# DeFi 核心项目面试深度问题补充设计

## 一、借贷核心智能合约开发模块（补充）

### 1. 场景化压力题

* 若上线后发现抵押品清算模块存在 “清算价格计算偏差” 漏洞（如未考虑区块 Gas 费对实际清算收益的影响），导致部分清算交易亏损，你会如何紧急处理？是否会暂停合约功能？修复后如何补偿受影响的清算参与者？
* 某用户通过 “重入攻击” 尝试窃取资金池资产，虽被合约安全机制拦截，但引发社区恐慌，你需要向用户和社区解释攻击防御原理，如何用非技术语言清晰说明，同时稳定社区信心？

### 2. 项目复盘类

* 回顾整个合约开发过程，你认为最关键的技术决策是什么？（如模块化拆分方式、安全审计工具组合等）若重新开发，哪些环节可以优化？（如更早引入第三方审计、增加模拟测试场景等）
* 减少 15% 合约调用成本的目标达成后，后续是否有持续优化空间？（如适配 EIP-4844 降低存储成本、引入账户抽象简化用户操作）目前已识别的优化方向有哪些？

## 二、多源价格预言机系统模块（补充）

### 1. 场景化压力题

* 某主流资产（如 BTC）出现 “极端行情”（1 小时内涨跌超 20%），Chainlink 数据更新延迟，Uniswap V3 因流动性不足价格卡顿，你的预言机系统输出价格与市场真实价格偏差扩大，此时是否需要手动干预？手动干预的触发条件和操作流程是什么？
* 黑客通过 “女巫攻击” 控制多个 Uniswap V3 流动性池，试图操纵价格，你的异常数据校验机制能否识别？若识别后暂时排除 Uniswap V3，仅依赖 Chainlink 数据，如何应对 Chainlink 单点风险？

### 2. 项目复盘类

* 系统上线初期，是否出现过 “数据源权重分配不合理” 导致的价格偏差问题？（如过度依赖某一数据源）如何通过历史数据复盘调整权重模型？调整后数据准确性提升了多少？
* 对比行业内其他预言机方案（如 Pyth、UMA），你的系统在 “数据可靠性” 和 “Gas 成本” 上有哪些优势和不足？后续是否有计划融合其他方案的优势（如 Pyth 的低延迟特性）？

## 三、高可用自动化清算机器人模块（补充）

### 1. 场景化压力题

* 区块链网络出现 “长时间拥堵”（如 ETH Gas 价格突破 1000 Gwei），清算机器人批量交易因 Gas 费用过高无法发起，大量用户抵押率持续低于安全阈值，此时是否允许 “降低清算收益” 以吸引清算者？调整清算收益的决策流程是什么？
* 某恶意用户利用 “清算机器人监控盲区”（如通过特殊交易时序规避抵押率计算），拖延清算时间，你会如何优化监控逻辑，覆盖此类特殊场景？

### 2. 项目复盘类

* 机器人运行过程中，是否出现过 “误清算” 情况？（如因价格数据延迟导致抵押率误判）误清算的原因是什么？如何通过技术优化（如增加价格确认次数、延长判断周期）降低误清算概率？
* 与行业内其他清算方案（如 Aave 的清算机制、Compound 的清算奖励模型）相比，你的机器人在 “清算成功率” 和 “操作成本” 上的表现如何？是否有数据支撑（如清算成功率 95% 以上、单笔清算成本降低 30%）？

## 四、Subgraph 数据索引服务模块（补充）

### 1. 场景化压力题

* 某热门活动期间（如协议流动性挖矿启动），用户交易量大增，Subgraph 索引速度跟不上链上数据产生速度，导致前端显示 “数据滞后”（如用户存款后未及时更新余额），此时如何紧急扩容？扩容过程中是否会影响现有查询服务？
* 前端团队反馈 “某类聚合查询”（如用户跨时间段借款总额统计）响应速度突然变慢，排查后发现是 Subgraph 索引规则存在 “低效查询逻辑”，如何在不中断服务的情况下修改索引规则？

### 2. 项目复盘类

* 设计初期是否低估了 “清算事件” 的索引复杂度？（如清算涉及多笔关联交易，需关联用户、抵押品、价格等多维度数据）如何通过重构数据实体关系提升索引效率？重构后查询速度提升了多少？
* 目前 Subgraph 数据存储成本占协议整体运营成本的比例是多少？是否有优化存储成本的方案？（如定期归档历史数据、采用压缩存储格式）已落地的方案节省了多少成本？

## 五、多层次风控体系模块（补充）

### 1. 场景化压力题

* 协议遭遇 “大规模漏洞攻击”，紧急暂停机制触发后，发现部分用户资产因 “合约锁定逻辑” 无法赎回，此时是否需要启动 “紧急提币通道”？紧急提币通道的技术实现方式和风险控制措施是什么？
* 某抵押品（如某 ERC20 代币）因项目方问题被认定为 “高风险资产”，需要紧急下调抵押品因子（如从 0.7 降至 0.3），大量用户面临 “补仓通知”，如何平衡 “风险防控” 和 “用户体验”？补仓通知的发送时机和频次如何设定？

### 2. 项目复盘类

* 漏洞赏金计划启动以来，共收到多少漏洞报告？其中有效漏洞（如 Critical、High 级别）占比多少？最严重的漏洞是什么？修复后如何验证漏洞已完全解决（如通过模糊测试、模拟攻击）？
* 回顾历次 “风险事件”（如市场波动、数据源异常），你的风控体系在 “响应速度” 和 “应对效果” 上有哪些可改进之处？（如缩短紧急暂停触发时间、优化抵押品因子调整阈值）已制定的改进计划是什么？

（注：文档部分内容可能由 AI 生成）