**开发团队年度绩效评估表**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | | 谢果龙 | | 岗位 | | 高级软件工程师 | | 评估时间 | | 2016/12/24 |
| 一、计划关键任务记录 | | | | | | | | | | |
| 序号 | 职责分类 | | 关键任务描述 | | 其他参与人员 | | 计划完成时间 | | 计划滞后比例 | 完成结果（是否有质量问题） |
| 1 | 开发工程师 | | Logic monitor 项目中的Router rules功能 | | Jason,Linda,Koma,Damon | | 2天 | |  | 无 |
| 2 | 开发工程师 | | 菜单结构梳理中的权限展示、编辑、验证 | | Koma,Damon,Mark | | 12天 | | 12% | 因为测试资源紧张，比原计划延后1.5天 |
| 3 | 前端开发 | | Report 项目第一阶段 | | Koma, Damon, Mark | | 2016-10-12 | |  | 无 |
| 4 | 前端开发 | | Report 项目第二阶段 | | Daniel, Damon, Mark | | 2016-10-28 | |  | 无 |
| 5 | 前端开发 | | Report 项目第三阶段 | | Jason, Damon, Lynn, Mark | | 2016-12-14 | |  | 无 |
| 6 |  | |  | |  | |  | |  |  |
| 二、2016年度工作小结（包括1、本年度主要任务与成绩；2、主要存在的问题、缺点及改进计划）   1. 本年度主要任务与成绩    1. 通过培训以及后期的实践，充分了解公司产品以及公司文化，并且较好的融入到了公司团队。    2. 参与logic monitor, 菜单改进项目中的部分功能开发，了解公司项目开发的基本流程，并保质保量的完成了开发任务。    3. Report 项目的第一、二负责除Availability 报表外的前端框架搭建、主要公共控件设计及实现、页面功能实现，保证框架大部分代码可复用，公共控件完全可复用，较好的实现了所有页面功能，第二阶段为充分了解后台代码，负责了Rating 报表的后台开发，并达到了目的。    4. Report 项目第三阶段与杭州同事合作开发完成。 2. 主要存在的问题    1. Report 项目前端虽然比较完整的实行所有功能，但是仍然可以进一步优化，如可以进一步将所有功能模块化，在不同的页面调用不同模块实现页面功能。导致这个问题主要原因是开始开发前没有充分研究所有报表的结构特点以及前端开发经验不足。    2. 在Report项目第三阶段Bug 修复过程中暴露的不够细心问题，过于依赖测试的反馈结果，解决表面问题，未能充分全面考虑与bug相关的其他可能性，把量放在了第一位，忽略了质的重要性。    3. 从Stanley等有丰富前端开发经验的同事的code review的结果看出，我离以最规范简洁的代码高效的实现功能要求的目标还存在一定距离。 3. 缺点及改进计划   缺点：   * 1. 开发前项目研究不够充分。   2. 不够细心的问题。   3. 代码简洁度不高。   改进计划   1. 在开发前做好充分的需求分析，做到开发前在纸上有一套粗略的但是完整的设计稿，写代码前验证改进该设计稿，直到有一个最优设计。 2. 不够细心主要体现在Bug修复阶段，给自己订立要求，简单bug 一次修复，复杂bug的不超过三次。对每个bug都安排充足时间进行验证，不因为bug简单而掉以轻心，先考虑质再考虑量。 3. 前端代码简洁性需要通过长期的实践和学习来提高，现阶段参加VM的开发是一个比较好的机会，ESLint 在项目中的应用，能够比较高效的指出我代码的规范性问题，我将通过做到开发的代码中不遗留任何ESLint 提示问题作为代码规范要求，并且通过学习其他有更多前端开发经验的同事的编码风格，来提高自己。 | | | | | | | | | | |
| 三、2017年工作目标与计划   * 1. 重视并改正修正Bug不够细心的问题，质量就是效率，以提高质量的途径来提高团队整体效率，做到Bug不重开，沟通一次到位。   2. 在完成所有公司分配的任务的同时，以Stanley，Roy等为目标，提高自己开发水平，做到熟练使用redux 和react 框架。   3. 阅读至少五本关于架构方面的书籍，并学以致用。   4. 提高临场发挥能力，做到条理清晰，目标明确。 | | | | | | | | | | |
| 四、对工作、团队、公司发展、现有制度与管理等方面的意见或建议；  公司会议有点多，建议简化。 | | | | | | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 五、绩效评价（参照指标描述，在分数栏中打分，并按其分值相加计算出价评结果。） | | | | | |
| 评价项目 | 评价指标 | 指标描述 | | 自我评价 | 主管评价 |
| 工  作  完  成  质  量  40分 | 代码格式  （10分） | 代码非常清晰，简洁易懂。（8－10） | | 8 |  |
| 代码比较清晰，可读性尚可。（4－7） | |
| 代码条理不清，可读性较差。（1－3） | |
| 代码  逻辑性  （10分） | 代码具有逻辑性，流程最简化，依赖性小。（8－10） | | 8 |  |
| 代码逻辑性尚可，但有一定依赖性。（4－7） | |
| 代码逻辑不强，可能存在维护性风险。（1－3） | |
| 代码  可用性  （10分） | 代码实现合理，效率较高，可保证软件运行。（8－10） | | 8 |  |
| 代码实现基本合理，软件运行速度一般，。（4－7） | |
| 代码实现不太合理，效率较低，影响软件运行。（1－3） | |
| 代码  复用性  （10分） | 代码可多次运用，复用性优秀（8－10） | | 8 |  |
| 代码只能少量运用，复用性一般。（4－7） | |
| 代码完全不能复用。（1－3） | |
| 工  作  难  度  20分 | 项目  重要性  （10分） | 大型项目，项目优先级高。（8－10） | | 8 |  |
| 一般项目，项目重要程度一般。（4－7） | |
| 小型项目，如Bug修复之类。（1－3） | |
| 专业  复杂度  （5分） | 项目非常复杂，有很大的挑战。（5） | |  |  |
| 项目难度一般，有一定的挑战性。（3－4） | | 4 |  |
| 小型项目，无难度。（1－2） | |  |  |
| 工作  创新度  （5分） | 具有创新思维，不断思考优化开发工作。（5） | |  |  |
| 有一定的创新，可以提出一些创新性建议（3－4） | | 3 |  |
| 按部就班，较少思考创新。（1－2） | |  |  |
| 工作量  10分 | | 工作量很饱满，在完成开发任务的同时，还承担了项目之外的工作。（8－10） | |  |  |
| 工作量饱满，基本完成项目开发任务。（4－7） | | 7 |  |
| 工作量不饱满，尚有空闲。（1－3） | |  |  |
| 工  作  态  度  30分 | 主动高效  （10分） | 独立提出切实可行的改进方案，并推进实施，取得良好的成效。（9－10） | |  |  |
| 工作中主动发现问题，提出有价值的改进建议。（7－8） | | 7 |  |
| 调动各方面资源以达成目标。（4－6） | |  |  |
| 被动执行安排的工作，遇到困难被动等待，对工作中问题视而不见。（1－3） | |  |  |
| 团队协作  （10分） | 协助对方获取成功，在达成团队整体目标的同时实现个人目标。（9－10） | |  |  |
| 认真听取对方意见，清楚表达自己意见，并且提出有价值的建议。（7－8） | | 8 |  |
| 与团队成员良好沟通，保证项目按期进行。（4－6） | |  |  |
| 不与团队成员沟通，完全按照个人设想工作。（1－3） | |  |  |
| 学习分享  （10分） | 主动学习，工作技能明显提高，分享并帮助团队成员进步。（9－10） | |  |  |
| 有意识地学习知识技能和业界先进经验，并在工作中实践，乐于分享。（7－8） | | 7 |  |
| 能够参加公司培训，在某些方面提高自己，有一定的分享。（4－6） | |  |  |
| 缺乏自我培养和提高意识，随大流。（1－3） | |  |  |
| 自评结果： 76 | | | 主管评分： | | |
| 主管反馈：包括继续发挥的优点和下一步努力方向。  （若评价指标自评与主管评分差距较大，建议在此列出具体的案例来说明问题，以便于后续沟通。）  主管签名： | | | | | |