**Review Report**

* What：Meet
* 从前端开发到全栈工程师的转型（‘达到入门水平’，meet）
* 通过从后端的技术，突破前端技术瓶颈（meet）
* 努力完成项目所要求的任务（meet），主动承担任务（meet），达到leader的转正要求（‘期望成长为项目组长’not meet），

推进项目进度，按时交付（meet），关于项目组内的工作总体来说是meet -

* 找到归属感，建立圈子，适应氛围（exceed）
* 以个人的项目经验进行前端项目重构，已达到尽可能小的成本下，完成另外一个产品应用（白小条），完成情况（meet）

* How：Meet
* Self
* 根据优先级进行工作安排
* 把工作的内容拆分，比如：业务逻辑，代码实现，模型结构
* 遇到的坑，自己或找同事协助解决后，记录到为知笔记，

以便后面可以快速定位问题，毕竟人的记忆力并不是那么可靠

* 个人刚进入公司，后端技术背景比较空白，通过对项目代码的阅读，针对现公司用的技术框架，针对性的私底

下花时间研究，然后减少在工作中要去找资料耗掉的时间，始终按时交付是原则性问题

* 不会的问题主动提问，不明的需求提出质疑，不会的技术敢于请教
* Team
* 我们组内会代码review，保证代码质量，以减少后期代码维护成本
* 我会在做项目之前，了解清楚项目背景，业务逻辑，以及组内对了解到的需求的理解是否一致，然后找产品去确认
* 同步信息，定时把握好进度，有任何风险点第一时间跟产品沟通，以便提前找到应对的措施，也好配合产品工作，与其他渠道进行沟通，减少不必要的麻烦与工作上互相扯皮的事情
* Person
* Strength
* a 项目经验比较丰富，接触过的技术框架比较多（偏前端）
* b 理解能力
* c 大局观
* d 沟通能力
* f 能量，抗压，接受能力
* Weakness（劣势）
* a 后端技术基础比较薄弱，接触到的框架比较少
* b 身体最近小毛病比较多
* c 偏肥不够帅
* Blocks and issues
* 信息同步这方面我个人一直有在保持，这也是我在哪里都那样做，不过发现信息不同步这个问题，好像在很多地方是通病，比如说：
* 第一种情况，有些功能修改，有可能导致其他地方需要修改，但是修改人没有提出，或者应该要有个意识去考虑会不会影响到其他哪些功能，然后去提问。
* 第二种情况，两个开发在开发分别负责前后端部分，一开始定了一个方案，后面有一端做修改了，没有同步通知信息，导致另外一部分后面测试的时候，花了更多的时间去排查问题，或影响了线上的功能都不知道，待客服反馈用户遇到的bug才知道，然后这种不必要的恶劣体验应该让用户避免
* 第三种情况，产品的信息同步不够及时，开完会，有更新的进展的时候，或者其他功能点要修改的时候，会遗留了跟开发去沟通，或者产品会先偏向先做好文档，做好归纳，再去跟开发过需求，但是这里有个很严重的问题就是，原本的需求如果被影响，会改变的话，应该第一时间跟开发沟通，然后避免时间成本的浪费。
* Action Items（针对劣势，提出改善方法）
* action 1 自己私底下做了个人blog的项目，全后端来实现，页面也是用的后端渲染模板的方式来进行，用的技术框架模仿现在公司的项目，以此提高工作效率，技术的掌握，对项目尽快上手，也有定期给leader进行review
* action 2 b,c点同样的解决方法，运动，减肥，吃少点
* Suggestions（开放式建议，任何方向都可以）
* action 1 care about "Block and issues" ，这个模块的问题是项目的建议
* action 2 转正的员工能否免费参加年度旅游（毕竟新加入的同事对跟大家一起出去玩的这个还是比较向往的，但是要请假然后要自费的成本相对比较高）
* action 3 可以尝试做一个养考拉，这个在链养猫火起来的时候我就有关注，后面加入公司，那时候大家对区块链的关注没那么高就没有提出过，最近公司的同事都不多不少也算是有去了解区块链了，所以提出养考拉这个sugget，大家也会比较容易知道是怎么一回事，毕竟我觉得公司不能一条腿走路，梭哈考拉理财的话，风险很高。