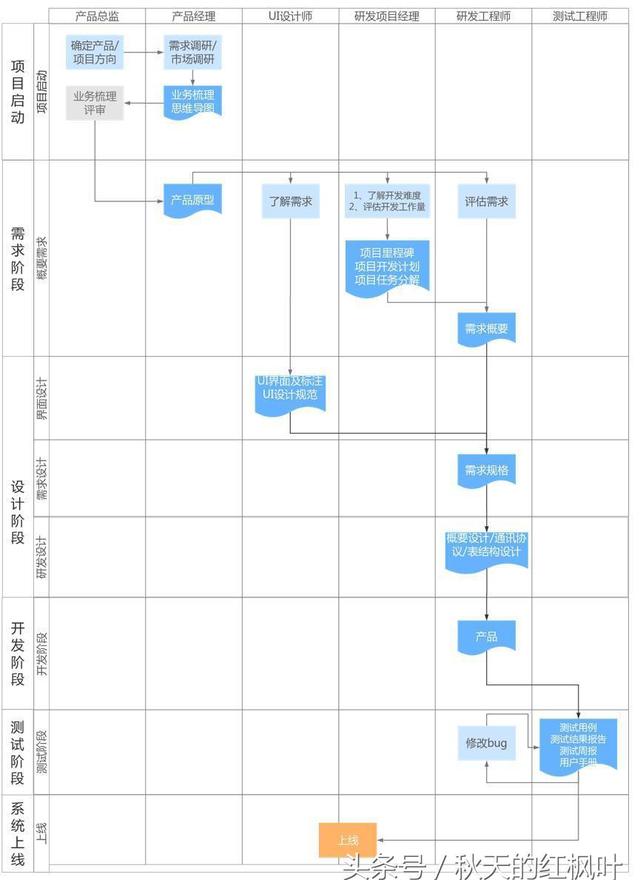
**软件开发基本流程**

## 一、开发流程图



## 二、过程产物及要求

**本表主要列出开发阶段需要输出的过程产物，包括产物名称、成果描述、负责人及备注，即谁、在什么时间、应该提供什么内容、提供内容的基本方向和形式是什么。**



## 过程说明

**（一）项目启动**

1、产品经理和项目干系人确定项目方向，产品型项目的干系人包括公司领导、产品总监、技术总监等，项目的话则包括客户方领导、主要执行人等。

2、公司领导确认项目组团队组成，包括产品经理、研发项目经理、研发工程师、测试团队等。

3、明确项目管理制度，每个阶段的成果产物需要进行相应的评审，评审有相应的《会议纪要》；从项目启动起，研发项目经理每周提供《项目研发周报》；测试阶段，测试工程师每周提供《项目测试周报》。

4、产品经理进行需求调研，输出《需求调研》文档。需求调研的方式主要有背景资料调查和访谈。

5、产品经理完成《业务梳理》。首先，明确每个项目的目标；其次，梳理项目涉及的角色；再来，每个角色要进行的事项；最后，再梳理整个系统分哪些端口，要有哪些业务模块，每个模块再包含哪些功能。

**（二）需求阶段**

1、进入可视化产物的输出阶段，产品经理提供最简单也最接近成品的《产品原型》，线框图形式即可。在这个过程中还可能产生的包括业务流程图和页面跳转流程图。业务流程图侧重在不同节点不同角色所进行的操作，页面跳转流程图主要指不同界面间的跳转关系。

2、产品经理面向整个团队，进行需求的讲解。

3、研发项目经理根据需求及项目要求，明确《项目里程碑》。根据项目里程表，完成《产品开发计划》，明确详细阶段的时间点，最后根据开发计划，进行《项目任务分解》，完成项目的分工。

4、研发工程师按照各自的分工，进入概要需求阶段。《概要需求》旨在让研发工程师初步理解业务，评估技术可行性。

**（三）设计阶段**

1、UI设计师根据产品的原型，输出《界面效果图》，并提供界面的标注，最后根据主要的界面，提供一套《UI设计规范》。UI设计规范主要是明确常用界面形式尺寸等，方便研发快速开发。UI设计常涵盖交互的内容。

2、研发工程师在界面效果图，输出《需求规格》，需求规格应包含最终要实现的内容的一切要素。

3、研发工程师完成《概要设计》、《通讯协议》及《表结构设计》，及完成正式编码前的一系列研发设计工作。

一、需求描述

二、概要设计

三、数据库表设计

四、业务流程设计

五、接口设计

**（四）开发阶段**

1、研发工程师正式进入编码阶段，这个过程虽然大部分时间用来写代码，但是可能还需要进行技术预研、进行需求确认。

2、编码过程一般还需进行服务端和移动端的联调等。

3、完成编码后需要进行功能评审。

**1 遵守公司级别的编码规范 ：**[Java编码规范](http://wiki.corp.yunnex.com/pages/viewpage.action?pageId=7405872)

**2 涉及密码的需要对称加密，密文存储**  
对应的类 AESUtil.encrypt和AESUtil.decrypt  
  
**3 XXXRequestDto和XXXResponseDto格式**  
入参和出参按照这个格式返回，模块的公共DTO可以共用  
比如经营诊断模块   
DiagnosisRequestDto extends BaseDto (模块的请求DTO)  
DiagnosisResponseDto extends BaseDto (模块的响应DTO)  
PropagandaKeyRequestDto extends DiagnosisRequestDto (具体业务的请求DTO)  
PropagandaKeyResponseDto extends DiagnosisResponseDto (具体业务的响应DTO)  
  
**4 运行时异常统一抛出 ServiceException**  
com.yunnex.ops.erp.common.service.ServiceException  
不能抛出RuntimeException  
系统统一拦截ServiceException，跳转到500页面，已提供

**5 物理删除的需要打印日志信息，方便跟踪痕迹，日志使用占位符格式**   
外围接口不管是拉还是推，入口和出口需要打印日志参数，一些核心的业务逻辑需要打印日志信息  
logger.info("syncShopPassword->同步密码的商户：shopId = {}", shopId);  
  
**6 前端和后端一些常识性的隐式需求参数做需要校验（数字，长度等）**比如电话，手机，邮箱

**7 多表连接查询，关联的表字段建立索引，看一下sql的执行计划，避免出现全表扫描**

**8 对于判断存在id在修改，不存在则新增统一用CrudService.save(T)方法，不要用dao判断。**

**9 service只能调用本模块的dao,service之间可以跨模块调用，不允许模块A的service调用模块B的dao，持续进行**

**10 尽量考虑代码性能，禁止在循环里面调用sql语句**

**11 后台开发针对核心功能，复杂功能需要提供 单元测试代码。**普通的增删改查可以不用提供，已经提供事物回滚机制。

**12 关于常量类，三层查找，先从项目级查找，再从模块常量，如果是类私有的 类常量。**  
项目常量类：CommonConstants  
模块常量类：举例DiagnosisConstant  
类常量  
  
**13 手动sql需要update表，设置mybatis拦截器设置preUpdate修改最近联系人和最后修改秒时间，已提供**

**14 设计数据库每个字段不能为空，给默认值**

**15 数据库设计字段给出含义明显的词，不能魔鬼数字**

比如待审核（Reviewing），完成评审（Review Completed），拒绝（Review Refused），取消 （Cancelled）  
不能用0,1,2,3,4 标识

**16 写脚本不能用replace(uuid(), 生产库不支持**

insert into erp\_order\_split\_good(id,original\_split\_id,original\_good\_id,good\_name,good\_type\_id,good\_type\_name,num,price,create\_by,create\_date,update\_by,update\_date) select replace(uuid(), '-', ''),id,original\_good\_id,good\_name,good\_type\_id,good\_type\_name,num,price,create\_by,create\_date,update\_by,update\_date from erp\_order\_split\_info ;

**17 重点关注的地方**  
@健壮性：功能准确前提下，是否考虑兼容空指针/数据异常/违法检查/边界值？  
@稳定性：模块之间是否松耦合？代码在各种环境各种操作下是否保证运行正确？  
@性能性：如果数据量很大是否能够支撑，并发性高是否考虑？多表连接的索引是否建立？是否可以改变方案使性能最大化？  
@安全性：接口调用是否有身份校验？敏感数据是否加密？  
@可扩展：如果需求改变了或者新增了是否好扩展，还是要全部推翻重来？  
@可复用：系统是否存在这样的工具类？会不会多个方法出现一样的代码？  
@可维护：代码是否按照编码规范编写？核心代码注释是否齐全？设计是否满足开闭原则？  
@可配置：是否能够动态配置，改变值可以不用修改代码？  
@高内聚低耦合：一个模块之内明确完成一个功能实现高内聚，模块之间要低耦合，模块下的类之间要多用组合少用继承。

**（五）测试阶段**

1、测试工程师按阶段设计《测试实例》，未通过的流程测试提交至Jira，分配给相应的开发人员调整。

2、研发工程师根据测试结果修改代码，完成后提交测试，测试通过后完成。

3、测试工程师编写《测试结果报告》，包括功能测试结果、压力测试结果等。

4、测试工程师编写系统各端口的《操作手册》、维护手册等。

**（六）系统上线**

与客户或者上级达成一致后，系统进行试运行，稳定后上线。

以上内容是他人见解，不一定绝对正确，但是有借鉴意义。​

程序开发的基本步骤为分析、设计、编码、编译连接、运行（函数链接到目标程序）、测试调试、维护修改。作为技术开发人员，主要参与的是设计阶段与开发阶段。​

## 四、贝蚁过程说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **活动编号** | **活动名称** | **活动内容** | **工期** | **参与人** |
| 1 | 需求分析 | 产品给出版本链接总入口，研发人员和测试人员第一时间熟悉下个迭代的需求，备注中记录疑问 ，对需求有第一印象的理解。 | 1~2天不等，非全职 | 研发，测试 |
| 2 | 需求评审 | 研发同事带着疑问和产品运营，测试人员三方评审需求，基于目前的业务体系分析新需求的可行性和实现方案，解决疑难需求，澄清有歧义的语句，中间环节测试和研发可随时提问。 | 2~4小时不等 | 研发，测试，产品，项目经理 |
| 3 | 需求讲解 | 研发人员经过1~2天对wiki上的需求进行详细分析和把控，把自己对需求的理解，做成的效果讲给运营确认，如果有理解不一致的地方当场和测试，产品讨论，把理解思路达成一致。 | 2~4小时不等 | 研发，测试，产品 |
| 4 | 概要设计 | 研发同事把概念模型转换成逻辑模型和物理模型，设计好实现思路，结合相关的业务场景用最好的设计模式，包含概要设计，接口设计，数据库设计，业务流程设计等工作。针对同一个功能每人可发表不同的意见，最后统一设计思路。 | 1~2天不等，非全职 | 研发 |
| 5 | 设计评审 | 经过1~2天的设计，对设计文档进行评审，找出设计方案可能错误的地方及时修正，找出疏漏的地方及时补上。同时共同制定相关的编码规范，会议完成后在wiki写下评审记录。 | 2~4小时不等 | 研发 |
| 6 | 代码开发 | 研发同事把设计转换成代码实现，在研发数据库执行sql脚本，研发对应的feature分支生成CRUD代码，上传业务代码，绘制流程图，期间有任何疑问及时抛出来，间歇性跟进研发进度，设定版本里程碑，按照里程碑识别风险，如果有技术风险第一时间给出方案，如果有进度风险需要加班补偿。 | 根据工作量资源情况， 2~4周不等 | 研发 |
| 7 | 代码评审 | 一般在转测前几天进行代码review,针对代码的基本编码规范和健壮性,稳定性,性能性,安全性,可扩展,可复用,可维护,可配置等多维度对代码进行走查，如果不符合基本规范及时修正，对模块之间的高内聚低耦合需要重视，避免大方向走偏，针对同一段代码每人可发表不同的意见，最后统一思路。结束后在wiki记录结果。检查和督促研发人员及时修正。 | 2~4小时不等 | 研发 |
| 8 | 联调自测 | 研发后期对评审的问题进行修复，对功能进行收尾，对复杂的逻辑需要提供单元测试用例并且有测试结果，涉及多系统配合统一在dev环境进行联调，对联调的问题及时修复，期间会对测试人员提供的 用例进行评审。 | 2~4天不等 | 研发 |
| 9 | 测试验收 | 首先提前一天从git获取版本对应的脚本变更在测试环境统一执行，部署测试包，研发同事有半天时间在测试环境测试自己相关的功能和主流程是否阻塞，如果有阻塞及时修复，第二天开放给测试人员使用，每天给出测试进度百分比，及时识别风险，晚上研发同事需要相应加班支持，测试阶段后期产品同事加入验收环节，及时找出问题进行修复。 | 根据工作量资源情况， 1~2周不等 | 研发，测试，产品 |
| 10 | 生产上线 | 测试人员把用例跑完，并且严重问题全部修复验证的情况下 提供提测&上线文档，并且对文档进行评审，代码合并到master和dev，检查 erp.properties配置项，执行脚本，配合运维同事进行版本升级，升级后部署流程图，补充菜单角色等数据，上线后进行。 | 当天 | 研发，测试，产品 |
| 11 | 上线支持 | 发版之后，测试人员需要建立测试商户，进行主流程的回归测试，有问题研发及时修正，产品同事对版本功能进行验收，对修复后的问题走bugfix部署。 | 1~2天 | 研发，测试，产品 |
| 12 | 问题分析 | 针对上个版本转测阶段造成的问题进行原因归类和分析，总结出措施，为了更好的控制 版本质量和尽可能减少不必要的问题。问题原因不排除代码缺陷，设计缺陷，历史问题，需求优化，小程序，操作问题，数据问题，环境问题，其他引发，修改引发等。 | 2~4小时不等 | 研发，测试 |
| 13 | 需求变更 | 产品同事提供jira跟踪单，指明wiki需求地址，研发根据需求涉及的功能和场景评估工作量，识别给出对本版本的转测时间变动，通知项目经理，产品，测试，给出新的转测时间，测试同事根据评估的测试时间给出新的上线时间。以此类推。 | 2~4小时不等 | 研发，测试，产品，项目经理 |