**链上各类活动模版**

代币项目方的基本需求

1. Hold代币数量
2. Holders数量
3. Volume

**Neo模型分析**

拆解Neo的Bond模型，有以下要素

1. 7天存款期，存款期之后不可存款
2. 持有30天，中间不可提现
3. 30天后自动退回本金+奖励
4. 所有人不是同时开奖，早存就会早退
5. 有募资上限，最高500w
6. 价格补偿机制，低于5U的价格会补偿

尝试分析一下，以上这些要素，想要基于现有的Quest模型，如何改进才能配置出来

1. 第一条，Quest目前只有活动时间这一个概念，套用在这个场景，就是7天的活动参与时间，这一点无需改进；
2. 第二条有两种做法，这两种都需要开发新模版
3. Hold Token只能验证持有，但不能帮项目方实现锁仓的效果
4. Stake可以使用现有的Deposit功能变相实现
5. 第三条，Quest需要增加一种延迟xx天发奖的奖励规则
6. 第四条，Quest需要在延迟的基础上，还增加每个人发奖时间不同的规则，但这一条的必要性存疑
7. 第五条，需要增加新的Quest结束条件，募资满额就结束了
8. 第六条，需要增加Quest的保证金机制，符合某种条件会触发提前发奖或奖励补偿

整体评估下来，对Quest的改动很大，包括如下内容：

1. 增加延迟发奖机制
2. 增加新的Quest结束条件，募资满额
3. 增加Quest保证金机制，项目方可以预存保证金
4. 增加一种Stake模版

我感觉，以上这些改动的通用性都不高，而且很多改动机制都是针对1个专属task的，如果考虑通用化设计需要额外的维护大量无用分支，还可以考虑三种方式来实现这个场景：

1. 增加一种community task，问题是task本身的复杂度不足以承载太复杂的场景，而且没有专属的活动页面，不利于传播；
2. 增加一种新产物，比如只能设置一个task的轻量级Quest，或者是复杂版task；
3. 增加一种Quest模板，模板固定参数，只有在这个模板下才能使用这个Stake新模板；

我认为第三种是相对来说扩展性较好的方案，后续还要进一步与开发讨论。

**运营活动分析**

1. API验证的交易大赛
2. 必须使用指定钱包的交易大赛
3. 要求用户连接指定钱包
4. 开盲盒玩法

**交易的基础设施**

1. **OO标准**