3天业务提出了新的需求变更，此时项目组也想清楚实现复杂的点，于是和业务交换了变更和简化的需求。如果到了后期实施时再想到难杂点，与业务解释起来则更为被动。同样，业务在项目时间过半后提出了影响较大的变更时，也需要耐心沟通“此事做不完”的细节原因。业务老师在现有项目工作量的制约下，也能理解项目组的难处。

1. 项目日常推进阶段

1）每天固定的时间来关注缺陷，尽可能选择有些疲劳容易被打断的时间来做这事，比如下午4点之后。上午时间，讨论重点的问题，或者需求较为复杂的场景，不影响开发实施进度。

2）多鼓励和激励组员，以事实和结果为导向（功能确实实现好了），相信组员（确认任务和时间是得到对方肯定的答复），更能提高积极性。对于未能完成好本质工作的，提出质疑批评同时帮助其找到问题点，沟通下一次的目标。

3）推进“传球”速度，接到项目后，把任务交给相应岗位人员后即小项目负责人已完成“传球”，接下来即关注拿球者的速度，避免一个人拿球时间过长。

### 攻关类项目立意

首次做攻关类项目，得知此类项目与业务类项目的立意点有区别，需要说明的是“为什么需要攻关”。

对于业务类项目或者技术类项目立意相对明确，如果能实现业务需求，或者解决技术优化目标即可。攻关类立项的立意点则更关注性能细节和对于未来业务数据预测，出发点也来自于业务提出的新需求“概念”，但对于系统的实际落地影响未知，因此需要更多的系统间比对、性能测试、数据报告论证说明“为什么这么实现可行”。

翻看了之前攻关类的项目材料，推荐“X-bond私有行情优化总结”（confluence可见），此项目历时2个多月，对于性能细节的描述也比较细致和专业。

因此结合来讲，攻关类的项目需要以下内容：

1. 选取其他系统需求和实现，对比优劣和差异。包括需求立意差异点分析、技术方案和实现差异分析等。
2. 基于生产数据量级评估各环节耗能，论证可行性。粒度达到链路耗时分析、数据结构的选择。
3. 业务数据预测。评估各个环节的性能测试，找到性能瓶颈，预测业务数据上限，最后以此结果跟业务协商具体的数据限制。

