**TAPD 项目管理执行规范**

|  |  |  |
| --- | --- | --- |
| 文件名 | TAPD 项目管理执行规范 | |
| 类型 | 组织级 | |
| 应用范围 | OA研发中心 | |
| 简要描述 | 规范TAPD系统中关于项目的创建与使用过程。 | |
| 目的 | 规范项目线上管理，加强项目可控性，为管理者及项目成员及时了解当前各项目运行状况，为项目考核及绩效管理提供可量化参考依据。 | |
| 编写人 | 李凯靖 | |
| 审核人 | 李静 | |
| 批准人 | 黎宗耀、汤波成、曾祯祥 | |
| 更新历史 | | |
| 版本号 | 更新日期 | 更新描述 |
| 1.0 | 2018-01-12 | 新建文档 |
|  |  |  |
|  |  |  |
|  |  |  |

目录

**[1 创建类型 3](#_Toc21169)**

[1.1 产品维度 3](#_Toc14581)

[1.2 项目维度 3](#_Toc16323)

**[2 操作流程 3](#_Toc18109)**

[2.1 创建项目 3](#_Toc11858)

[2.2 创建需求 4](#_Toc22007)

[2.3 创建迭代 5](#_Toc2062)

[2.4 创建任务 6](#_Toc27174)

[2.5 创建缺陷 7](#_Toc16501)

[2.6 执行并更新需求/任务状态 8](#_Toc22194)

[2.7 项目验收 9](#_Toc23466)

# 创建类型

关于TAPD创建项目的规则，各部门按照实际产品管理需要，自行定义项目创建的颗粒度，其中创建新的项目时，建议选择【轻量敏捷项目管理】。

## 产品维度

* 定义：以产品功能模块划分范围，适用于较为独立、基本稳定运营的系统且将持续进行优化，或者采用敏捷管理的项目。
* 迭代方式：由项目负责人基于实际业务响应节奏，自行定义迭代周期。

## 项目维度

* 定义：以工作目标划分范围，适用于新产品/新系统开发、系统重构、跨多系统需求实施以及有明确专项性目标的项目（比如IT审计、核心改造等需要集中需求框范围管理的）。
* 迭代方式：由项目负责人按可验收的成果交付物分阶段定义迭代周期。

# 操作流程

## 创建项目

立项评审通过后，项目负责人需在两个工作日内完成项目的创建；常规维护类需求可在已有项目中创建需求并纳入迭代管理，具体视各产品线产品管理规则。

项目创建后，项目负责人需将相关干系人（产品、开发、测试、QA、BP等）加入对应的项目中，并将项目创建的相关信息以邮件形式发送相关干系人。

有关项目的权限设置，各业务线可通过【项目设置】-【成员与权限】-【用户组权限】划分不同的用户组，定义各用户组的权限。其中，有以下原则要求：

* 需对职能经理、QA、BP开放数据权限（即能查看/导出所有的缺陷、需求、迭代、文档、附件、报表等数据）。

## 创建需求

产品负责人创建需求并持续维护需求池，其中对颗粒度较大的功能需求，可遵循“用户角度可接受的最小功能”原则拆分为若干子需求，为避免需求层级过深，一般建议需求拆分不超过三层；

* 关于需求管理：创建需求时，产品负责人需要对需求的关键信息进行描述，如需求的背景与目的、边界与约束等；
* 关于需求状态的基本类型与含义如下表，各业务线可根据实际情况进行扩展：

|  |  |
| --- | --- |
| **状态名称** | **说明** |
| 需求调研 | 产品人员对原始需求进行分析，此时需求还不能划入迭代中 |
| 评审中 | 完成需求调研，项目负责人组织相关干系人进行确认，确认通过后可划入迭代 |
| 待开发 | 需求完成确认并划入对应的迭代计划中，开发负责人对开发任务进行分解与安排 |
| 开发中 | 开发实现中 |
| 待测试 | 开发阶段交付物等待被验证/测试 |
| 测试中 | 对开发阶段交付物进行验证/测试 |
| 已发布 | 完成验证并达到上线要求，产品发布在生产环境中 |
| 已验收 | 通过产品/用户方验收 |
| 已取消 | 需求失效，需求停止实施 |

* 关于需求状态的流转原则要求：

|  |  |
| --- | --- |
| **状态流转** | **执行人** |
| 评审中→待开发 | 项目负责人 |
| 待开发→开发中 | 开发负责人 |
| 开发中→待测试 | 后端开发/开发负责人 |
| 测试中→已发布 | 测试负责人 |
| 已发布→已验收 | 项目负责人 |
| 任何→已取消 | 项目负责人 |

## 创建迭代

项目负责人根据迭代计划创建对应的迭代，明确迭代周期信息（迭代开始时间、结束时间），并将相关需求划入迭代中进行管理。

* 关于迭代管理：产品团队需定期梳理需求，所有需求最终必须纳入对应迭代规划中进行管控；需求纳入迭代前要做好确认（如需求评审），纳入迭代后如出现变更，需纳入变更评估影响的范围内，并做好变更审批记录。
* 关于紧急需求：针对临时插入的紧急需求，如未处于任何版本发布周期内，可将其进行特殊标识，并单独处理，待上线验收后，再补充到就近的迭代版本中，以便于后续定位版本进行排查。
* 关于未规划的需求：对于近期暂时未做规划的需求，在规划清晰明确后，再纳入对应的迭代版本计划中。
* 关于迭代版本号管理：由项目负责人结合实际情况，根据规范要求规划本次迭代，版本号以VX . Y . Z进行标识，各字母含义如下：

|  |  |
| --- | --- |
| **标识名称** | **说明** |
| V | **代表版本Version。** |
| X | **主版本号**，初始值为1；主版本号的升级主要用于标识——在本次迭代周期内对产品进行了重大修改，或者产品功能增加、局部修改累积较多（上次迭代次版本号达到9），在用户体验、核心业务流程等方面与上一主版本有较大区别。 |
| Y | **次版本号**，初始值为0，数值范围为0-9；次版本号的升级主要用于标识——在本次迭代周期内，基于原有产品基础上，增加了部分功能，或者对原有功能进行了深化，在用户体验、核心业务流程等方面与上一版本有所区别。 |
| Z | **修订版本号**，初始值为0，数值变化范围为0-99；修订版本号的升级主要用于标识——在本次迭代周期内，基于原有产品基础上增加了少量功能，或者对上一迭代周期遗留的需求进行开发，对上线/遗留bug进行修复，在用户体验、核心业务流程等方面基本不改动。 |

备注：1. 在所有的版本号标识中，若版本号的前一位改变, 其后的版本号位都要归零；

1. 如果业务线内部因特殊需要有相关的迭代版本号管理措施，请通知QA进行报备，评估确认后按业务线的管理规则执行，其余无特殊情况下，按本规范要求执行。

## 创建任务

当迭代计划确认后，职能负责人基于需求进行任务拆分，任务名称以【任务类型】为标识对各种类型的任务进行区分，例如：

【UI】XXXX

【前端】XXX

【后端】XXX

【测试】XXX

* 关于任务的颗粒度：以各职能部门要求的任务分解颗粒度为准。
* 关于任务的管理： 创建任务时，职能负责人需要对任务工时与工期进行初步的评估（预计开始时间、预计结束时间、预计工时），并分配处理人；处理人在开始执行任务时，需填写实际开始时间，在完成任务后，及时将任务状态改为“已完成”并填写实际工时。
* 关于任务状态的基本类型与含义如下表，各业务线可根据实际情况进行扩展：

|  |  |
| --- | --- |
| **状态名称** | **说明** |
| 需求调研 | 需求调研任务执行中 |
| 设计中 | UI设计任务执行中 |
| 待开发 | 开发任务等待被安排/等待被执行 |
| 开发中 | 开发任务已完成 |
| 待测试 | 测试任务等待被安排/等待被执行 |
| 测试中 | 测试任务执行中 |
| 已完成 | （工作任务被验收后）任务完成 |
| 已取消 | 任务取消 |

* 关于任务状态的流转：以各职能部门要求为准，其中取消任务由职能负责人执行。

## 创建缺陷

测试团队将测试发现的/用户反馈的缺陷，创建到缺陷模块中。

* 关于缺陷管理：创建缺陷时，测试人员需要对缺陷进行描述，将缺陷关联到对应的需求与迭代中，并明确该缺陷的发现阶段；对于重新开启的缺陷，需要对重新开启的原因进行说明。
* 关于线上缺陷关联需求和迭代，无特殊情况下，请在当前迭代完成缺陷修复；
* 关于缺陷状态的基本类型与含义如下表，各业务线可根据实际情况进行扩展：

|  |  |
| --- | --- |
| **状态名称** | **说明** |
| 新 | 发现缺陷等待被确认/修复 |
| 接受/处理 | 开发接受缺陷并安排对缺陷进行处理 |
| 已解决 | 开发人员完成缺陷修复，等待测试验证 |
| 已验收 | 针对已解决的缺陷进行验证，测试通过后 |
| 重新打开 | 验证不通过/缺陷重新出现等 |
| 已拒绝 | 开发人员确认这不是一个缺陷 |
| 已关闭 | 缺陷关闭 |

* 关于缺陷状态的流转：缺陷处于已拒绝/已解决时，由测试方对缺陷进行验证确认后，再流转为已验收，如问题仍未被修复，则流转为重新打开。
* 关于缺陷发现阶段命名管理：在TAPD的缺陷创建与管理中，需要增加一个必填字段填写缺陷发现阶段，缺陷发现阶段以“Alpha、Beta、Release”进行标识，各标识的具体含义见下表：

|  |  |
| --- | --- |
| **标识名称** | **说明** |
| Alpha | **内部测试**阶段发现的缺陷，一般只在内部运行，不对外公开；主要是项目组成员对产品进行测试，检查产品是否存在缺陷、错误，验证产品功能与需求规格说明书是否一致 |
| Beta | **Beta环境测试**阶段发现的缺陷，当软件进入Beta环境给用户代表进行测试，该版本相对于Alpha版已有了很大改进，消除了严重的错误，但还存在着一些缺陷，需要经过进一步测试，以便在正式上线前进行改进和完善 |
| Release | **正式环境测试**所发现的缺陷，通常产品已发布在正式环境并已交付给用户使用。 |

* 关于缺陷类型命名管理：在TAPD的缺陷创建与管理中，需要增加一个必填字段填写缺陷类型，各标识的具体含义见下表：

|  |  |
| --- | --- |
| **标识名称** | **说明** |
| 功能问题 | 影响了重要的特性、用户界面、产品接口、硬件结构接口和全局数据结构。并且设计文档需要正式的变更。如指针循环，递归，功能等缺陷。 |
| 接口问题 | 与其他组件、模块或设备驱动程序、调动参数、控制块或参数列表相互影响的缺陷。 |
| 界面问题 | 人机交互特性：屏幕格式，确认用户输入，功能有效性，页面排版等方面的缺陷 |
| 环境问题 | 由于设计、编译和运行环境引发的问题。 |
| 配置问题 | 由于配置库、变更管理或版本控制引起的错误。 |
| 安全问题 | 系统存在的某种可能被入侵者恶意利用或攻击的问题 |
| 性能缺陷 | 不满足系统可测量的属性值，如：执行时间，事务处理速率等。 |
| 其他 | 其他问题 |

* 关于缺陷等级命名管理：在TAPD的缺陷创建与管理中，需要增加一个必填字段填写缺陷严重程度等级，以“致命、严重、一般、建议”进行标识，各标识的具体含义见下表：

|  |  |
| --- | --- |
| **标识名称** | **说明** |
| 致命 | 致命的错误，造成系统或应用程序崩溃（crash）、死机、系统悬挂、或造成数据丢失、主要功能组完全丧失 |
| 严重 | 严重错误，指功能或者特性（feature）没有实现，主要功能丧失，导致严重的问题，或致命的错误声明 |
| 一般 | 不太严重的错误，这样的缺陷虽然不影响系统的基本使用，但没有很好的实现功能，没有达到预期的效果。如次要功能丧失，提示信息不太正确，或用户界面太差，操作时间长等。 |
| 建议 | 一些小问题，对功能几乎没有影响，产品及属性仍可使用，如有个别错别字、文字排列不整齐等 |

## 执行并更新需求/任务状态

各任务执行人按计划执行任务并及时在系统中更新任务状态，工作成果如需求文档、设计文档、测试用例等项目过程产出物提交至TAPD【文档】中。

* 项目负责人对整体进度负责，各职能负责人需定期汇报工作进展，协助项目负责人处理项目问题及风险；
* 各负责人务必及时更新需求/任务状态。

## 项目验收

产品发布上线后，项目负责人以邮件形式向验收人发起验收申请，验收人对项目进行验收并在验收通过后回复邮件，项目负责人将父级需求状态扭转为“已验收”并将验收邮件作为附件上传。