



敏捷开发项目管理 指导规范

文件状态：	当前版本：	2.2
<input type="checkbox"/> 草稿	作 者：	XXX
<input checked="" type="checkbox"/> 正式发布	完成日期：	2017-2-8
<input type="checkbox"/> 正在修改		



版本历史

版本	修订人	修订日期	修订内容	批准人	批准日期
1.0	XX	2015-03-18	新建文档		
1.2	XX	2015-04-14	敏捷项目管理模式建立		
1.3	XX	2015-04-16	临时项目管理模式建立		
1.4	XX	2015-05-14	文档中整合与新增激励绩效模块		
1.5	XX	2015-05-15	删除文档中绩效模块		
1.6	XX	2015-05-17	重新整理文档，只保留敏捷项目管理，增加管理概论章节		
1.7	XX	2015-05-21	去掉组织结构章节	XX	2015-05-21
1.8	XX	2015-08-27	将故事清单文档删除，并将故事清单添加到立项报告中	XX	2015-08-26
2.0	XX	2017-01-19	将开发的模式改成敏捷模式	XX	2017-01-20
2.1	XX	2017-02-06	对邮件标题，文件名，部分过程文档进行字段修订，追加故事清单模板	XX	2017-02-06
2.2	XX	2017-02-08	在迭代执行阶段加入紧急开发任务	XX	2017-02-08



目录

目录

敏捷开发项目管理	1
1 定义	4
2 过程管理	5
2.1 启动	6
2.2 规划	6
2.2.1 需求分析	7
2.2.2 迭代规划	7
2.2.3 任务拆解	8
2.3 执行	9
2.3.1 紧急任务	9
2.3.2 项目开发	10
2.3.3 部署验收(测试)	10
2.3.4 部署交付(正式)	10
2.4 收尾	11
2.5 会议	12
2.5.1 迭代规划会议	12
2.5.2 任务拆解会议	13
2.5.3 每日站会	13
3 成果列表	13
3.1 文档	13
3.2 邮件	14

1 定义

敏捷开发，作为系统部应用开发组的业务工作模式，以迭代为周期，时间为 1 周或 2 周。每一次迭代中包含多个项目。其工作成果以项目版本或文档版本为体现。

迭代 - **sprint**: 一个敏捷团队的一段完整工作时间，在现在团队中工作时间为 1 周或 2 周且均以周一作为开始，以本周末或下周末作为结束。

项目 - **project**: 为了实现一个或一组特定目标的开发工作，在现在的团队中项目的体现为以产品的版本号为工作成果的一系列工作集合。

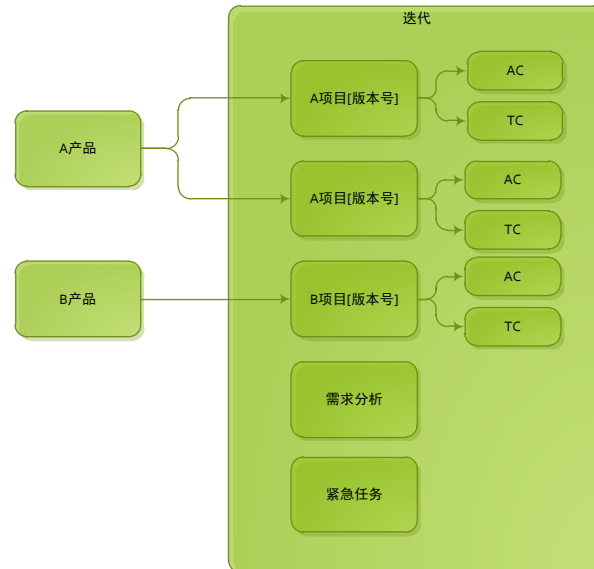
产品 - **Product**: 以解决工作中某个特定业务的服务，在现在团队中我们的产品是一套以程序为中心的完整的解决方案，一款产品有多个模块和一系列文档组成。

PB - Product backlog: 产品待办事项，在现在团队中的体现为一个业务上的需求或一款产品需要提供的特性。

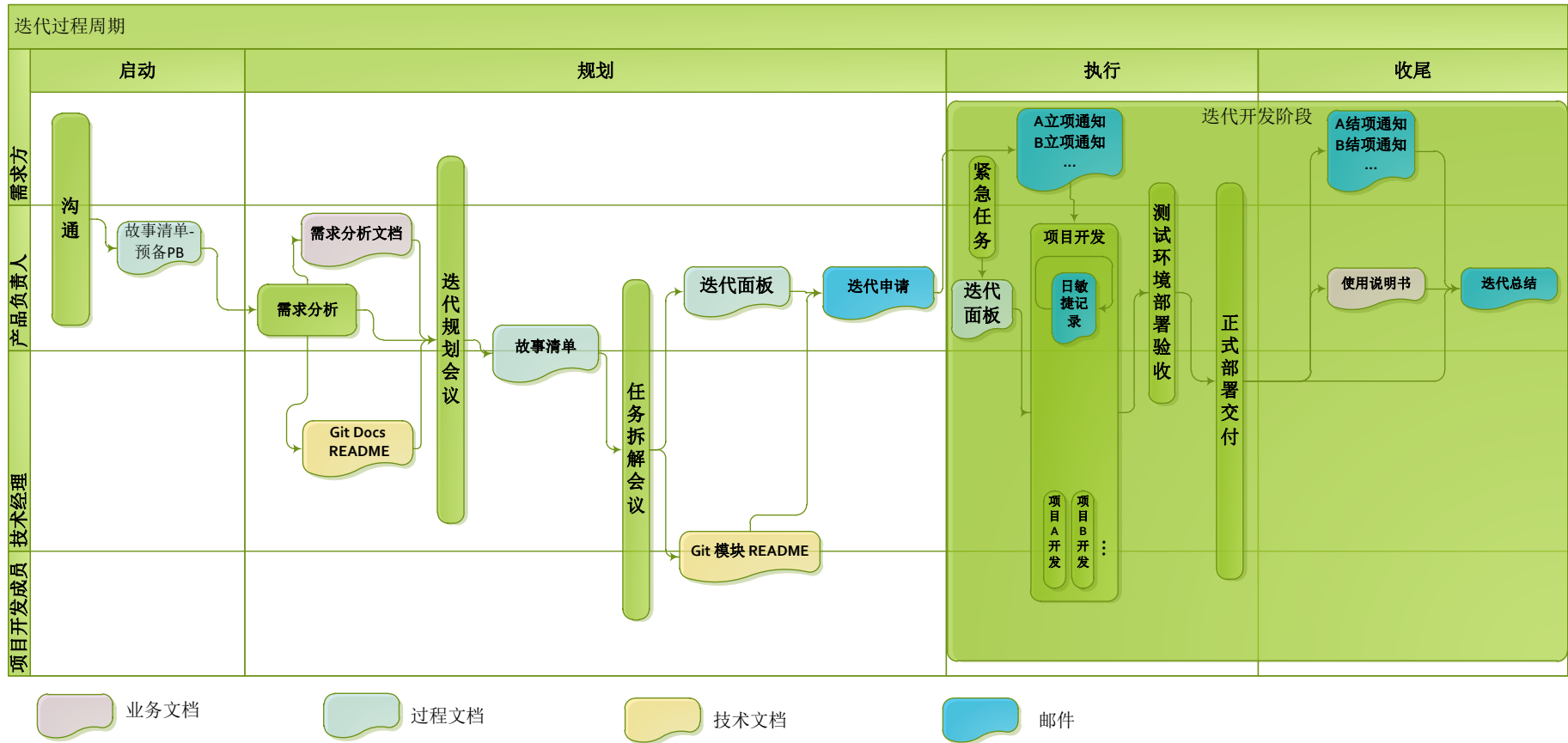
SB - sprint backlog: PB 的子集，在现在团队中的体现为一个项目。

AC - Accept Case: 完成 SB 所需要的可验收的工作细节。

TC - Test Case: AC 的验收指标。



2 过程管理





在我们的敏捷开发过程中，团队是逐步建立的，并且团队由产品负责人、技术经理、部门成员协商组建，在迭代中团队成员与阶段的对应关系：

启动阶段：产品负责人

规划阶段（迭代规划会议前）：产品负责人、技术经理

规划阶段（迭代规划会议后）：产品负责人、技术经理、团队成员

执行阶段：产品负责人、技术经理、团队成员

收尾阶段：产品负责人

2.1 启动

输入	流程	输出	参与者	形式	备注
初始需求	沟通	《故事清单-PB》	产品负责人	会议、访谈	——

启动的过程即领导批准可以开始动用资源的过程，此过程在我们的部门内部通常为口头形式(也有部分为书面)，此时应由确立的产品负责人跟进，对原始需求定义章程，并为需求方简单整理《故事清单-PB》。

2.2 规划

输入	流程	输出	参与者	形式	备注
《故事清单-PB》	需求分析/设计	文档：《需求分析文档》 Git：《Docs Readme》	技术经理、产品负责人、需求方	——	——
《需求分析文档》 《系统概要设计》 文档：《故事清单-PB》	召开迭代计划会议	文档：《故事清单-SB》	高管,主管,技术经理,产品负责人	会议	



《需求分析文档》 《Docs Readme 详细设计》 《故事清单-SB》 《Docs Readme 数据库设计》	召开任务拆解会议	文档《迭代面板》 Git: 《模块 Readme》	技术经理,产品负责人,程序开发人员	会议	发送迭代申请邮件
--	----------	------------------------------	-------------------	----	----------

规划阶段分为需求分析、迭代规划、任务拆解。

2.2.1 需求分析

在开始需求分析之前，产品负责人需在为知笔记为系统创建业务文档目录，技术经理在 Gitlab 中为系统创建代码存放目录。从已经整理的原始需求与基础指标开始着手进入系统项目的需求分析阶段，目前我们团队接收到的需求可以大致分为两个方向：一个涉及业务逻辑，一类涉及技术研究。产品负责人分析整理业务逻辑，技术经理分析整理技术研究。

1、业务逻辑

复杂业务形式的项目比较有代表性的即为社招系统，故障管理系统，此类项目的难点不仅在角色，人员，流程关系上复杂，还需要对在用户体验上进行合理的设计，此类型业务在需求分析阶段由产品负责人负责分析，技术经理需要对产品负责人的需求分析的合理性与技术实现性提出修改意见。产品负责人根据技术经理的修改意见做出一份达到需求方满意的《需求分析文档》

《需求分析文档》中主要从复杂业务逻辑(流程)，角色，原型方面在项目前期为需求方提供一个可预见型的项目原型，并为项目组的后期开发提供依据。

2、技术研究

业务逻辑的实现是建立在合理的技术架构之上的，每个项目中的技术经理就是要保证项目实现的技术架构的合理性、可行性。技术经理根据产品负责人产出的《需求规格说明书》中的分析原型，出具《Docs Readme》，《Docs Readme》包括：《Git Readme 系统架构》《Git Readme 数据库设计》。

《Git Readme 系统架构》中主要涉及程序的逻辑/时序,技术选型，核心技术点，为领导确定项目风险起关键性作用

《Git Readme 数据库设计》中主要描述项目的数据库详细设计

2.2.2 迭代规划

经由领导层/需求方确认需求分析产出后，即可进入迭代规划阶段，以每周三的迭代规划会议为分界点。在【迭代规划会议】之前，产品负责人按照需求分析阶段的《需求分析文档》拆解出项目的《故事清单-SB》，并保证《故事清单-SB》里的每一个事项完成时间不超过一周。在【迭代规划会议】上，产品负责人讲解《故事清单-SB》，主管为每个故事事项确定优先级，并确定下一个迭代的内容和开发人员。

2.2.3 任务拆解

迭代规划会议之后进入任务拆解阶段，由产品负责人组织召开【**任务拆解会议**】，成员包括产品负责人、技术经理、团队开发人员。在【**任务拆解会议**】上，需要对《故事清单》里的任务做详细设计。为了共同提升我们的技术能力，团队中每一个人员都要参与详细设计；在【**任务拆解会议**】上，产品负责人为团队讲解此次迭代《故事清单》的每一个故事，保证团队的每一个成员对每一个故事都能正确理解，技术经理根据《*Git Readme 概要设计*》《*Git Readme 数据库设计*》将每一个故事拆解成合理的模块，再由团队成员讨论成合理的详细设计方案。会议结束后技术经理根据大家的讨论结果修订《*Git Docs Readme*》，并由技术经理或指定成员完善《*Git Readme 模块详设*》。

会议上除了全团队补充修订完成详细设计外，还有一部分重要内容：根据迭代中项目的详细设计将《故事清单》进行细节拆分形成 AC 与 TC 并完成《迭代面板》，《迭代面板》应该提供最合适的产品行为描述，且每项 AC 均需达到可验收的标准且每项 AC 往往会对应多项 TC 来确保功能的可验收性，如 AC 中往往会包含用户登录，用户注册，采集数据至 Mysql 等，TC 中包含使用正确用户名可以成功登陆并跳转至正确页面，使用错误用户名不可成功登陆且弹出错误提示等等。会议后产品负责人将产出的资料《迭代面板》发送团队的全体成员，并给主管发送一份迭代申请的邮件。

注意：1、一个迭代周期总是固定并一致的，一次迭代的标准时间是 1 周。

2、完成《迭代面板》时加入敏捷开发中的出牌阶段，为每一个迭代任务分配一个工作时长（由于每个人的技能特长不一样，这个时长是一个相对值），每个迭代任务的工作时长不能超过 8 小时

这个会议的时间可能持续较长，但这也是我们共同提升的一个大好机会。

迭代申请邮件：

发件人：产品负责人
收件人：主管，各职能组长
抄送：团队成员，副总裁
标题：迭代申请-团队名-[2017 年 2 月第一周]-周期 1 周
邮件内容：
 “《迭代申请》内容”
附件：
 《迭代申请》 《迭代面板》

注意，后续邮件采用全部回复形式，从第一封迭代申请开始，后续邮件追上封全部回复

2.3 执行

输入	流程	输出	参与者	形式	备注
《需求分析文档》 《Git Readme 系统架构》 《Git Readme 数据库设计》 《Git Readme 模块详设》 《迭代面板》	开发	《迭代面板》(更新)	技术经理、产品负责人、开发人员	——	每日发送敏捷记录

迭代开发就不在详述了，期间均由产品负责人督促举行【日站会】并负责发送日敏捷记录邮件，日敏捷记录里包括每日站会的会议纪要和燃尽图。

日敏捷记录邮件：

发件人：产品负责人
收件人：主管，各职能组长
抄送：团队成员，副总裁
标题：日敏捷记录-周-星期 -团队名- [2017 年 2 月第一周] -周期 1 周
邮件内容：
燃尽图
日站会会议纪要

注意，邮件采用全部回复形式，从第一封迭代申请开始，后续邮件追上封转发

2.3.1 紧急任务

运维项目是为一线工作人员使用的，需要保证项目长期稳定的运行。紧急任务是指已上线系统功能存在目前影响正常业务运转的功能性 BUG 与已经过部门级领导审批的确定必须立即进展的临时性工作。接到紧急任务时，职能组长指定具体产品负责人负责。产品负责人接到任务后，产品负责人判断此任务是否可划分为紧急任务，如果是就加入到当前的《迭代面板》中，如果不是就加入到《故事清单-PB》中。紧急任务的一旦加入到《迭代面板》中，就必须完成。

2.3.2 项目开发

每次迭代可能包含多个项目。每个项目开始的前一天，产品负责人需要给需求方发送立项通知邮件
立项通知邮件：

发件人：产品负责人
收件人：需求方
抄送：项目组所有干系人，团队成员,组长,主管
标题：项目名称[版本号]立项通知-团队名-[2017年2月第一周]迭代
邮件内容：
 本项目开始开发，功能如下：
 1、***
 2、***

 详细内容见附件
附件：
 《需求文档》

2.3.3 部署验收(测试)

内部验收为我们测试环境的验收，测试环境部署（在测试 Jenkins 环境中构建成功）完成后，由产品负责人对其进行功能性验收。注意，只有当在测试环境验收完成后，才可以进行正式环境的上线部署申请。

2.3.4 部署交付(正式)

正式部署交付即将在测试环境已经测试通过的产品部署至正式环境（运维部署），并开通相应权限，由产品负责人进行简单点击测试，发送交付邮件。
自此时项目的资源由项目组转移至产品负责人。

结项通知邮件

发件人：产品负责人
收件人：需求方
抄送：项目组所有干系人，团队成员，组长，主管
标题：产品名称[版本号]验收通知-团队名-[2017年2月第一周]迭代
邮件内容：
 本项目已完成，并已上线正式环境，请验收并及时回复确认邮件。
 功能点：
 1、xxx
 2、xxx
附件：
 《使用手册》

结项通知发出后，产品负责人需要与需求方沟通，为项目的工作成果的打分。需求方在结项通知邮件后回复反馈意见。

2.4 收尾

输入	流程	输出	参与者	形式	备注
--	整理	Email: 《迭代验收》	技术经理、产品负责人	——	

收尾阶段工作为整理项目的产出文档，产品负责人完成相应工作，发送至各项目干系人，（注意：收尾执行必须与项目交付同天内完成）

迭代验收邮件

发件人：产品负责人
收件人：项目组所有干系人
收件人：主管，各职能组长
抄送：团队成员，副总裁
标题：迭代总结报告-团队名-[2017年2月第一周]-周期1周
邮件内容：

“《迭代总结》内容”

附件：

《迭代总结》，本次迭代中所涉及项目的流程邮件(压缩包)

2.5 会议

会议清单

会议名称	迭代计划会	任务拆解会	每日站会
主持人	产品负责人	产品负责人	产品负责人
参会	敏捷团队成员 职能组长 部门主管 高层领导	敏捷团队成员	敏捷团队成员
时间	周三	周四-周五	工作日
输入	《故事清单》	《迭代计划》	《迭代计划》
产出		《迭代申请》 《迭代面板》	

2.5.1 迭代规划会议

产品负责人组织召开迭代计划会，产品负责人讲解《故事清单》里的事项，主管，组长或其他高层确定下一个迭代需要完成的 PB，并为与 PO，技术经理协商为每个 PB 确定一个交付时间，职能组长为下一个迭代确定团队开发人员。

参与人员：产品负责人、主管、职能组长、技术经理

时间：每周三 17 点

2.5.2 任务拆解会议

产品负责人组织召开任务拆解会议，开发团队将《故事清单》会中确定的 SB 拆解成可完成的任务，拆解过程中加入出牌环节，为每一个任务分配一个合理的完成时长。

参与人员：产品负责人、技术经理、团队成员

时间：每周四-周五

产出文档：迭代面板

2.5.3 每日站会

产品负责人组织召开每日站会，开发人员交流任务完成的情况，领取未完成任务，及时更新项目敏捷面板

参与人员：产品负责人、技术经理、团队成员

时间：每个工作日

产出文档：迭代面板，日敏捷报告(Email)

3 成果列表

3.1 文档

类型	存放位置	名称	模板	必有	操作类型
过程文档	为知笔记/应用开发/部门报表/	故事清单	●	●	更新
	为知笔记/应用开发/部门报表/迭代记录	团队名- [2017 年 2 月第一周] -周期 1 周	●	●	新增
技术文档	http://gitlib.dev.gyyx.cn/[项目英文简称]/docs/[大版本号]/	Docs Readme 系统概要设计	●	×	新增/追加
		模块 Readme 详细设计	×	×	新增/追加
		Docs Readme 数据库的详细设计	×	×	新增/追加
业务文档	为知笔记/应用开发/核心业务/项目中文名/	需求文档	×	●	新增
		需求规格说明书	×	●	新增



	系统使用说明书	×	●	新增
	版本更新记录	●	●	新增

3.2 邮件

邮件类型	模板	必有	责任人	默认抄送
迭代申请 邮件	●	●	产品负责人	主管/技术经理/需求部门直接上级/中心副总/
立项通知 邮件	●	●	产品负责人	
日敏捷报告 邮件	●	●	产品负责人	
会议纪要 邮件	●	●	产品负责人	
结项通知 邮件	●	●	产品负责人	
迭代总结报告 邮件	●	●	产品负责人	

*管理概论

PMI 项目管理

	<h3>项目边界</h3> <p>项目的一个重要的事情就是认清项目边界，项目的完成与我们目前所说的项目落地为不同的概念，而且是最容易混淆的概念，项目落地涉及项目的运营，并非仅凭项目经理个人能力所能实现的（但是项目交付成果的质量会起到重大作用）。</p> <p>在我们新一版本的项目管理规范中，重新定义此边界，即验收方在部署验收(测试)阶段通过项目即进入收尾程序，且项目的正式交付仅由项目经理进行简单的功能性测试后整理相应的记录文档，即项目结项。</p> <p>项目成功不代表项目的运营一定有较高产出回报</p> <p>注意：以我们的特殊环境，虽然项目结项了，项目经理对每个版本需要继续跟进后续运营，且后续运营成果将关系到项目组绩效。</p>
	<h3>阶段与成本</h3> <p>PMI 为我们介绍了一个项目的阶段与成本对应关系，在我们的新版项目管理中的体现即为项目组的逐渐建立过程，从项目的启动开始，逐步的建立项目组，至项目的结束阶段(收尾)，项目组解散。</p> <p>其实，我们的曲线与 PMI 标准会有少许变化，我们将在组织与准备(规划)阶段增大成本的投入，这样更能保证项目的一次性成功落地。</p>

Scrum 敏捷开发

