# 事项（或任务）

## 概念

任务分为几种类型：

计划任务?

快速任务（todo）? 除bug\测试任务外，其它都可

1. 需求分析 （系统类型）
2. 测试 （系统类型）
3. 开发 （系统类型，普通任务）
4. 管理 （系统类型，普通任务）
5. Bug （系统类型，由QA创建）
6. 其它自定义类型，如美工设计

计划任务包括：

一般计划任务【开发、管理、需求分析、其它自定义】

TODO，任务类型同上。

测试任务

Bug不属于计划任务，但可以在工作日志中出现

需求

特性(feature)

improvement

用户可以在一个任务下面创建子任务，则此上级任务自动转换为任务组。

**任务组**不包含负责人，不做为日志的撰写项

不包含起止时间，起止时间由子任务决定

任务组可直接创建

任务组可由任务手动转化而来，可以选择【将原任务转为子任务】，如果该任务已经开始执行，则此选项为必选。

## 属性

1. 标题、描述
2. 项目、计划
3. AssignTo，暂不可多人
4. 状态
5. 优先级
6. 附件
7. Comments，可以不断添加

历史记录

### 测试任务

1. 任务名称
2. **Build。**可以打build同时创建测试任务
3. 模块及模块负责人，可以从上一个测试任务中导入
4. 计划开始、结束日期
5. 实际开始、结束日期
6. 当前状态(未开始、进行中、被阻塞、已完成)

### 计划任务

1. 前置任务
2. 计划时间，开始-结束

### Bug

分为测试bug和外部bug

**测试bug：**

1. 标题
2. 所属测试任务
3. 所属项目、计划（根据所属测试任务得出）
4. AssignTo
5. 所存在的Build，默认最后第二个
6. 解决的Build，默认最后一个自动生成的Build，由开发人员填写
7. 描述【步骤】【结果】【期望】
8. 严重等级（紧急程度）
9. 优先级
10. 系统、浏览器
11. 附件
12. 所处状态，新建、已接受、处理中、已解决、重新打开 [允许自定义]
13. 类型。记录以往的输入，智能提示
14. 模块。记录以往的输入，智能提示

**外部bug**

1. 标题
2. 版本号
3. AssignTo
4. 解决的Build，默认最后一个自动生成的Build，由开发人员填写
5. 描述【步骤】【结果】【期望】
6. 严重等级（紧急程度）
7. 优先级
8. 系统、浏览器
9. 附件
10. 所处状态，已接受、已估时、已解决
11. 类型
12. 模块

**历史记录**

### TODO

快速任务

1. 标题、描述
2. 所属的release
3. 计划花的时间，确定的时间，可以不用做平衡，直接从其它计划中扣除时间。

计划在今天、明天完成，或设定开始-结束时间

## 变更历史

记录

* 修改的字段
* 字段原来的值
* 字段修改后的值
* 修改人

# 版本号命名规则

主版本号 . 子版本号 [ 修正版本号 [. 编译版本号 ]]  
Major\_Version\_Number.Minor\_Version\_Number[Revision\_Number[.Build\_Number]]

1. 项目初版本时，版本号可以为 0.1 或 0.1.0, 也可以为 1.0 或 1.0.0，如果你为人很低调，我想你会选择那个主版本号为 0 的方式；当项目在进行了局部修改或 bug 修正时，主版本号和子版本号都不变，修正版本号加 1；
2. 当项目在原有的基础上增加了部分功能时，主版本号不变，子版本号加 1，修正版本号复位为 0，因而可以被忽略掉；
3. 当项目在进行了重大修改或局部修正累积较多，而导致项目整体发生全局变化时，主版本号加 1；
4. 另外，编译版本号一般是编译器在编译过程中自动生成的，我们只定义其格式，并不进行人为控制

## Build

属性

1. 是否对外发布，如果对外发布的话，修订版本号加1
2. 如果不对外发布的话，只做测试打包

* SVN号
* 是否对外发布
* 版本号（自动生成）
* 源码地址
* 构建者
* Build日期

可以选择自动创建一个未来Build用以关联测试的解决Build