* [1 总体规则](http://pms.ipo.com/pages/viewpage.action?pageId=12164134#id-3.3.1.1项目管理需求及移交阶段-1总体规则)
* [2 GNU版本号定义及管理策略](http://pms.ipo.com/pages/viewpage.action?pageId=12164134#id-3.3.1.1项目管理需求及移交阶段-2GNU版本号定义及管理策略)
  + [2.1 GNU版本号定义](http://pms.ipo.com/pages/viewpage.action?pageId=12164134#id-3.3.1.1项目管理需求及移交阶段-2.1GNU版本号定义)
  + [2.2 GNU版本号管理策略](http://pms.ipo.com/pages/viewpage.action?pageId=12164134#id-3.3.1.1项目管理需求及移交阶段-2.2GNU版本号管理策略)
* [3 需求要求](http://pms.ipo.com/pages/viewpage.action?pageId=12164134#id-3.3.1.1项目管理需求及移交阶段-3需求要求)
* [4 需求收集](http://pms.ipo.com/pages/viewpage.action?pageId=12164134#id-3.3.1.1项目管理需求及移交阶段-4需求收集)
* [5 产品移交](http://pms.ipo.com/pages/viewpage.action?pageId=12164134#id-3.3.1.1项目管理需求及移交阶段-5产品移交)

**1 总体规则**

正常业务或技术需求,应遵循本文档中所有的定义及流程规范.

每次版本的文档都要记录在pms中相应的地方,在jira中也应有对应的版本任务.没有记录的工作视为无效工作.

项目的迭代要引入版本号的控制,避免极小需求引起的频繁上线.

**2 GNU版本号定义及管理策略**

**2.1 GNU版本号定义**

* + 版本号(version number)是版本的标识号。每一个项目都有一个版本号。
  + 版本号能使用户了解当前的项目是否为最新的版本以及它所提供的功能与设施。
  + 每一个版本号可以分为**主版本号、子版本号、修正版本号**三部分。

**2.2 GNU版本号管理策略**

* + 项目初版本时，版本号定义为 1.0.0, 其中1为主版本号，第一个0为子版本号，第二个0为修正版本号;

| **项目功能** | **主版本号** | **子版本号** | **修正版本号** | **版本号** |
| --- | --- | --- | --- | --- |
| 当项目在进行了主线逻辑修改或增加大功能时 | 加 1 | 复位为0 | 复位为0 | 2.0.0 |
| 当项目在进行了内容修改或优化或原有的基础上增加了部分小功能时 | 不变 | 加 1 | 复位为 0 | 1.1.0 |
| 当项目在进行了 bug 修正 | 不变 | 不变 | 加 1 | 1.1.1 |

**3 需求要求**

1. 所有业务核心流程各平台必须一样,但配置值可以存在差异
2. prd必须遵循统一模板
3. 所有业务或技术需求应当配备一定的**数据运营分析需求与监控需求**

**4 需求收集**

| **版本及需求实施规则** | | **实例** | |
| --- | --- | --- | --- |
| 项目每隔一段固定的时间开放一次上线窗口 | | 2周 | |
| 每次上线都对应一个版本(版本变更规则见上方),版本号由开发人员确认 | | 1.5或者2.0 | |
| 如果对应上线窗口无上线,则版本不变 | |  | |
| 每个项目都有一个需求池,产品经理不断将需求放入需求池([需求池模板](http://pms.ipo.com/pages/viewpage.action?pageId=9388614)),在每次的需求沟通会上,把需求放入对应的版本([版本跟进模板](http://pms.ipo.com/pages/viewpage.action?pageId=9381171)) | |  | |
| 需求沟通会上根据优先级确认一个月内的版本需求(无法确认的继续在需求池中滚动),并检查下个版本需求是否已经全部移交 | |  | |
| 每个版本上线日前一天就是下一个版本需求收集的截止日 | | 例如本次上线1.5版本,则1.6(或者更高)版本的需求收集也截止 | |
| 每个版本均会在pms项目中建立对应版本页面,在jira中建立对应版本jira,分支也应按照版本信息创建 | |  | |
| 每个版本需求量的上限为开发时间6天,测试时间3天 | |  | |
| 超出需求量则需要跨版本迭代 | |  | |
| 如遇紧急需求,需进行评估,是插入当前版本还是放入下一个版本.  如插入当前版本,而不需要把其他需求移到后续版本中,则可以调整开发测试时间,满足需求  如插入当前版本,导致有需求被提出,则当前版本之前约定的所有需求均需要延后至下一个版本  如果插入下一版本,则根据优先级及工作量,调整下一版本的需求 | |  | |
| 如果正在开发的版本发生需求变更,需邮件周知前后端开发,测试及相关主管. | |  | |
| 需要修改多个系统的需求,对应系统的版本号同时按照对应的规则递增版本号 | |  | |
| 每次新增需求,开发人员需要评估开发及测试工作量,产品需要评估其优先级,并记录在需求池中 | |  | |
| **项目需求沟通会议,议程及相关说明** | | |
| **目的** | 1. 确认当前版本需求的相关问题 2. 确认下个版本需要完成需求的相关问题 | |
| **议程** | 会前 | 技术leader需要了解当前版本所有需求的具体内容,安排好对应的人员及工作时间  产品经理需要移交完当前版本所有需求的文档 |
| 会中 | 确认当前版本需求已经全部移交,能够正常进入开发工作  确认下个版本需要完成的需求,并创建PMS中对应的版本页面和对应的Jira任务 |
| 会后 | 产品经理在每个需求的相关文档完成后需要确认需求优先级,并找技术leader确认项目工时,然后记录在需求池中 |
| **相关说明** | 需求移交根据需求工作量决定,由技术leader评估是会议移交还是线下移交 | |
| **意义** | 1. 产品经理可以更加关心产品,得到技术人员的反馈,包括问题和进度 2. 让开发人员尽早了解可能的需求,心里有底,能够更早的做准备 3. 加强沟通和交流 4. 了解人力资源的安排,询问开发员工的工作情况 5. 需求简单过一下,每个版本预期上线时间点的安排 | |

**5 产品移交**

1、需求移交会前工作

* + 产品经理应当提前发出产品prd,交互原型和设计稿.
    - 1天(含)的开发工作量可以不提前发出
    - 2-3天的开发量提前**半天**发出
    - 4-5天开发工作量提前**1天**发出,以此类推
    - 5天(含)以上项目,开发人员须在移交会前做好需求拆分和项目概要设计文档(参见[概要设计](http://pms.ipo.com/pages/viewpage.action?pageId=9388479)).
  + 产品文档发出后,开发及测试人员需仔细阅读文档及产品原型,及时发现其中的问题,并在产品移交会前与产品经理或相关人员讨论解决.如果在移交后再发生问题,将影响开发人员或测试人员考核.

2、需求移交会

* + 需求移交会一般是由产品经理召集；
  + 需求移交会上，产品经理讲解产品设计、业务流程及业务规范，回答开发、测试人员对于产品的疑问；
  + 开发及测试人员确认没有问题后才算移交通过.若发现产品大量需求不确定，测试人员有权终止移交。
  + 项目移交如有需求不明确可以要求再次移交,一旦移交后必须严格按照需求开发和测试.所以需求必须明确,没有需求就没有标准，开发和测试人员都只能按各自的理解和经验去做，最终很有可能和产品的期望不符

3、移交会结束

* + 移交完成的项目,需要在移交会第二个工作日下班前给出详细排期.
  + 建立版本开发Jira及structure，PMS建立版本内容