深圳国泰安教育技术股份有限公司	版本	密级	页数
	V1.0	机密	共 38 页

项目级敏捷流程

作者	熊芸	编制日期	2017-02-03
审核	付艳华	审核日期	2017-02-22
批准	陈工孟	批准日期	2017-03-01

国泰安教育技术股份有限公司
版权所有 侵权必究

文档修改记录

版号	修改日期	修改内容	修改人
V0.1	2017-02-03	根据敏捷过程改进要求编制该文档	熊芸
V1.0	2017-03-16	根据评审意见进行修改	熊芸
V1.0.1	2017-06-02	增加三种层级的完成标准	熊芸
V1.0.2	2017-07-18	更改审核和批准日期	熊芸

目录

前言

敏捷软件开发提倡轻而灵活的过程，反对把过程描述的事无巨细、大而笨重，但绝不是说敏捷不需要流程。在敏捷开发过程中，存在核心的工程活动和关键的管理活动，这些活动的定义、要求、及活动之间的相互关系，构成了项目级敏捷的过程框架，也构成了本文的主线。本文对过程中必须达到的内容提出明确要求，在能为团队提供帮助时，也给团队扩展的余地，产品团队可以在本文框架定义的活动基础上，进一步扩展子活动和补充活动方法的细节。

1） 本文包含的主要内容：

（1）项目级敏捷流程的目的和适用范围；

（2）项目级敏捷中的角色定义，这些角色在后续活动中引用，方便读者理解这个过程；

（3）项目级敏捷的过程和活动指导，项目级敏捷的迭代开发过程的三大阶段和 18 个关键活动。

2） 项目级敏捷活动总揽及关键概念说明：

18 个关键活动的指导，包含活动目的、输入、输出、角色、活动描述、活动要点、方法等。

1. 概要

1.0. 目的

本文规范国泰安项目级敏捷开发的过程、角色，为各产品线项目级敏捷的实施和流程制定提供约束和参考。

1.1. 范围

项目级敏捷定义：项目级敏捷指产品完成立项、系统设计规划后，在**需求逐步明确的条件下**，具有迭代、持续集成和自适应特征的软件开发模式。项目级敏捷聚焦单个项目组或多个项目组协同的软件开发过程和能力改进，对产品级的交付和非研发领域（用户服务、市场等）没有变化和影响。

1.2. 术语和定义

对本文中用到的术语进行定义。

编号	角色名称	简称	英文全称
1	产品负责人	PO	Product Owner
2	敏捷教练	SM	Scrum Master
3	项目经理	PM	Project Manager
4	项目团队－架构师	SE	System Engineer
5	项目团队－UE 的工程师	UE	User Experience Design
6	项目团队－UI 的工程师	UI	User Interface Design
7	项目团队－前端工程师	FE	Front-End Development
8	项目团队－3D 建模工程师	ME	3D Modeling Engineer

9	项目团队－3D 动画设计工程师	ADE	3D Animation Design Engineer
10	项目团队－3D 美术设计师	AE	3D art designer
11	项目团队－数据库工程师	DBA	Database Engineer
12	项目团队－开发工程师	SWE	Software Engineer
13	项目团队－测试工程师	TE	Test Engineer
14	项目团队－软件质量保证工程师	QA	Quality Assurance Engineer
15	项目团队－配置管理工程师	CM	Configuration Management Engineer
16	团队成员	TEAM	Team
17	商务 BD	BD	Business Development
18	运维工程师	/	Operation and Maintenance Engineer
19	实施工程师	/	Implementation Engineer
20	客户	CUSTOMER	Customer

2. 项目级敏捷过程活动和指导

2.0. 完成标准

清晰和明确的完成标准保证了每次迭代是高质量的。并且完成标准需要共同协商，团队自我承诺会更认真，最后完成标准可用于准确评估团队工作进展 用于准确评估团队工作进展。国泰安公司三个层级的完成标准如下。

内部完成标准：

1） 通过回归测试；
2） 通过性能测试；
3） 更新配套手册。

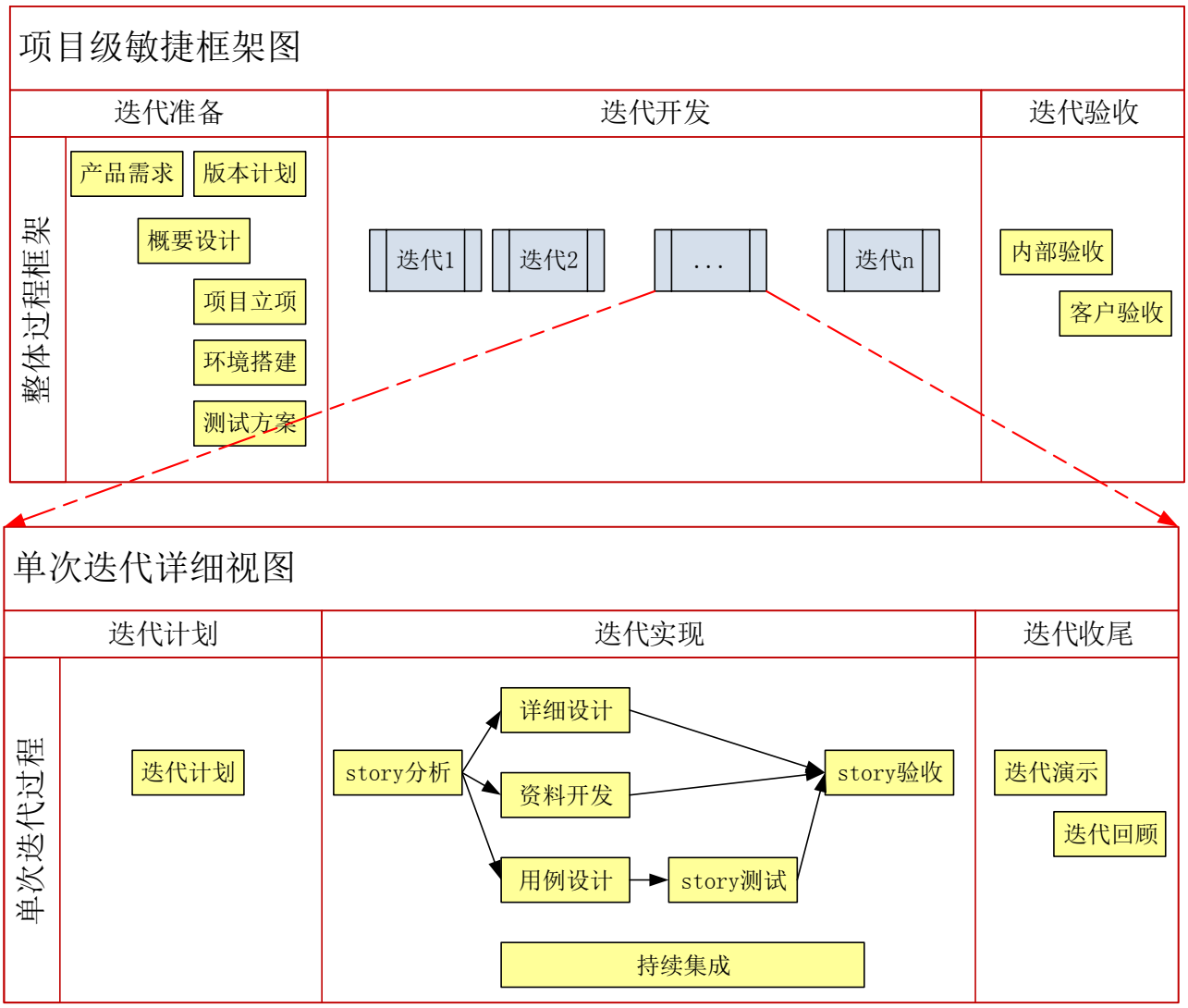
迭代完成标准：

1） 所有 story 验收完成；
2） 系统测试用例通过率超过 95%；
3） 主流程测试通过，且 PO 整体验收通过；
4） 如果迭代版本需要对外发布，需满足遗留缺陷加权分不超过 2 分。

story 完成标准：

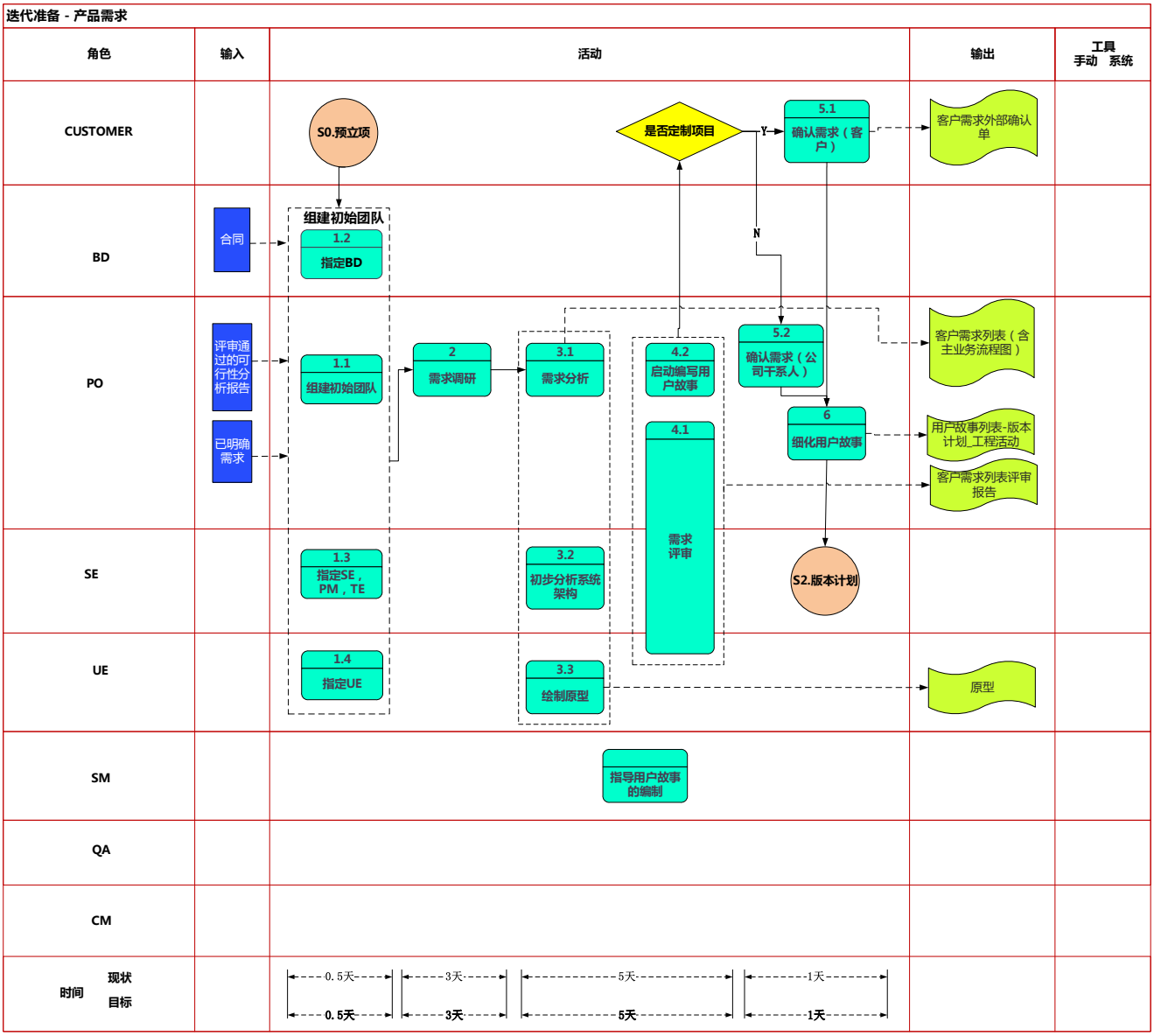
1） 代码合入主干；
2） 代码符合规范；
3） 研发自测通过；
4） story 测试通过；
5） story 验收通过。

2.1. 项目级敏捷活动总览



项目级敏捷迭代准备

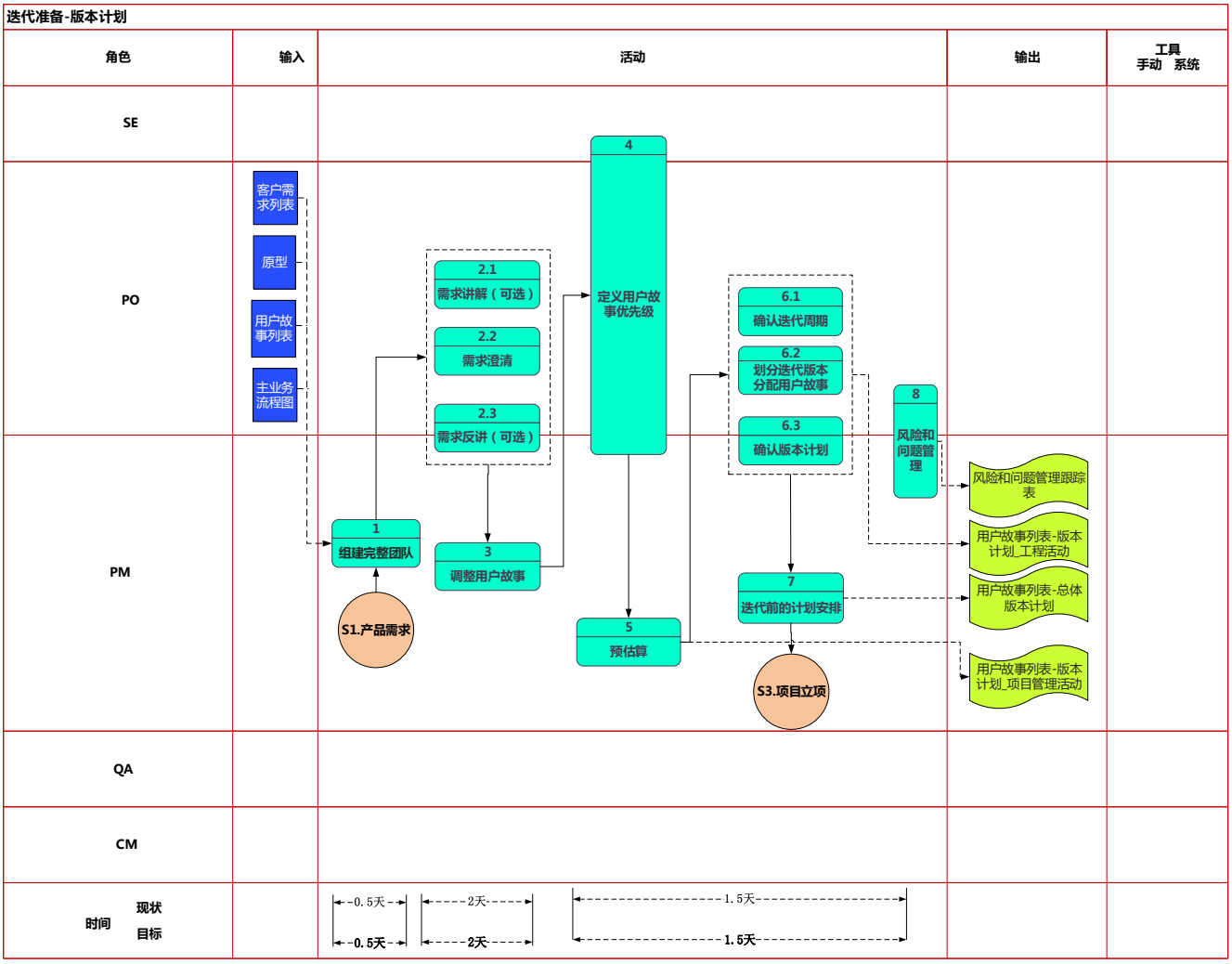
2.1.1. 产品需求



活动名称	**产品需求**
活动目的	1. 输出《客户需求列表》； 2. 初步确定需求的优先级； 3. 进行需求的进一步分析，输出主业务流程图； 4. 根据需求，制作原型； 5. 在需求渐进明细过程中，编制《用户故事列表》； 6. 进行需求的客户确认。
责任角色	PO
参与角色	SM、PM、SE、TE、UE、BD、CUSTOMER
输入	合同 评审通过的可行性分析报告
输出交付件	客户需求列表、用户故事列表、客户需求列表评审报告、客户需求外部确认单

活动描述	1. 组建初始团队进行客户需求的调研、分析及确认。 2. PO 进行需求调研，根据需求调研结果编制《客户需求列表》，《客户需求列表》的需求颗粒度要求分解到业务场景。并初步确定需求的优先级。 3. PO 基于需求调研的结果进一步进行客户需求分析，输出主业务流程图，如涉及到界面需求，PO 需进行界面需求分析，输出《UE 界面需求》。如需要，PO 可设计原型，必要时 UE 人员协助完成。 4. PO 负责组织产品需求评审会议，并邀请相关干系人参加。 5. PO 整理所有需求清单与相关干系人再次沟通确认。 6. PM 基于《客户需求列表》进行工作量的量级估算，用于合同评估。 7. SE 根据《客户需求列表》初步分析系统架构
活动要求	1. 《客户需求列表》的需求颗粒度需要分解到业务场景。 2. 客户需求的优先级可以在客户需求列表初步确定，渐进明细过程中继续更新优先级，最晚需在版本计划里最终确定。 3. 《用户故事列表》的 story 分解粒度要求 3~5 天之内，最小单位为 0.5 天，并且优先级高的 story 尽量排列在前面。 4. 主业务流程图的表现方式以 Visio 为主、Axure、图片为辅。 5. 客户需求评审时，顺序建议先展示主业务流程图，然后原型与用户故事列表结合来评审。 6. 如果是自研项目，将所有的需求清单与公司干系人进行再次沟通确认，公司干系人包括 BD、SE、PM、TE、事业部总经理、SQA。如果是定制项目，PO 可邀请 PM 到客户现场进行需求的再次沟通确认，重点是获得客户负责人的确认证明，方式可以是邮件确认、签字确认或盖章确认。
补充说明	
方法与工具、IT	Visio、Axure
备注	

2.1.2. 版本计划

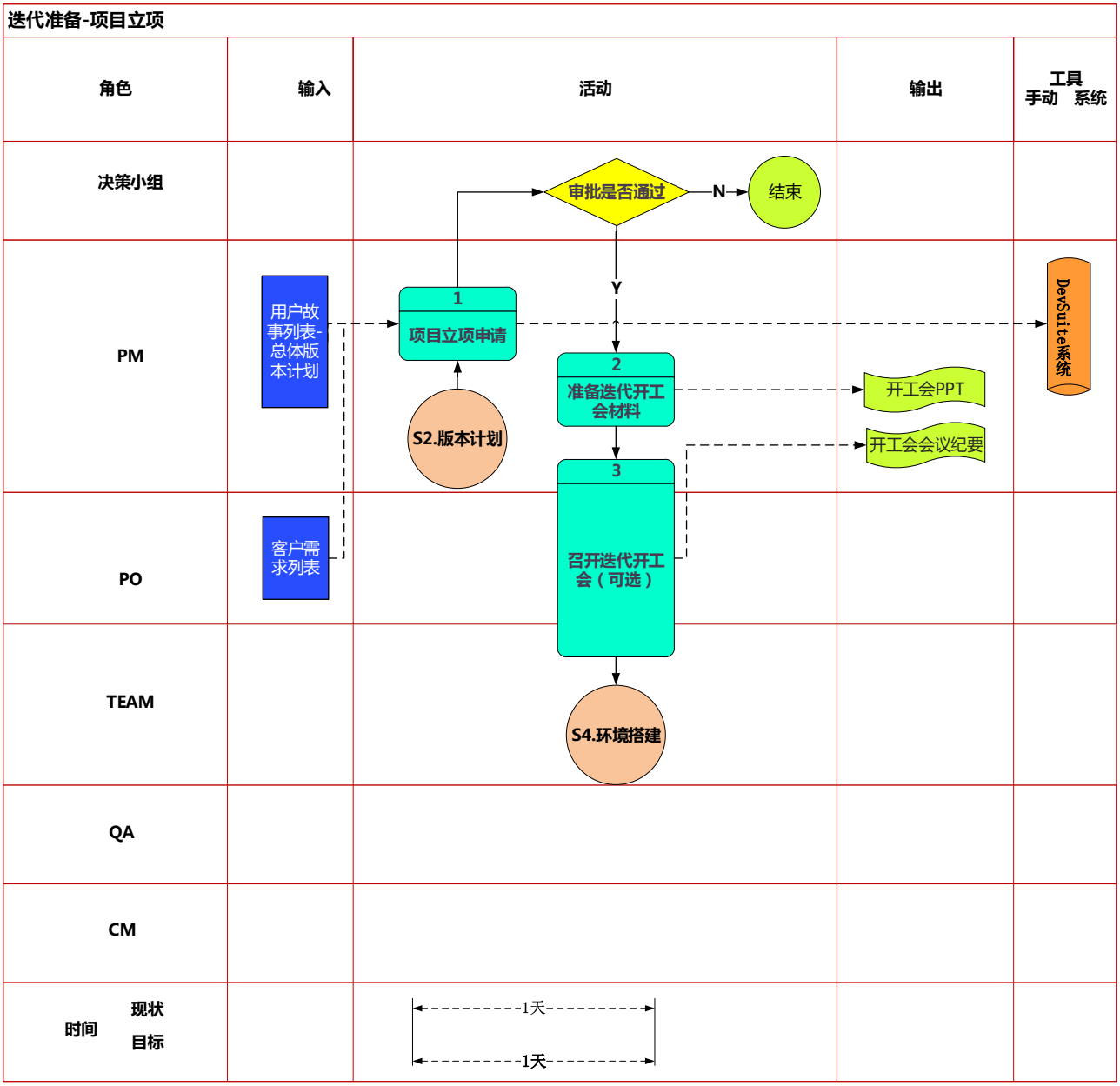


活动名称	**版本计划**
活动目的	1. 组建完整项目团队成员，确认角色与职责； 2. 确认团队对需求理解一致，调整用户故事，明确用户故事优先级； 3. 进行用户故事工作量预估算； 4. 确定迭代周期、迭代用户故事分配，输出初始的版本计划； 5. 迭代前的计划安排； 6. 初步识别项目风险和问题，并制定应对计划。
责任角色	PM
参与角色	PO、SM、SE、UE、SWE、TE
输入	1. 客户需求列表 2. 原型 3. 主业务流程图 4. 用户故事列表
输出交付件	用户故事列表-总体版本计划、用户故事列表-版本计划_工程活动、用户故事列表-版本计划_项目管理活动、风险和问题跟踪表

活动描述	1. PM 确认完整团队是否已完成组建，如果已组建完整团队则开始第 2 项活动，否则 PM 需协调各部门资源完成完整团队组建，包括各角色职责与分工。 2. PM 组织 PO、SM、SE、UE、SWE、TE 对需求进行讲解、澄清和反讲，需求讲解及需求澄清主要由 PO 主导，需求反讲重点针对有疑问的需求。如果项目团队对需求已经理解到位，需求讲解及需求反讲活动可选。 3. PO 按照用户故事的拆分原则，进行用户故事的调整。 4. PO 参考《客户需求列表》的需求优先级，综合考虑 SE 给出的架构风险和技术风险，给出用户故事优先级的综合评估。 5. PM 组织 PO、SM、SWE、TE 进行用户故事的预估算，包括软件工程活动工作量和项目管理活动工作量；软件工程活动工作量包括开发工作量（含对故事的理解、设计、开发编码、BUG 修复、沟通、评审（开发主导，包括测试工程师评审的工作量），协助 PO 进行 story 验收的工作量等直到某个 story 完成时的所有工作）和用户故事测试工作量（对故事的理解、细化测试策略和方案、用例设计与编写及修改、沟通、评审（测试主导，包括开发工程师评审的工作量），执行测试、回归测试、编写测试结论或报告、协助 PO 进行 story 验收的工作量等直到某个 story 完成时的所有工作）。 6. PO 根据项目交付期望，确定迭代周期，划分迭代版本、分配用户故事；同时必须邀请版本计划决策干系人确认。 7. 版本计划输出后，PM 进行迭代前的计划安排，包括人力资源到位情况、环境准备情况、项目管理工具就绪情况。 8. PM 组织项目团队，识别项目风险和问题，并制定相应的行动计划和应对措施，并记录到《风险和问题管理跟踪表》。
活动要求	1. 需求讲解、澄清和反讲尽量控制在 4h 之内。 2. 用户故事优先级不晚于版本计划活动确定，同时需要考虑架构、技术风险。 3. 版本计划至少应包含： 优先级、迭代阶段、用户故事负责人、总体工作量、估计开发工作量、估计测试工作量；项目管理活动安排。 4. 工作量预估算，需要综合考虑可投入人力资源和预期时间目标；估算方法可采用人天估算法和故事点估算法。 5. 版本计划需要和利益相关人充分沟通并达成一致，同时以会议纪要形式将版本计划发给相关干系人，并作为版本计划的约束条件。
补充说明	1. 版本计划应遵循用户价值和风险优先的原则，优先级的判定原则： 1) 对客户的重要程度 2) 业务流程 、依赖关系 3) 架构风险 、技术风险，尽量进行风险定量分析，至少进行风险定性分析 4) 工作量大小

	5) 需求的稳定度 其中 1)和 2)属于第一关注点，3)属于第二关注点，4)和 5)属于第三关注点。 2. 估算方法推荐使用人天估算法。 3. 用户故事调整原则参照用户故事拆分过程指引。
方法与工具、IT	人天估算法、用户故事点估算法
备注	

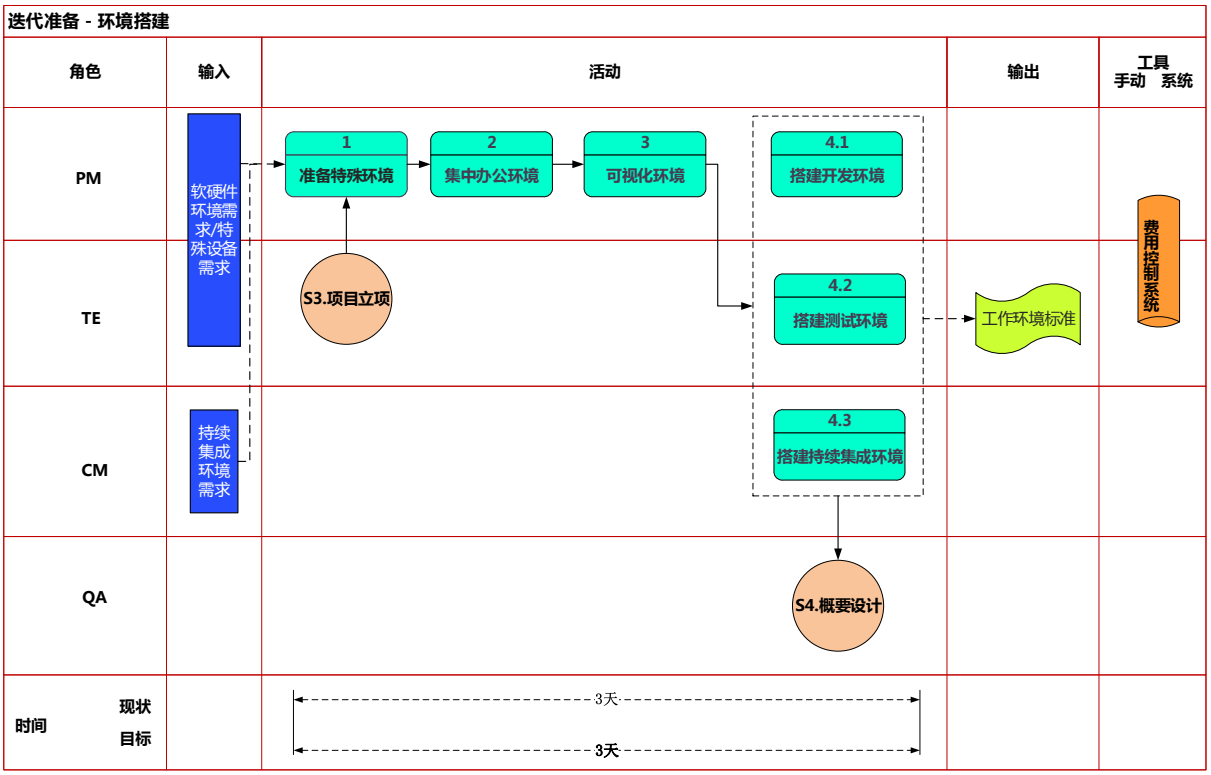
2.1.3. 项目立项



活动名称	**项目立项**
活动目的	1. 确定项目的成本、人力资源； 2. 获得项目组成员对项目目标、版本计划、风险和问题、制度和规范的

	一致认同； 3. 获得组织管理层的认可及审批。
责任角色	PM
参与角色	PO、PM、UE/UI、SE、TE、SWE
输入	1. 用户故事列表_总体版本计划 2. 客户需求列表
输出交付件	开工会 PPT、开工会会议纪要
活动描述	1. PM 根据《国泰安产品版本命名规范 V1.1》的要求确定项目名称及版本。 2. PM 根据项目立项流程审批指引，在 DevSuite 上提交项目立项申请，并正确填写项目立项申请的相关信息，附上《客户需求列表》、版本计划。 3. PM 根据项目的团队成熟度、项目的规模等决策是否需要召开开工会。原则是普通及以上项目（大于 10 人/月）必须召开开工会，其他类型可选。 4. PM 编制《开工会 PPT》，召集项目团队及关键项目干系人召开开工会。 5. 开工会主要内容包含如下： 1) PM 介绍本项目迭代计划、项目组成员及角色； 2) 明确项目的迭代目标、运作机制要求 3) 识别项目现存的问题与风险，并制定相应解决措施。 6. 输出《开工会会议纪要》，并归档 SVN 库。
活动要求	1. 项目的所有信息须填写完整。特别是： 1) 定制项目需包含合同金额、合同编号等信息； 2) 涉及转准生产环境项目必须明确运维人员，并确定是否出具准生产监控报告等。 2. PM 确保信息完整性及准确性，QA 进行立项信息复核。 3. PM 跟进项目立项申请审批人的审批情况。 4. 确认所有项目成员及关键干系人参会； 5. 会议时长控制在 1h 内，尽量避免过多的问题讨论。
补充说明	
方法与工具、IT	DevSuite 指引——项目立项
备注	

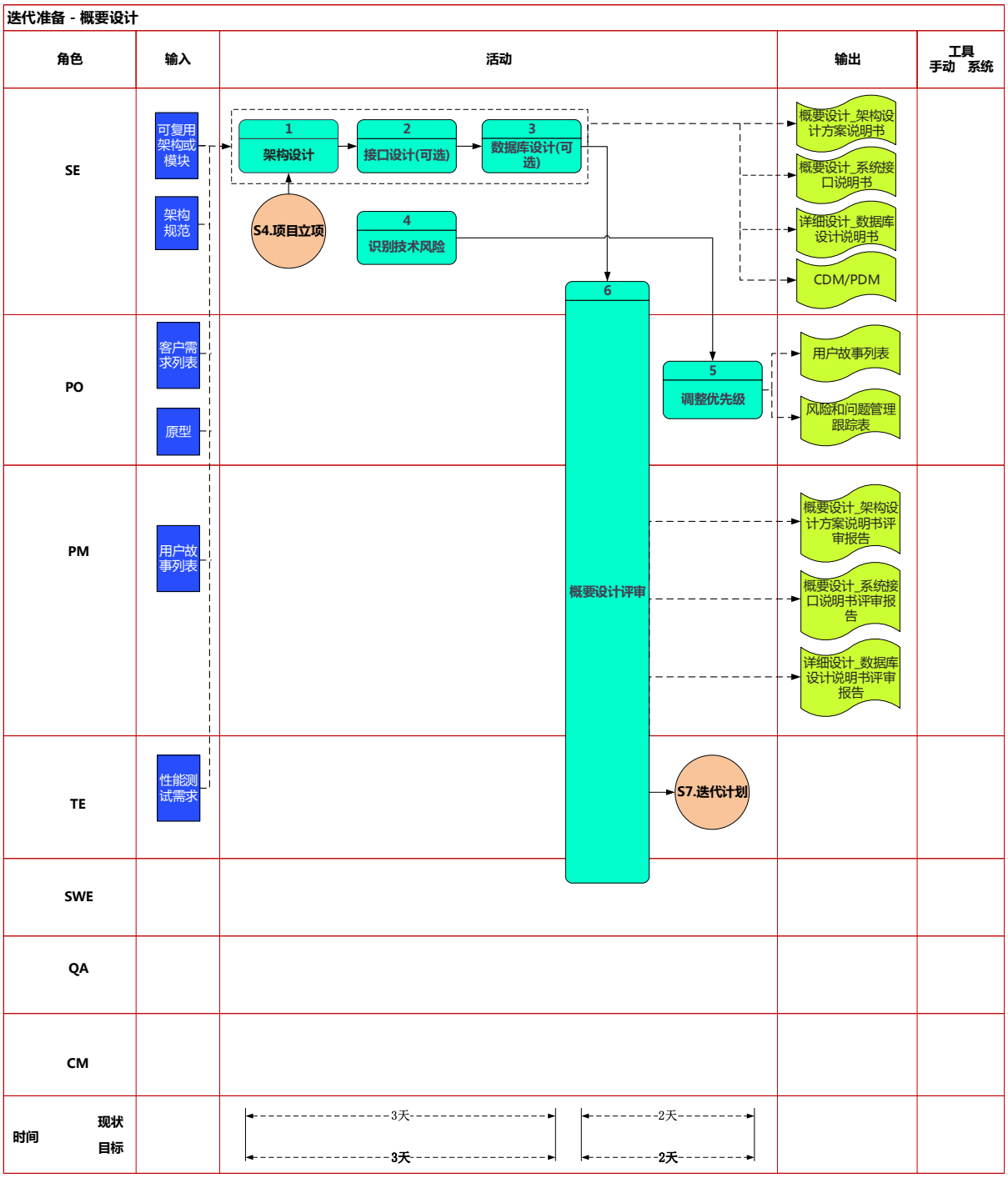
2.1.4. 环境搭建



活动名称	**环境搭建**
活动目的	总原则：所有开发人员、测试人员的工作环境尽量保持一致，如服务器配置、客户端版本等。 1. 组建集中办公环境，包括工位和办公电脑； 2. 组建项目可视化环境，包括白板、任务墙、贴纸、签字笔、磁铁、扑克牌、长直尺； 3. 组建开发环境，包括 SVN 客户端、开发服务器、数据库、开发人员开发软件； 4. 组建测试环境，包括测试服务器（操作系统、数据库）、插件； 5. 组建项目特殊环境，包括 zspace、kinect、HTC vive、Hololense。
责任角色	PM
参与角色	TE、CM
输入	1. 软硬件环境需求/特殊设备需求 2. 持续集成环境需求
输出交付件	工作环境标准
活动描述	1. PM 负责集中办公环境的申请，如果是跨地域团队，则由 PM 委托人负责异地办公环境的申请。对接行政管理部。 2. PM 负责项目可视化环境的申请，如果是跨地域团队，采用 DevSuite 线上可视化环境，项目预立项后线上可视化环境会同步构建完成。如果是同地域团队，采用线下可视化环境，对接行政管理部。 3. PM 负责开发环境的搭建，开发人员协助，所有开发人员的所有工作环境需保持一

	致，且必须与评审通过的软硬件环境需求保持一致，如数据库服务器配置、web 服务器配置等。 4. TE 负责测试环境的搭建，开发人员协助，所有测试人员的工作环境需要与评审通过的软硬件环境需求保持一致，如数据库服务器配置、web 服务器配置等。 5. CM 负责持续集成环境的搭建。 6. PM 负责特殊环境的搭建。
活动要求	1. 项目集中办公环境在项目立项审批之前或预估算时启动，最晚需要在项目立项审批完成时完成。 2. 项目可视化环境在项目立项审批同时启动，包括白板、贴纸、签字笔、磁铁、扑克牌、长直尺。最晚需要在项目立项审批完成时完成。 3. 项目的开发环境，测试环境，项目的持续集成环境需在迭代计划开始之前完成。 4. 项目特殊环境在可行性评审通过之后启动，最晚需要在使用之前完成。
补充说明	
方法与工具、IT	DevSuite
备注	

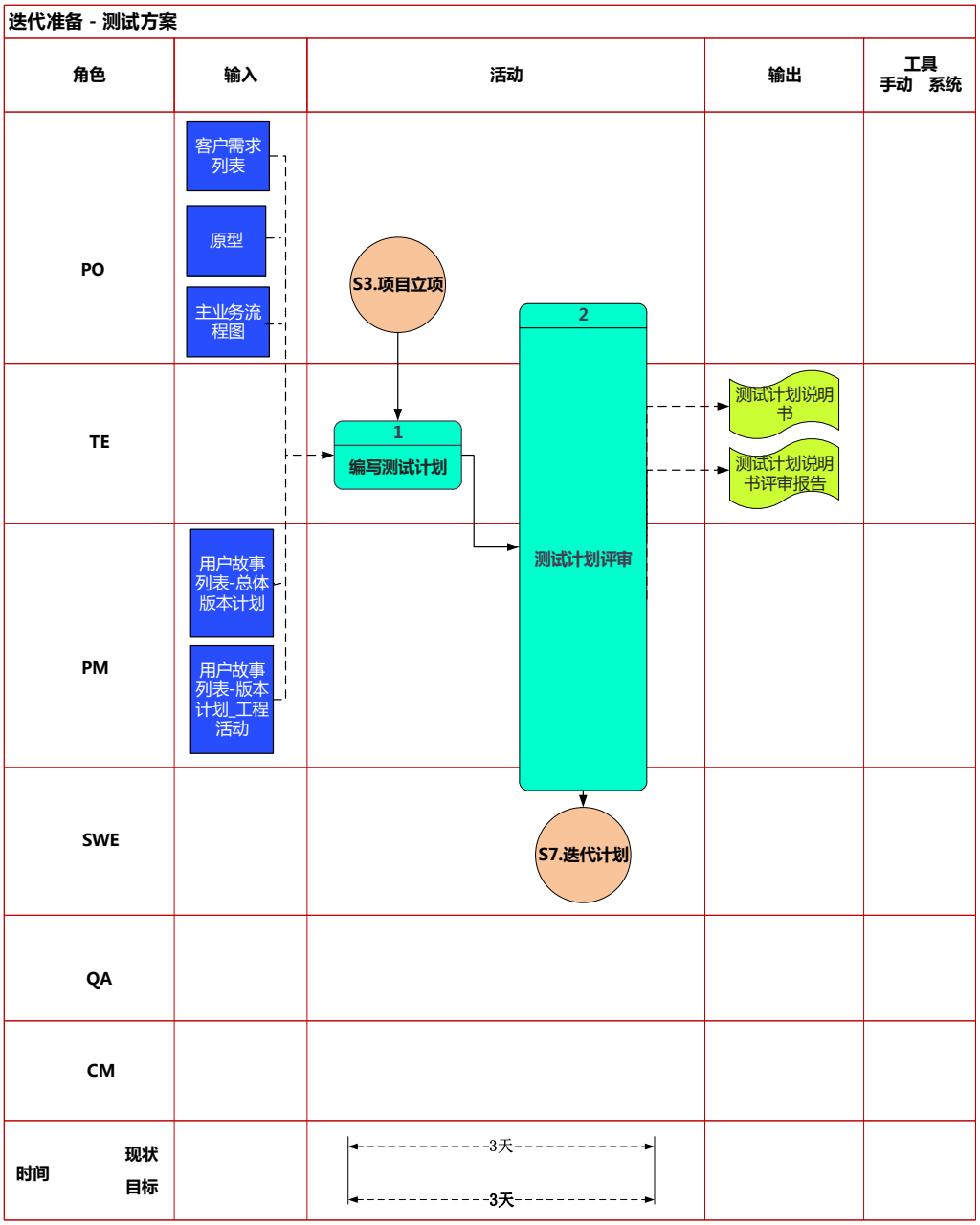
2.1.5. 概要设计（可选）



活动名称	**概要设计（可选）**
活动目的	1. 将客户需求转换为软件结构和数据结构，为详细设计做好基础； 2. 将系统按功能进行模块划分、建立模块的层次结构及调用关系、确定模块间的接口； 3. 便于 SWE 了解系统架构、模块设计、数据结构、接口定义，保证 SWE 从不同角度去观察、理解各个子模块，用于指导后续编程工作。
责任角色	SE
参与角色	PO、PM、SWE、TE

输入	1. 客户需求列表 2. 原型 3. 性能测试需求 4. 用户故事列表 5. 可复用架构或模块 6. 架构规范
输出交付件	概要设计_架构设计方案说明书、概要设计_系统接口说明书、详细设计_数据库设计说明书、用户故事列表、CDM/PDM、风险和问题管理跟踪表、概要设计_架构设计方案说明书评审报告、概要设计_系统接口说明书评审报告、详细设计_数据库设计说明书评审报告
活动描述	1. SE 根据《客户需求列表》进行整体分析，从技术角度将客户需求转换为软件结构、数据结构、接口定义，从而确定软件、数据和接口的总体框架。设计时需遵循公司的架构规范，并同步考虑公司可复用架构或模块，提高复用率，降低开发成本。 2. 从架构、技术风险的角度协助 PO 进行用户故事优先级的综合评估，并协助 PM 进行风险管理。 3. PM 组织 PO、SE、SWE、TE 进行概要设计评审。从设计易用性、可测性、复用程度、设计与需求关系等多维度，共用讨论，确定完整的设计文档。
活动要求	1. SE 需要与 PO、SWE 保持密切沟通，一方面是保证 SE 完整并准确地理解客户需求，另一方面是保证 SWE 按照 SE 的架构设计进行编码实现。 2. 根据客户需求，SE 将系统整体分解更小的子系统和组件，从而形成不同的逻辑层和服务。 3. SE 确定各层的接口 ，层与层之间的关系。不仅要对整个系统分层，即“纵向”分层，还要对同一逻辑层分块，即“横向”分层。 4. 尽量复用公共基础模块。 5. 架构设计需包含：逻辑视图、开发视图、运行视图、物理视图
补充说明	
方法与工具、IT	PowerDesigner
备注	

2.1.6. 测试方案

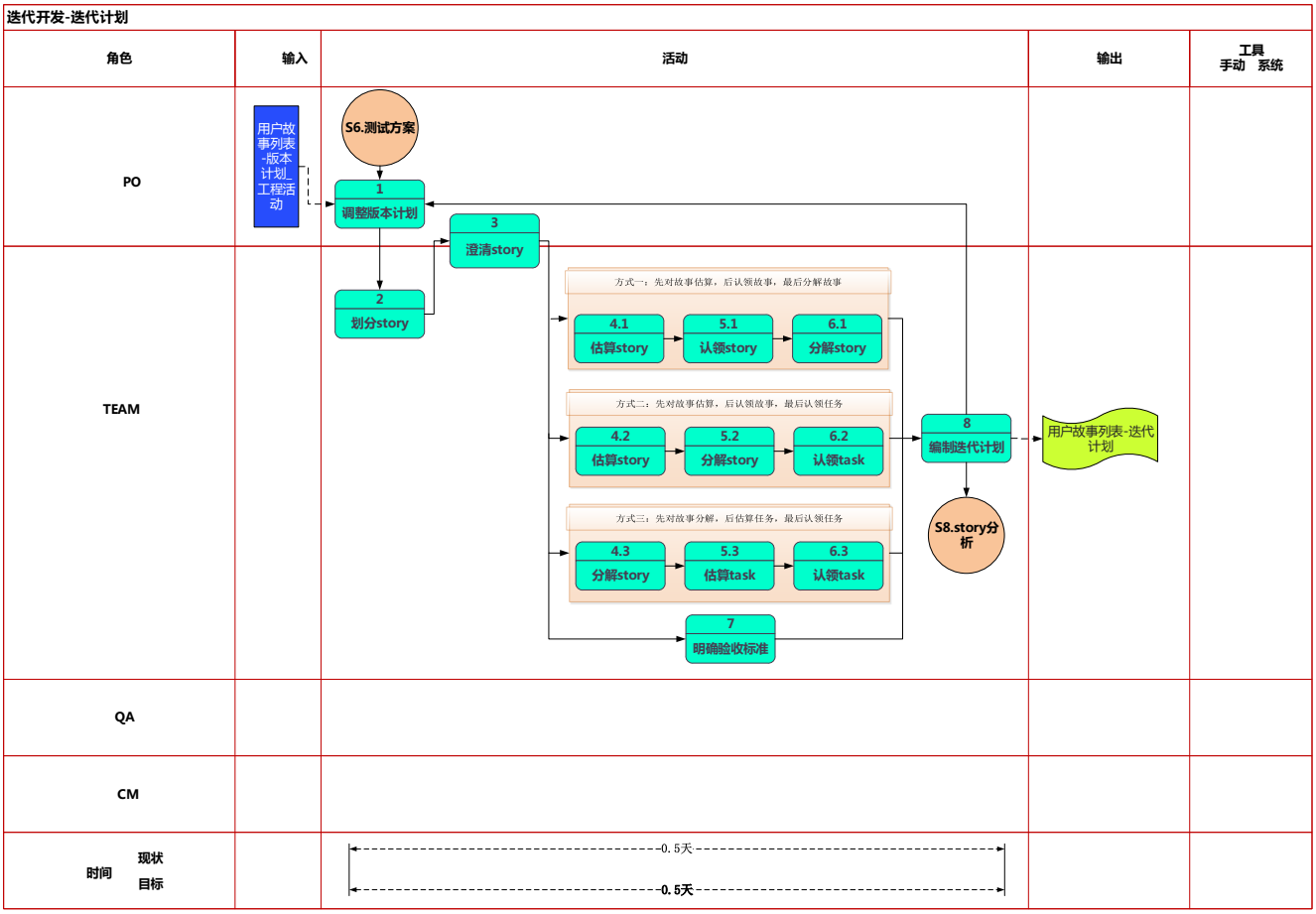


活动名称	**测试方案**
活动目的	1. 根据版本计划，确定总体测试时间安排并明确每个迭代测试时间安排； 2. 确定每个版本测试方案； 3. 确定每个版本测试人员及工作分配； 4. 风险和问题识别并制定应对计划。
责任角色	TE
参与角色	PO、PM、SWE、UI
输入	1. 客户需求列表 2. 原型 3. 主业务流程图

	4. 用户故事列表-总体版本计划 5. 用户故事列表-版本计划_工程活动
输出交付件	测试计划说明书
活动描述	1. TE 根据客户需求、版本计划、可投入人力资源状况等，给出初始的测试方案。 2. TE 组织 PO、PM、SWE、UI 评审测试方案。
活动要求	1. 测试计划在项目过程中不对外部客户公开，随着迭代的进行，可能存在测试策略及方案的调整； 2. 测试计划内容至少应包含： 1) 总的测试目的及范围； 2) 各个迭代的测试目的及范围； 3) 总的测试策略及开始时间和结束时间； 4) 各个迭代用到的测试方案及开始时间和结束时间及人员计划； 5) 各个迭代测试的风险及应对方案； 6) 迭代变化的应变方案。
补充说明	1. 测试计划需要简洁明了。 2. 针对可以预见的风险，提前给出应对方案。
方法与工具、IT	
备注	

2.2. 项目级敏捷迭代开发

2.2.1. 迭代计划



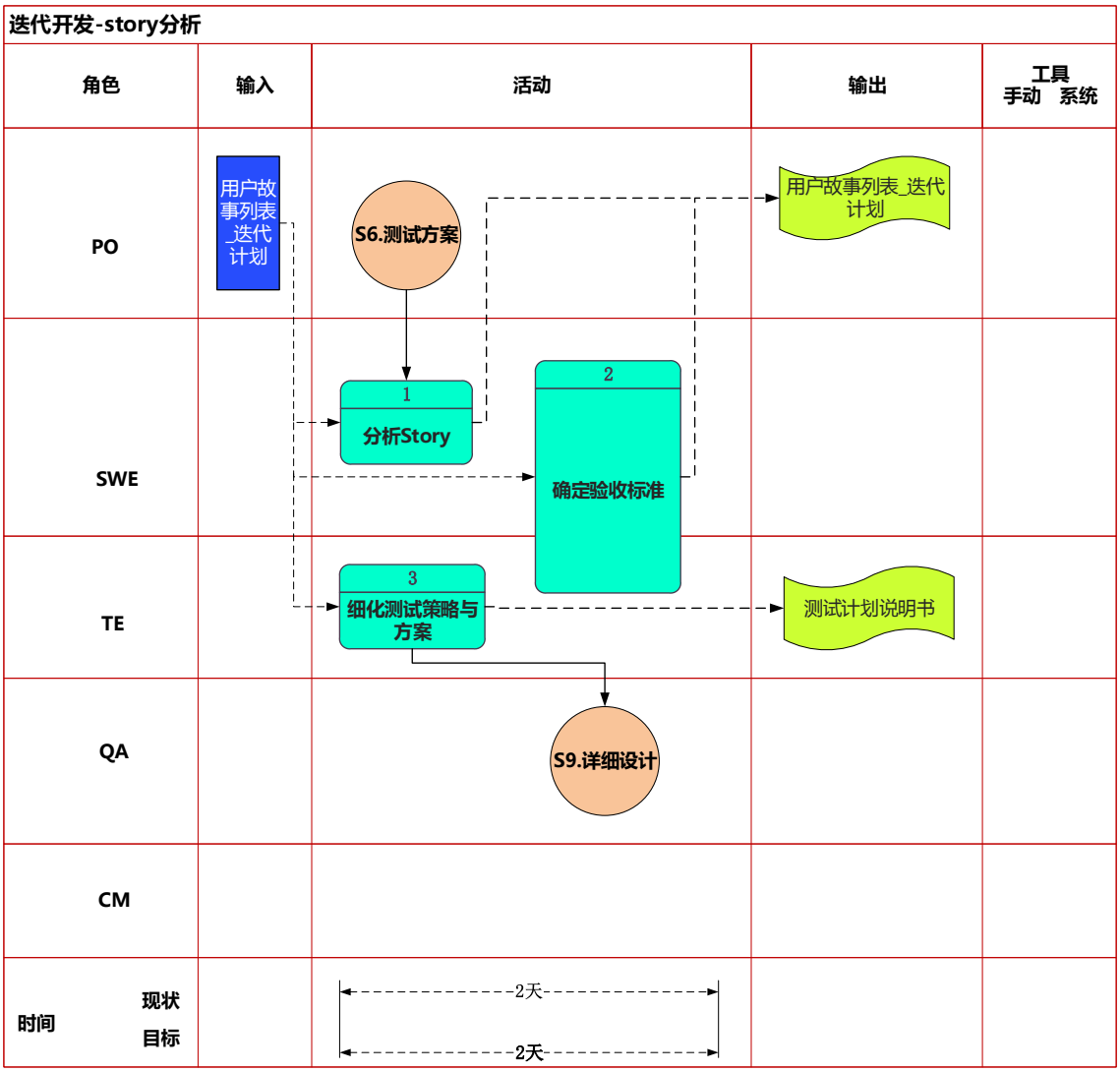
活动名称	迭代计划
活动目的	1. 根据当前情况，调整版本计划； 2. 进行 story 拆分或合并，并确定 story 集成/开发顺序。 3. 将当前迭代开发的 story 列表细化为项目组成员的工作任务计划。 4. 明确 story 的验收标准。 5. 明确项目组要交付的 story 的交付验收计划，保证尽早验证，尽早发现问题、解决问题，以便各 story 按要求高质量交付，避免带病迭代。
责任角色	SM
参与角色	PO、SE 、SWE、TE
输入	用户故事需求列表
输出交付件	用户故事列表_迭代计划
活动描述	1. PM 根据上轮迭代完成情况、现有人力资源、需求变更情况（可能有更高优先级的需求插入），及时调整并确定当前迭代计划，并和 SE、TE、SWE 达成一致意见。 2. 调整如涉及利益相关人，PM 需及时沟通。

	3. PM 重新评估风险和相应的应对措施。 4. PM 组织 PO、PO、SE 、SWE、TE 对本次迭代的 story 进行划分讨论，并最终达成一致意见。同时明确 story 集成/开发顺序。 5. 先由 PO 对有疑问的需求进行澄清，并明确本轮迭代 story 的验收标准。 6. 然后进行工作量确定性估算（不仅包括开发工作量，还要考虑 story 测试、story 验收等活动的工作量）、工作分解以及工作认领事宜。 7. PM 根据工作的分配情况编制详细的迭代计划并取得所有参与人员的一致认同。
活动要求	**1. 讨论故事：** 在迭代计划会议中，团队获得一个已经排好优先级的故事集合，以此作为选代计划会议的输入，会议开始时，客户从最高优先级的故事开始，读给开发人员听。然后由开发人员提问，直到他们充分理解故事，能从故事中分解出任务。没有必要理解故事的所有细节，过分地深入每个故事的细节会让会议变得冗长、低效，因为会议中不是每个人都需要聆听所有故事的所有细节。在计划会议后，开发人员仍能和客户一起理清故事的关键细节。 **2. 分解任务：** 尽管故事的确可以小到作为工作单位，但将它们分解出更小的任务，一般更符合项目的需要。首先，对于很多团队来说，实现故事的开发人员不止一个。需要由多个开发人员共同完成故事，要么是因为开发人员在某些特定技术上的专业性，要么是因为工作划分是完成故事的最快途径。其次，故事是对用户或客户有价值的功能的描述，它们并不是开发人员的待办事项(to-do list）。把故事分解成任务常常是有用的，因为这有助于发现那些可能会被遗忘的任务。 分解准则： 因为故事已经很小了，所以没有必要围绕任务的期望大小设定非常精确的准则．从故事分解任务时，使用以下准则. • 如果故事的某个任务特别难于估算（例如， 系统支持的数据格式列衰， 需要得到远程副总裁的批准），就把那个任务从故事的其余任务中分离出来。 • 倘若有些任务可以很容易安排给多名开发人员共同完成， 就分割它们。 • 若有必要让客户了解故事某一部分的完成情况，可以把那部分拿出来作为一个任务。 **3. 承担责任：** 一旦确定故事的所有任务，就需要有团队成员自愿执行每个任务。如果任务写在自板上，开发人员可以简单地把自己的名字写在他们认领的任务旁边。 实际上，确保完成任务是团队中每个人的责任。团队要有一种“同舟共济”的心态。而且，在迭代快要结束时，如果有开发人员不能完成他接受的所有任务，团队中的其他成员应该尽量勇于承担。虽然任务是由每个入认领并承担责任的，但在迭代期间，这并不是一成不变的。在迭代中，随着开发取得进展，他们会更加了解任务，可能有些工作比预想的简单，但有些却比预想的困难，因此任务的认领及承诺也需要做出调整。在迭代

	结束时，不应该有人说“我完成了我的工作，但是 Tom 还有一些任务没有完成”。 **4. 估算并确认：** 假如一个项目团队的速率是每轮迭代 40 个故事点，那么不断重复前面的步骤一一讨论故事、分解任务直到团队讨论完客户提供的前 40 个故事点的故事。这时每个开发人员负责估算自己承担的工作量。最好的方法仍是以理想时间来估算。 一旦开发人员估算好自己的每个任务，就需要把这些估算加起来，进而做出实际的评估，看看在迭代中能否完成所有的任务。如果有团队成员还承担了其它项目的工作，他没有把握在此项目迭代时间内完成相应的任务，这种情况下： • 留着所有任务，寄希望于一切顺利。 • 请求团队中其他成员接手一些我的任务。 • 与客户讨论，放弃一个故事（或者分割故事，然后放弃其中一部分）。
补充说明	1. 迭代计划会议是客户为团队调整故事优先级的最佳时机，在开始迭代计划的时候，将上一迭代遗留问题，story 重新分析，调整优先级，修改版本计划。 2. 任务的大小没有强制的范围，一般每个 task 的最小单位为 0.5 天，即工作量为 0.5 的倍数，不能有 0.8，1.7 等值。Task 最好为 0.5~2 天。 3. 每个任务都有开发人员承担。 4. 对于三种任务分配方式的选择： 第一种：对故事先估算，后再认领故事，最后分解任务。 选择这种方式的场景：比如一些用户故事已经很小，不能再拆分成更小的用户故事，而且每个用户故事功能简单，不需要再拆分为 task 的情况。 第二种：先估算 story，然后分解 story，最后认领 task 选择这种方式的场景：如果对用户故事所涉及的功能，逻辑关联等已明确，直接估算用户故事。估算完后用户故事很大，则需要分解为更小的用户故事。但分解后的用户故事功能复杂，需要将用户故事分解为更小的 task，以便认领 task。 第三种：先分解 story，然后估算 task，最后认领 task 选择此方式的场景一般是用户故事过于复杂，开发测试很难直接估算此 story 的工作量，这时候需要进一步的先分解 story，然后根据分解后的 task 估算工作量，最后认领任务。 5. 开发人员职责 • 负责参加迭代计划会议。 • 负责帮助把所有故事划分为任务，而不只是自己想做的故事。 • 负责为认领的任务承担责任。 • 负责确保承担适当工作量的工作。 • 在整轮迭代计划中，负责监控自己剩余的工作，同时也要监控队友剩余的工作。如果很快就能完成自己的工作，就有责任帮助队友承担部分工作。 6. 客户(客户代表)职责

	• 负责对迭代中包含的故事排列优先级。 • 负责指导开发人员交付他们能提供的最大商业价值。这意味着， 从发布计划设定之后，若有更高价值的故事，要负责调整优先级以交付最大的商业价值。 • 负责参加迭代计划会议。
方法与工具、IT	扑克牌评估法 ： http://www.uml.org.cn/SoftWareProcess/201108264.asp 看板
备注	提前准备好估算扑克牌

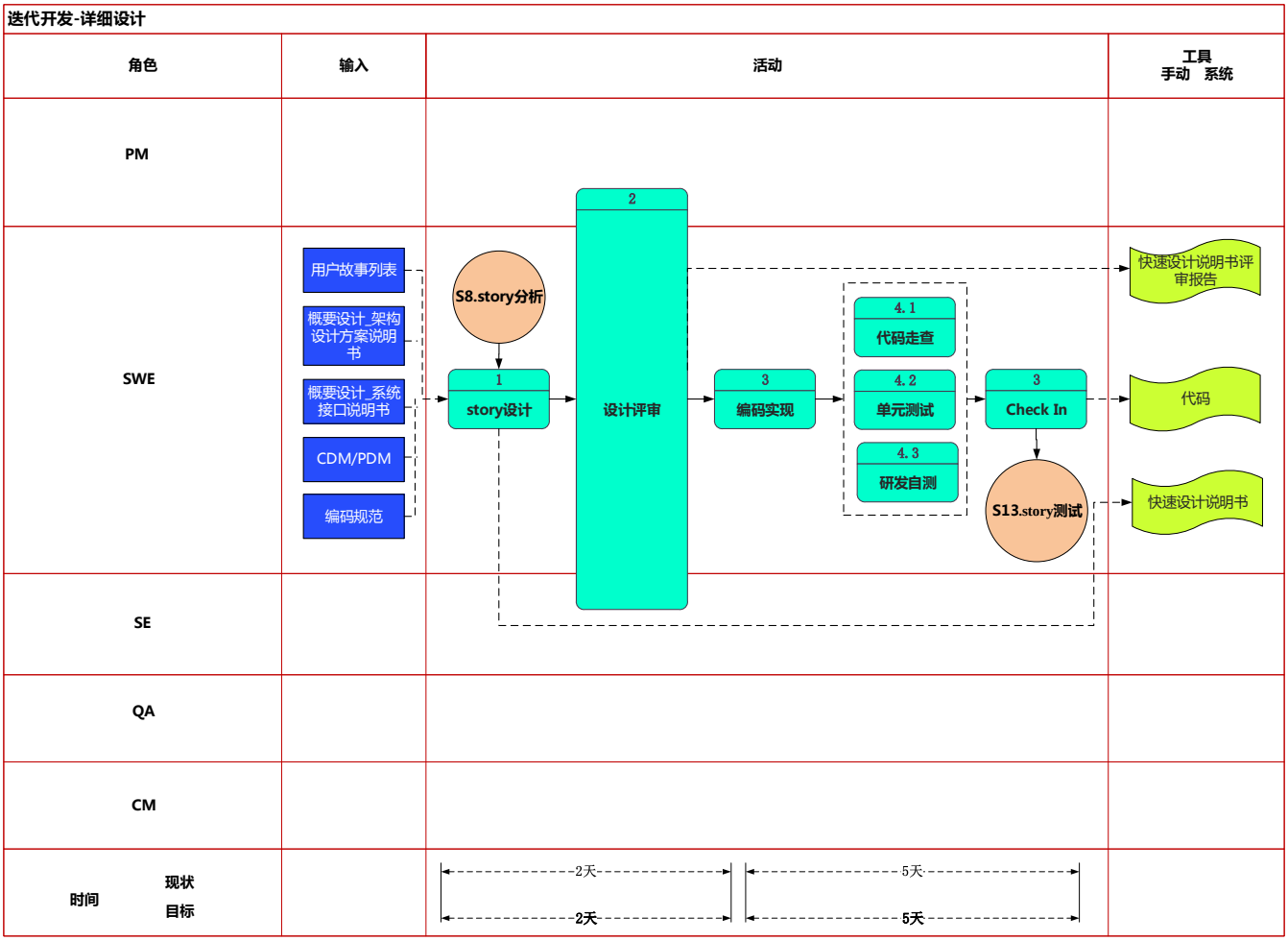
2.2.2. story 分析



活动名称	**story 分析**
活动目的	对 story 细化分析，并确定 story 验收标准，用于指导 story 的开发与验证
责任角色	PO
参与角色	PM、TE、SWE
输入	用户故事列表
输出交付件	story 细节描述（更新用户故事列表）、story 验收标准（更新用户故事列表）、

	细化的测试方案（含测试策略）
活动描述	1. PO 组织 story 相关 SWE、TE、PM 参加，针对 story 功能处理、性能等质量属性、story 间交互影响等细节进行进一步分析，补充和完善 story 细节。 2. SWE、TE 根据分析结果进一步确定 story 的验收标准，并与 PO 达成一致，并同步更新用户故事列表的验收标准。 3. TE、SWE 分析 story 相互影响及对系统原有功能的影响，细化本轮迭代测试策略。
活动要求	1. 要确保所有团队成员对所要开发的任务都已清楚并达成一致； 2. 每个 story 都要有明确的验收标准。
补充说明	
方法与工具、IT	DevSuite
备注	

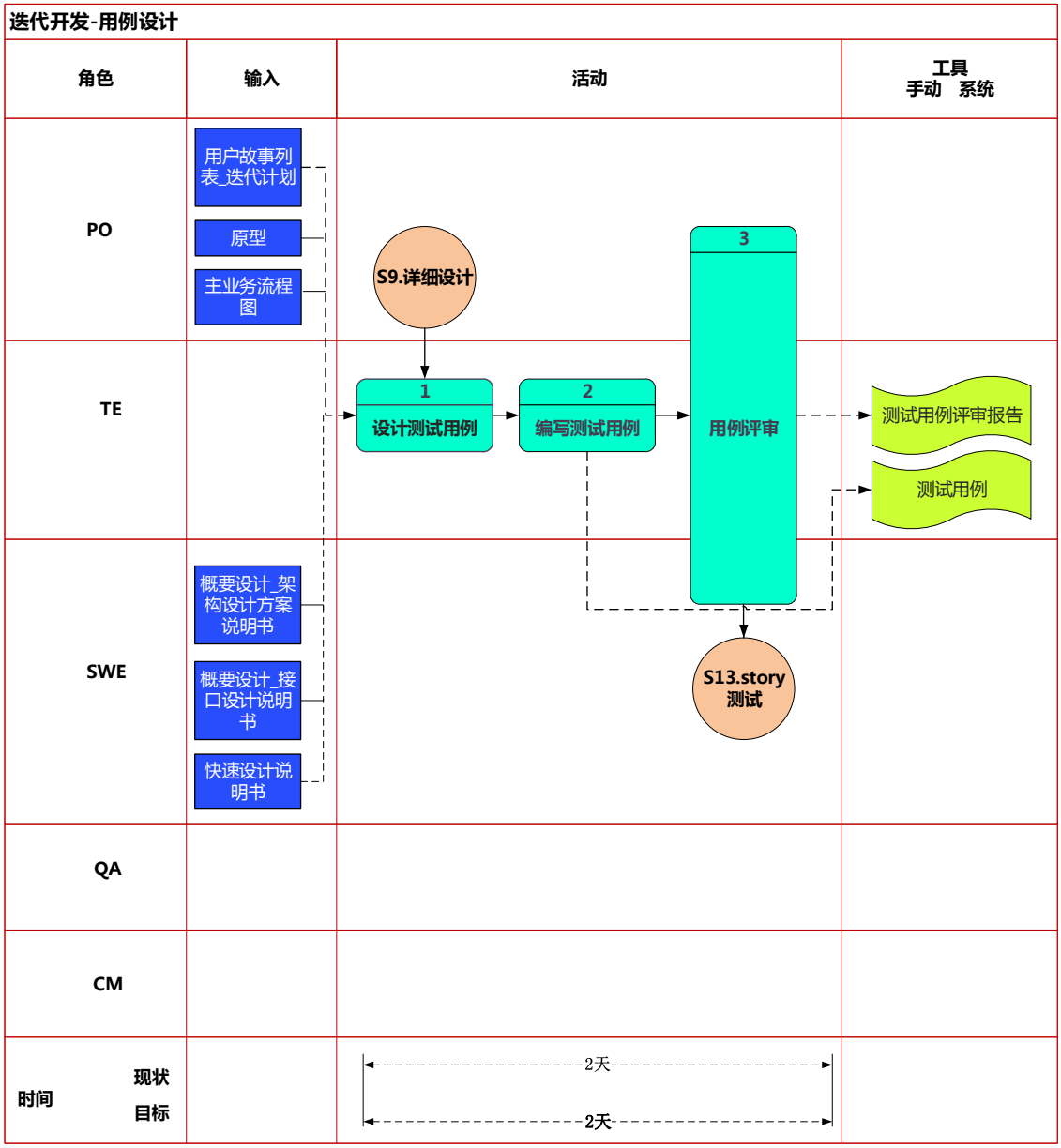
2.2.3. 详细设计



活动名称	**详细设计**
活动目的	SWE 根据 story 及验收标准，进行 story 的详细设计及编码实现工作。
责任角色	SWE

参与角色	PM、SE
输入	1. 用户故事列表 2. 架构设计方案说明书（可选） 3. 编 系统接口说明书（可选） 4. CDM\PDM（可选） 5. 编码规范
输出交付件	快速设计说明书、代码
活动描述	1. SWE 对 story 进行需求再次梳理及 story 设计，输出快速设计说明书。 2. SWE 发起设计评审，PM、SE、TE（可选）对设计文档进行评审。 3. 评审通过后，SWE 进入编码实现活动。 4. 需通过代码走查、单元测试、研发自测环节保证 story 实现的质量。 5. 确保代码无误，check in 到版本配置库中。
活动要求	story 设计需包含：界面设计、模块及接口、逻辑及控制、数据库设计等元素。根据 story 特性选择性输出。 1. SE 在评审过程中，关注 story 设计是否破坏原有系统结构。 2. 编码实现前需将配置库的最新代码更新到本地。 3. story 提交前必须完成 story 编程实现所有活动，通常包括：静态代码检查、代码走查、代码编译链接、单元测试等。如一个 sotry 涉及多个 SWE，需要相关开发人员进行本地联调才能提交 story。 4. SWE 应遵守团队在迭代计划时指定的编码规范。
补充说明	
方法与工具、IT	SVN
备注	

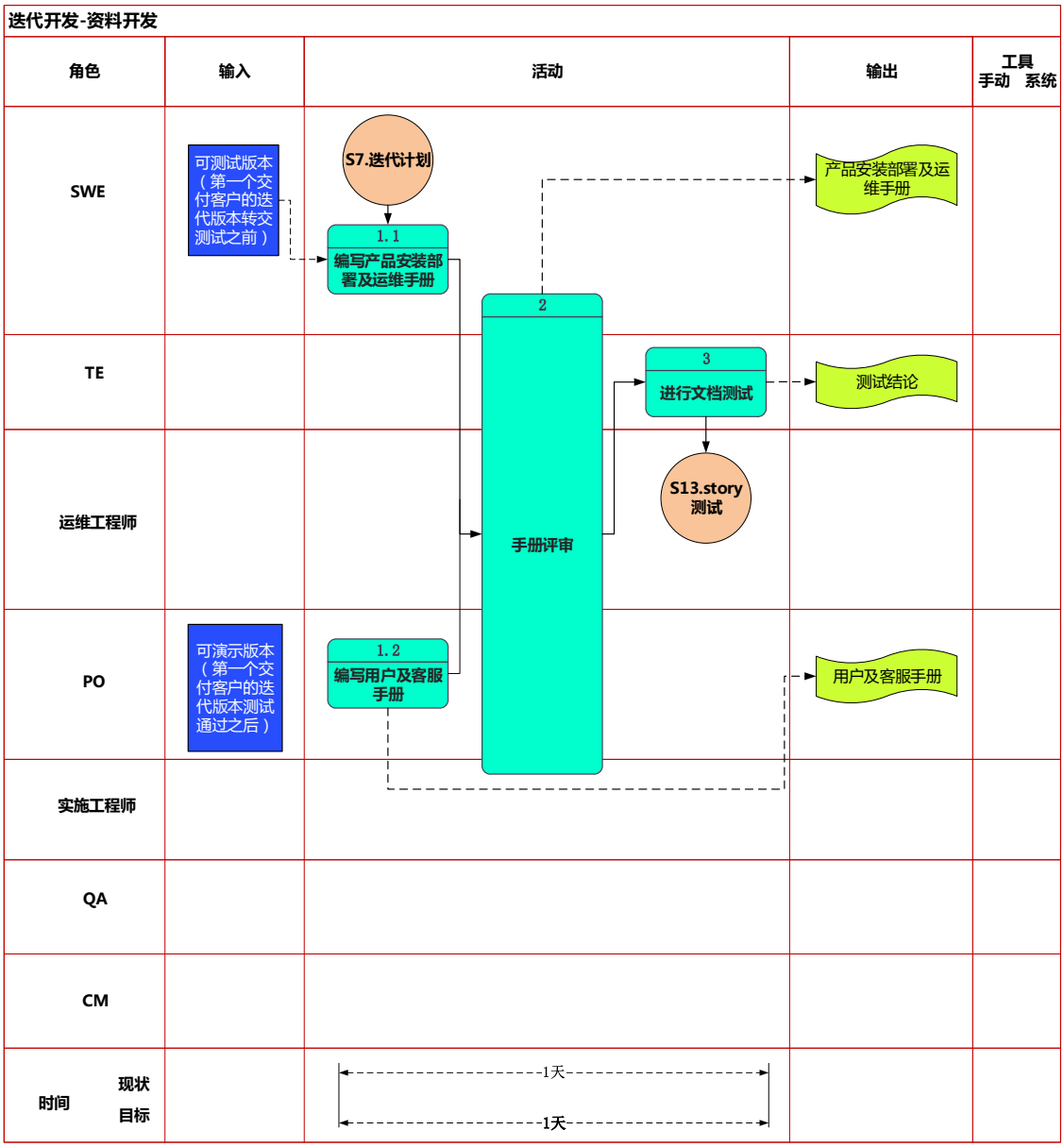
2.2.4. 用例设计



活动名称	用例设计
活动目的	1. 根据概要设计、story 详细设计挖掘出测试点； 2. 根据测试点设计测试用例； 3. 根据主业务流程确定 story 测试用例优先级；
责任角色	TE
参与角色	PO、PM、SE、TE、SWE
输入	1. 用户故事列表_迭代计划 2. 原型 3. 主业务流程图 4. 概要设计_架构设计方案说明书 5. 概要设计_系统接口说明书 6. 快速设计说明书

输出交付件	测试用例、测试用例评审报告
活动描述	1. TE 根据本次迭代的 story 确定测试点，根据测试点设计相关的测试用例，再依据主业务流程及 story 确定测试重点并确定测试用例的优先级。 2. PM 组织 PO、SE、SWE、TE 评审测试用例。 3. TE 和利益相关人沟通，达成一致后，将测试用例基线化，正式发布给所有利益相关人。
活动要求	1. 测试用例在项目过程中不对外部客户公开，随着迭代的进行，可能存在测试用例的变更包括添加、修改或者删除。 2. 测试用例设计总原则： 1) 覆盖当前需求的所有测试点，无遗漏； 2) 测试用例设计思路清晰，结构化明显； 3) 测试用例颗粒度适中，保证一个输入一个输出； 4) 测试用例优先级合适，符合二八原理。
补充说明	1. 复用原则，能公共化的用例设计公共测试用例。 2. 测试用例设计的时间长短根据迭代及项目大小来确定。
方法与工具、IT	
备注	

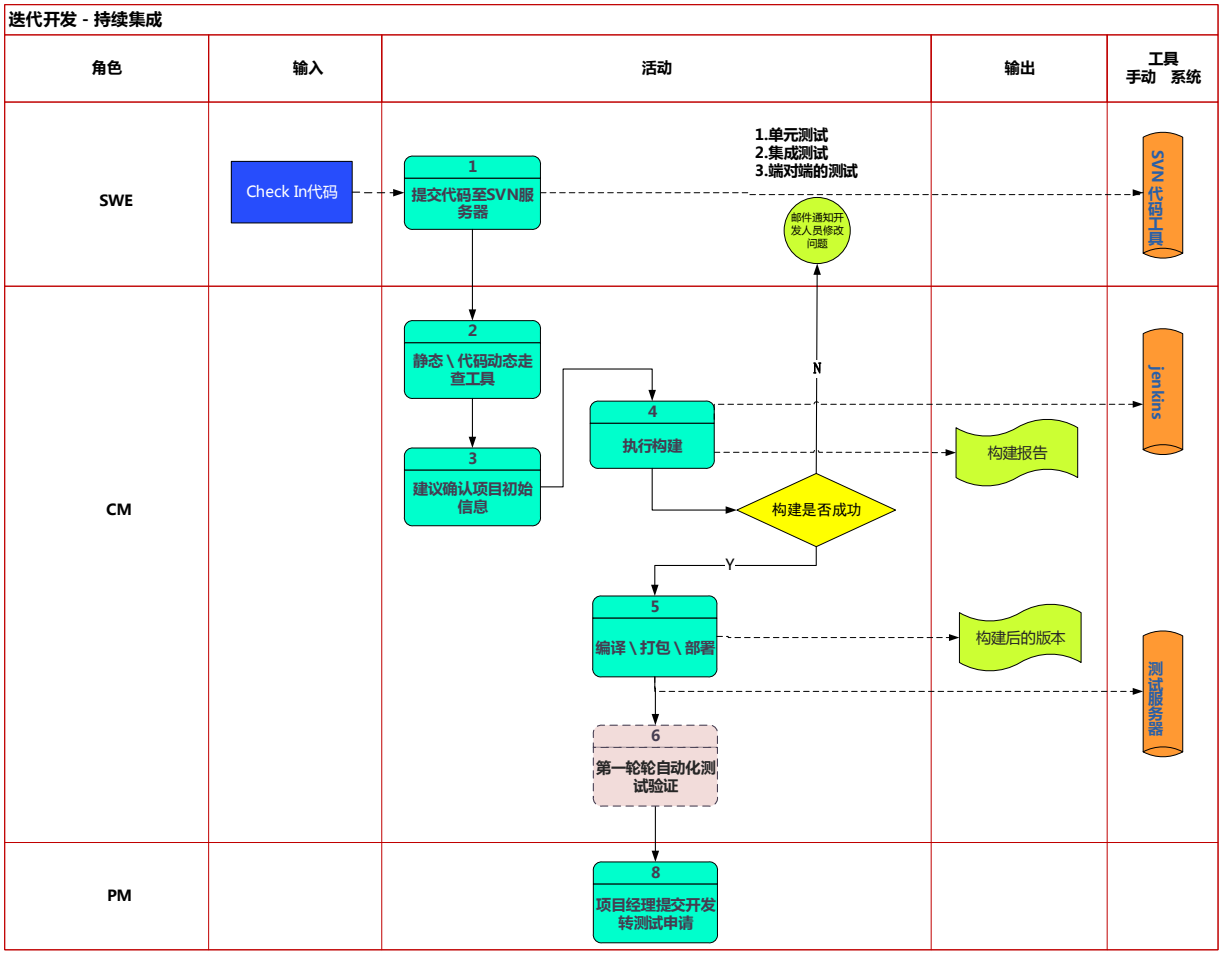
2.2.5. 资料开发（可选）



活动名称	资料开发（可选）
活动目的	1. 输出产品安装部署及运维手册，以保证实施运维团队能够按照指导手册正确部署； 2. 输出用户及客服手册，以保证用户能够参照指导正常使用系统。
责任角色	PM
参与角色	PO、SWE、TE
输入	1. 可测试版本 2. 可演示版本
输出交付件	产品安装部署及运维手册、用户及客服手册

活动描述	1. 当前版本如需交付给客户，可测试版本在提交 TE 之前，SWE 需完成编写《产品安装部署及运维手册》。 2. 开发团队和测试团队评审《产品安装部署及运维手册》，并且 TE 需要对《产品安装部署及运维手册》进行文档测试，给出测试结论，如测试不通过，则 SWE 继续修改产品安装部署及运维手册直至测试通过。 3. 当前版本如需交付给客户，PO 根据可演示的版本，完成编写《用户及客户手册》。 4. 开发团队和测试团队评审《用户及客户手册》，PO 根据评审意见修改，并且 TE 需要对《用户及客户手册》进行文档测试，给出测试结论，如测试不通过，则 PO 继续修改《用户及客户手册》直至测试通过。
活动要求	1. 提交发布客户的迭代版本，必须输出《产品安装部署及运维手册》、《用户及客户手册》。 2. 《产品安装部署及运维手册》需邀请实施团队、运维团队参与评审验收工作。 3. 测试过程的问题录入到 DevSuite 系统中，但不作为系统缺陷率统计。 4. 《用户及客户手册》需邀请实施团队参与评审验收工作。
补充说明	如当前版本不交付客户，可以启动编制产品安装部署及运维手册、用户及客户手册，不强制要求完整输出，但必须在交付客户版本时输出。
方法与工具、IT	DevSuite
备注	

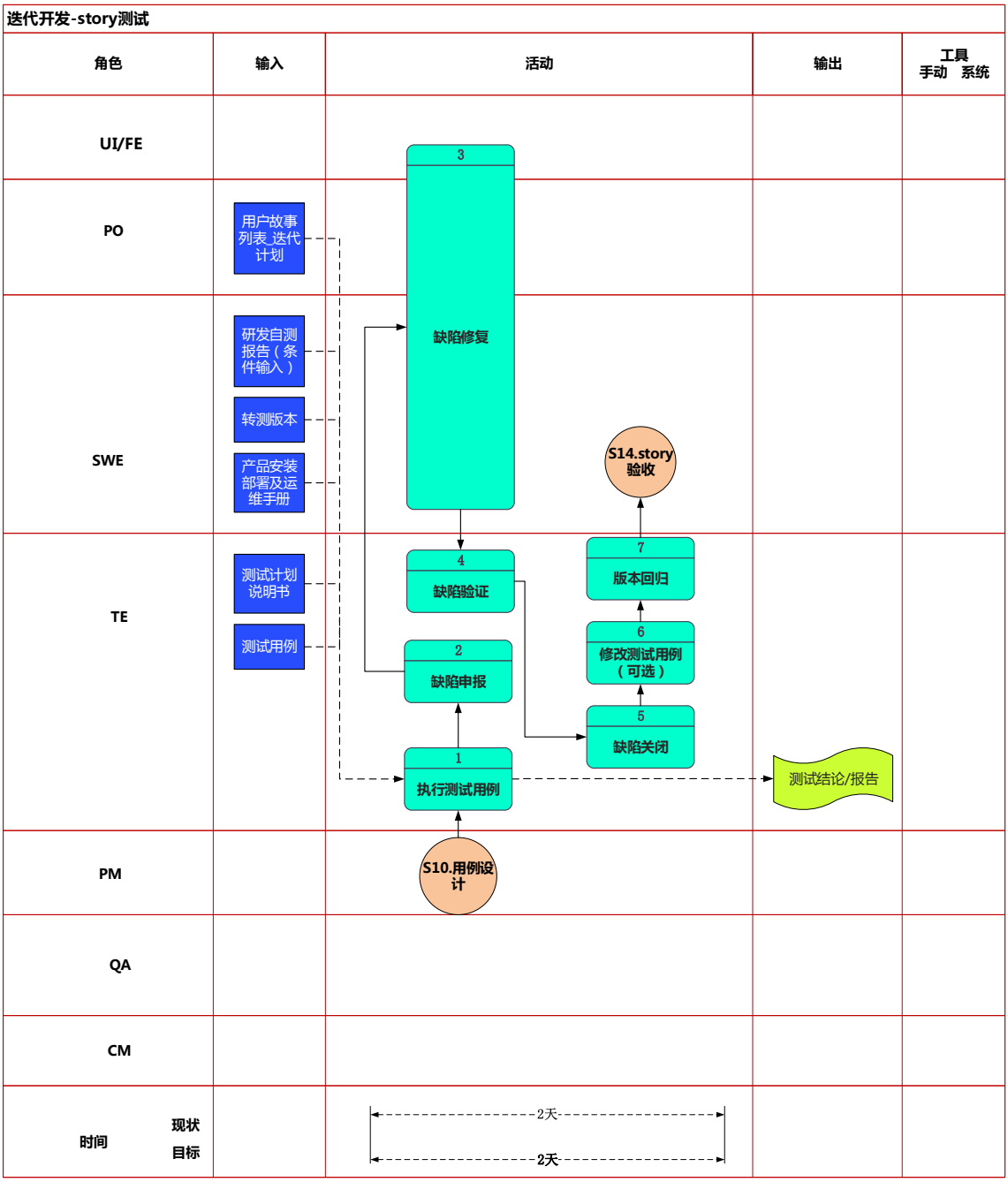
2.2.6. 持续集成（可选）



活动名称	**持续集成（可选）**
活动目的	总原则：持续集成在迭代开发过程中贯穿始终。 1. 高频度进行系统功能的回归，提供产品质量的快速反馈； 2. 缺陷在引入的当天就被发现并解决,避免缺陷的累积； 3. 每天生成可部署，可测试的软件版本，为 story 验收提供输入。
责任角色	CM
参与角色	SWE、TE
输入	代码 check in 触发或者每天定时，一次或者多次或者人工触发
输出交付件	持续集成的结果报告、构建成功，可测试或试用的版本
活动描述	1. SWE 进行本地构建后，合入代码到配置库,并向服务器执行 Check in 的动作。 2. 在 Jenkins 上自动执行静态代码或态代码走查，并将发现的问题反馈给 SWE。 3. 在 Jenkins 系统上配置项目的初始化信息，包括代码库的路径、部署服务器路径等工作。 4. 在 Jenkins 执行第一轮自动化测试用例。 5. 在 Jenkins 执行构建工作，执行成功进入编译、部署，不成功则返回开发组修复问问题，修改完成后重新编译。

	6. 在 Jenkins 执行成功进入编译、部署，将编译好的安装包放到指定的目录下。 7. 在 Jenkins 执行第二轮自动化测试验证。 8. 由 PM 提交开发转测试申请。
活动要求	1. SWE 必须先在本地构建成功，才可提交代码到配置库。 2. 持续集成服务器检测到代码更新或设定集成构建的周期到，自动触发一次构建，大系统的持续集成可分为项目级构建和版本级构建两个层次，项目级构建先进行，成功后再进行版本级构建，构建全程自动化。 3. 集成构建至少要每天构建一次。 4. 持续集成服务器提供构建结果的反馈。 5. 当构建失败，CM 应组织相关项目组修复问题，并协助恢复构建；成功构建生成的版本放到版本服务器，可供测试人员进行 story 测试。 6. 修复失败的构建要求作为团队最高优先级的任务，及时响应；大系统持续集成需分层分级，并建立各层次统一的测试策略
补充说明	1. 如果自动化测试用例数量较多，导致构建时间太长，可将持续集成分为本地构建、集成构建和发布构建，对用例进行分层分级管理。 2. 各级测试用例在持续集成配置库中进行拉通管理与维护。 3. 如果涉及软硬件，鼓励尽早进行软硬件集成。根据依赖关系，当集成时即集成，尽量避免爆炸式集成。 4. 本地构建要包含下载最新版本代码、编译、代码静态检查、圈复杂度检查等）、冒烟测试（系统基本功能用例）、提交代码。 5. 集成构建在本地构建基础上要增加更多用例（如所有特性的基本用例），保证每天都有一次快速的质量反馈。 6. 发布构建要求包含所有用用例，保证测试完备。
方法与工具、IT	工具：持续集成工具、包括编译工具、代码静态检查工具、测试工具等
备注	

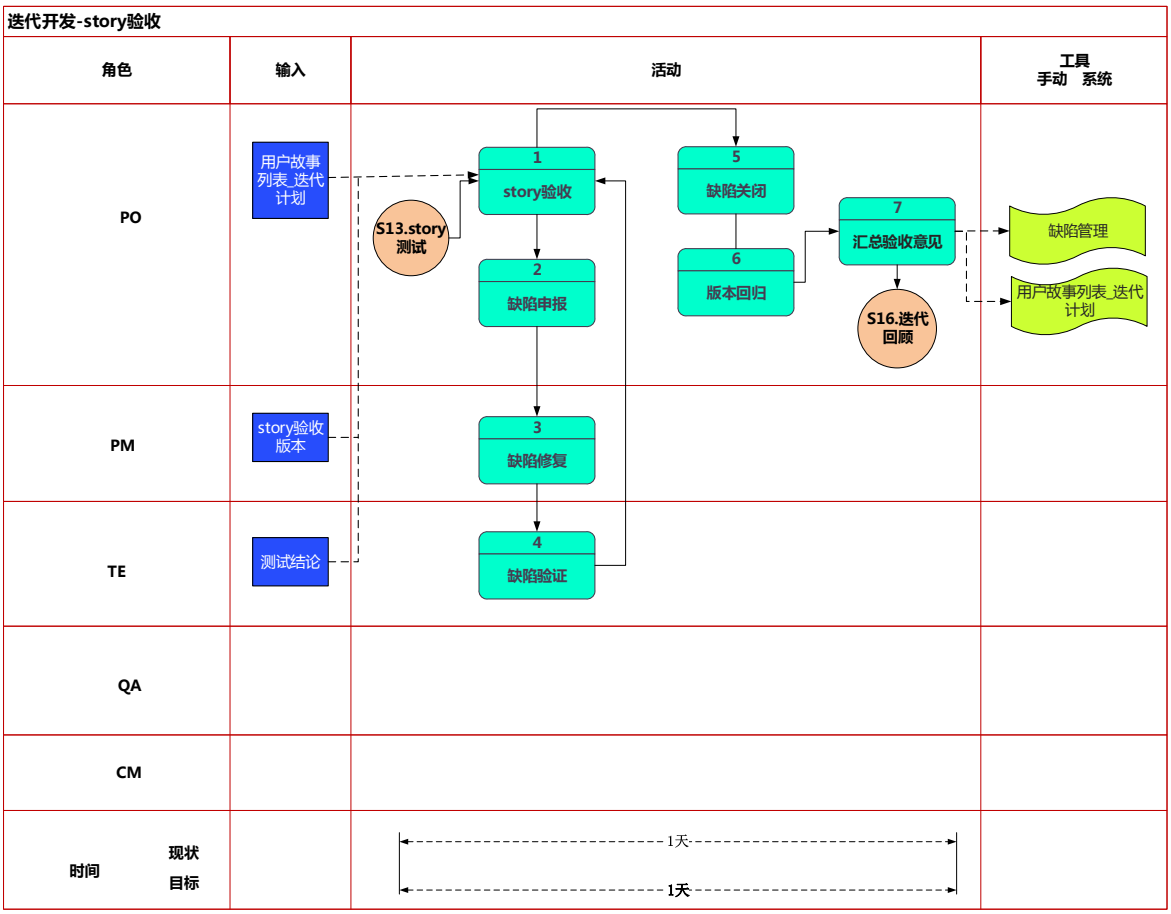
2.2.7. story 测试



活动名称	**story 测试**
活动目的	通过关注 story 的集成测试，保证迭代交付特性可用
责任角色	TE
参与角色	PO、PM、SE、SWE
输入	1. 用户故事列表_迭代计划 2. 转测版本 3. 产品安装部署及运维手册 4. 测试计划说明书 5. 测试用例

输出交付件	测试结论（如果是最后一轮迭代，提供测试报告）
活动描述	1. TE 对本版本开发的所有特性进行整体测试，从基础的前端到后台，从单个功能到整体集成，从功能到性能以及文档/加密测试，发现产品的缺陷并保证缺陷的修复。 2. PM 组织 PO、SE、TE、SWE 确定每一个迭代内提供的版本数量，并确保 SWE 做好研发的自测，提供稳定可测试的版本，TE 通过技术手段发现产品内隐藏的缺陷并保证缺陷的修复，最终给出可以发布的版本，并将版本基线化，正式发布给所有利益相关人。
活动要求	1. 测试所发现的缺陷及测试结论在项目过程中不对外部客户公开； 2. 测试过程中要求： 1) 测试时间的保证，不能因为版本延误等压缩测试时间； 2) 转测质量的保证，对质量不好的版本，测试有打回的权利； 3) 测试环境的保证，测试过程中不许对测试环境进行破坏； 4) 测试的独立性，确保测试过程中不要干扰测试的正常进行。 3. 测试结论（报告）的要求： 1) 测试结论（报告）：测试时间、人员、缺陷数、遗留缺陷数、风险建议； 2) 最后一轮迭代按照测试报告的模板填写并发送报告。
补充说明	1. 测试过程中可能会和 PO 讨论 story 或需求； 2. 测试过程中可能会对测试用例进行添加、修改或删除； 3. 经过讨论，如果研发部门能做好单元测试和持续集成，自测报告可以省略，同时准入用例执行也将省略，直接进入测试用例执行活动。
方法与工具、IT	DevSuite，RF、LR、fiddle 或其他功能、自动化或辅助测试工具
备注	

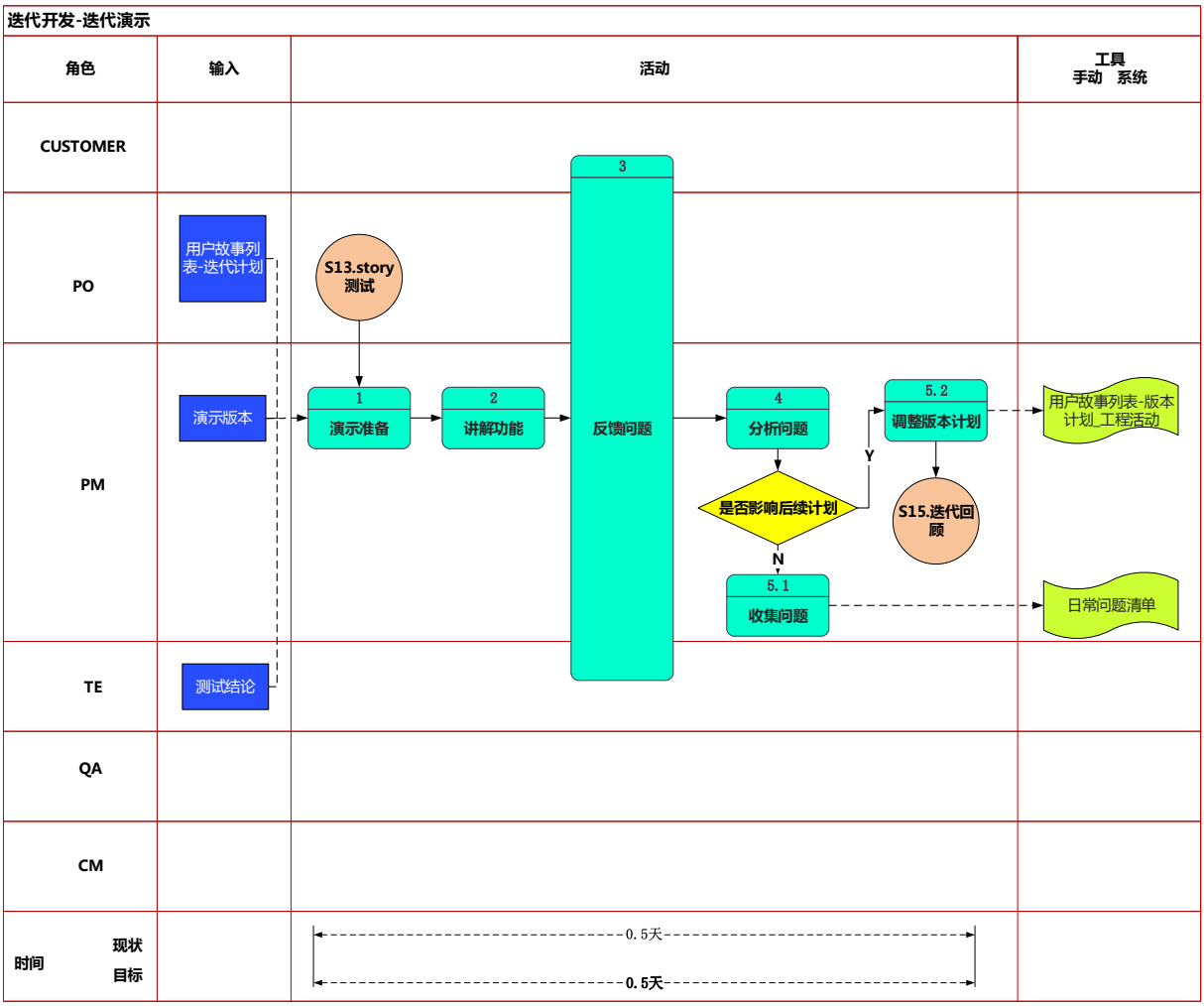
2.2.8. story 验收



活动名称	**story 验收**
活动目的	对完成开发的 story 进行验收，达到 story 的验收标准。
责任角色	PO
参与角色	PO、PM、SE、TE、QA
输入	1. 用户故事列表-迭代计划（验收标准） 2. story 验收版本 3. 测试结论/报告
输出交付件	story 验收结果（更新用户故事列表-迭代计划）
活动描述	1. PO 根据当前迭代开发的 story 列表及 story 验收标准，确定产品设计方向无偏离，大体能够实现总体需求，为后续的 story 集成测试提供理解上的帮助。 2. story 验收过程中发现的缺陷录入到 DevSuite 中，并由 SWE 修复、TE 测试之后交由 PO 验证，关闭缺陷。 3. PO 给出总体的 story 验收意见。

活动要求	1. 测试所发现的缺陷及测试结论在项目过程中不对外部客户公开； 2. story 验收测试过程中要求： 1) 仅仅对 story 提供的标准做相关验收，不做细致测试； 2) 确保 story 设计方向的正确性，为后续 story 集成测试提供基础； 3) 测试环境的保证，测试过程中不许对测试环境进行破坏； 4) 测试的独立性，确保测试过程中不要干扰测试的正常进行。
补充说明	story 验收和 story 测试的活动的区别在于：story 验收重在证实，story 测试重在证伪。
方法与工具、IT	DevSuite
备注	

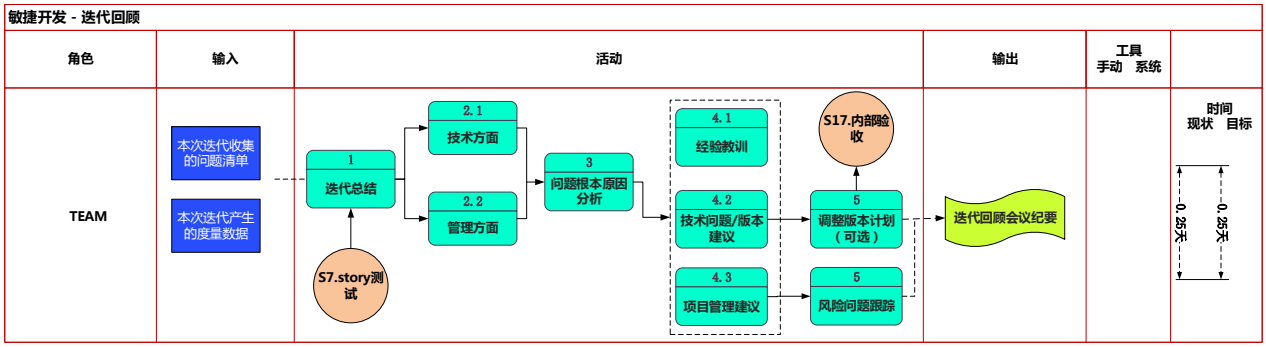
2.2.9. 迭代演示



活动名称	**迭代演示**
活动目的	检验迭代成果，是否完成及满足需求。
责任角色	PM
参与角色	CUSTOMER、PO、PM、UI/FE、SWE、TE

输入	1. 用户故事列表-迭代计划 2. 测试结论 3. 演示版本
输出交付件	日常问题清单、用户故事列表-版本计划_工程活动（更新）
活动描述	1. PM 提前准备迭代演示环境及迭代演示的内容，并确定参与迭代演示的人员。 2. PM 简单描述当前迭代的目标，包含的用户故事列表。 3. PM 开始进行功能演示。同步介绍当前迭代所完成的非功能性需求，比如性能测试、兼容性测试等。 4. 参加演示的干系人针对演示的内容进行问题反馈。 5. PM 负责收集相关干系人的反馈意见，并对反馈意见进行分析。如果是对版本的建议作为调整版本计划的输入，其他改进建议记录到日常问题清单中。
活动要求	1. 确保关键角色参与迭代演示，活动有效完成。 2. 演示内容能够覆盖本次迭代完成的用户故事。 3. 迭代演示仅关注业务层次，不涉及技术细节。注意力集中在“我们做了什么”，而不是“我们怎么做”。 4. 拒绝演示大量细节的 BUG 修复和微不足道的特性。
补充说明	
方法与工具、IT	SVN
备注	

2.2.10. 迭代回顾（可选）

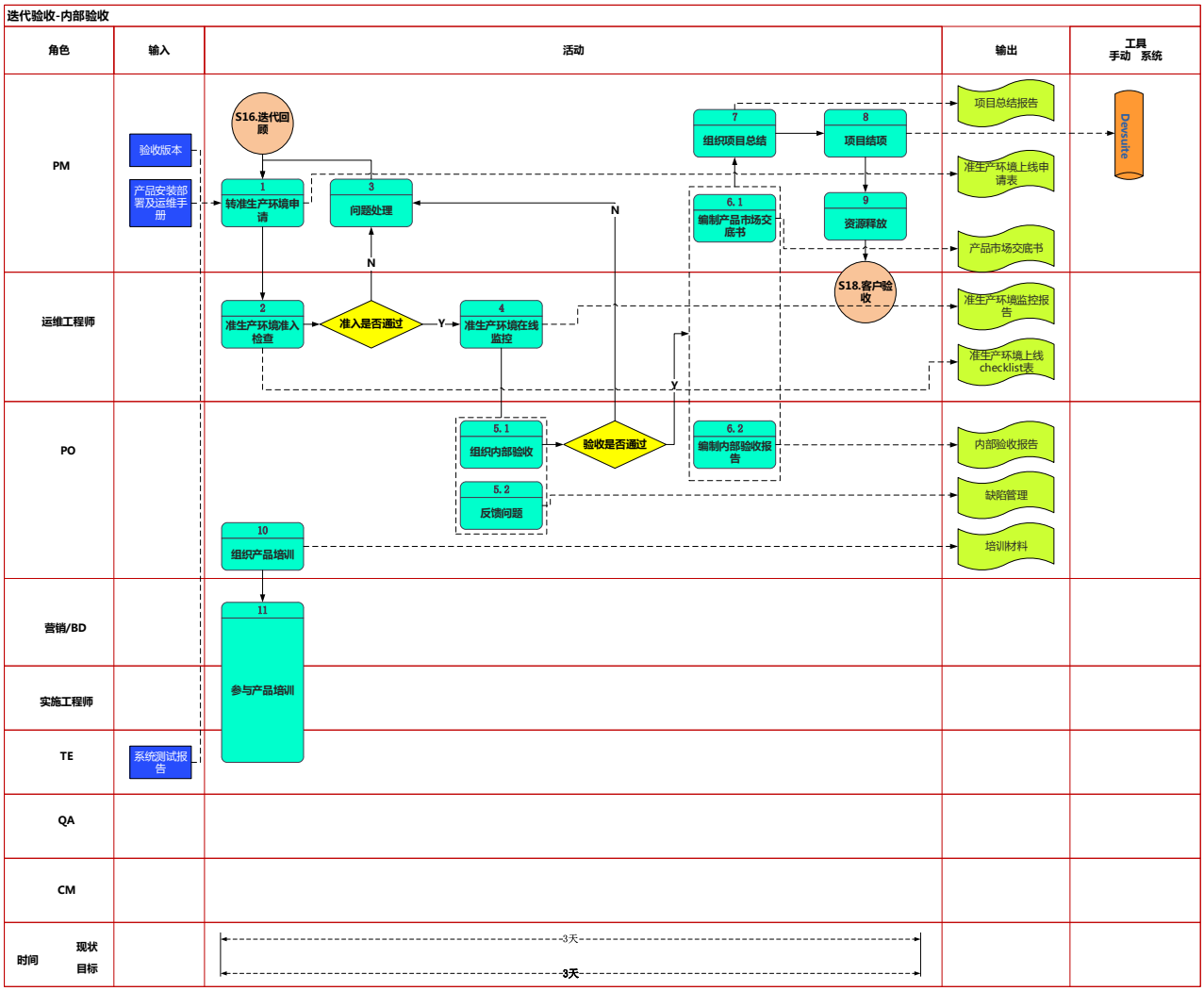


活动名称	**迭代回顾（可选）**
活动目的	回顾本次迭代过程，对好的方面和不好的方面进行经验积累，主要目的有： 1. 帮助所有团队成员包括 PM 进行自我提升； 2. 促进团队成员的相互了解，保持团队旺盛的战斗力；

	3. 便于更好地开展下个迭代。
责任角色	PM
参与角色	TEAM、SM
输入	本次迭代产生的度量数据、本次迭代收集的问题清单
输出交付件	迭代回顾会议纪要
活动描述	1. PM 提前准备好本次迭代产生的度量数据、收集的问题清单。并在 QA 辅助下，进行初步的度量数据分析。 2. PM 组织项目组成员识别技术方面和管理方面的问题要素，对识别出的要素进行优先级排序，选取 1~3 个优先级高的问题。 3. PM 组织项目组成员采用 5W 法对选取的因素进行根本原因分析，总结本次迭代的经验教训，入经验教训库。然后采用头脑风暴法寻找一些改进建议措施，对版本的建议提交给 PM，作为调整版本迭代计划的输入，团队内的改进建议也提交给 PM，作为制定下轮团队迭代开发计划的输入。对于版本建议和团队建议要进行问题跟踪至闭环。
活动要求	1. 回顾会议要全体团队成员参加，自由讨论，气氛宽松自由，畅所欲言，头脑风暴发现问题，共同分析根因。 2. 以迭代项目组为单位召开回顾会议。 3. 会议组织者不一定是 PM，敏捷教练引导，团队成员可以轮流担任。 4. 每个迭代都要做迭代回顾才算迭代结束。 5. 会议结论要跟踪闭环。
补充说明	1. 迭代回顾会议时长在 1h~2h 为宜。 2. 指导书：迭代回顾会议指导书 3. 根本原因分析、5W 法、头脑风暴法：请参见迭代回顾会议使用方法介绍
方法与工具、IT	
备注	

2.3. 项目级敏捷迭代验收

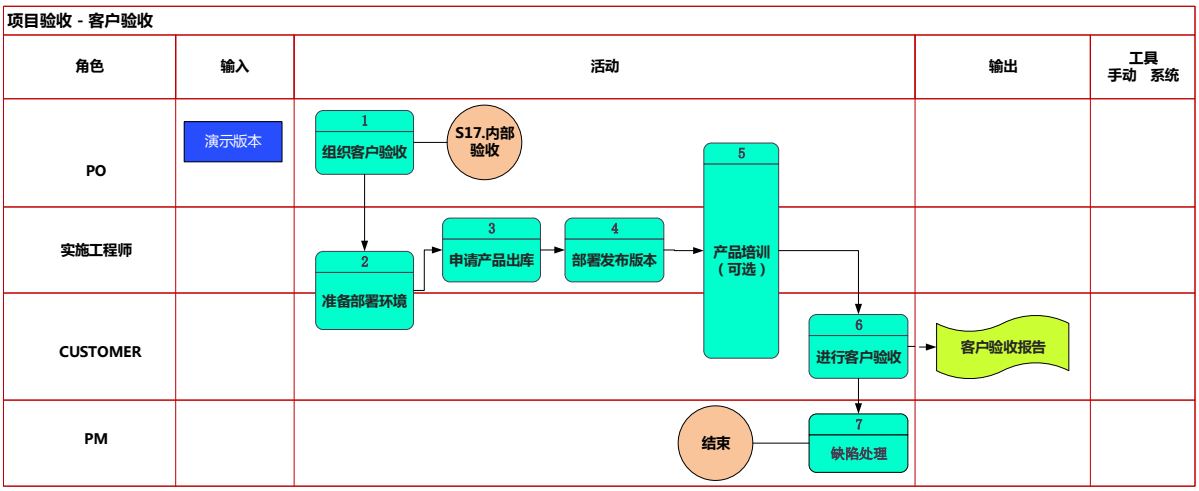
2.3.1. 内部验收



活动名称	**内部验收**
活动目的	确保所有功能需求及非功能需求的质量和达到的效果满足客户要求，项目按期正常结项，产品基本可以投入运营。
责任角色	PO
参与角色	PM、PO、运维工程师、实施工程师、BD
输入	1. 产品安装部署及运维手册 2. 系统测试报告 3. 验收版本
输出交付件	准生产环境上线申请表、准生产环境监控报告、项目总结报告、准生产环境上线 checklist 表、培训材料、内部验收报告、产品市场交底书
活动描述	1. 测试工作结束且测试报告评审通过后，PM 填写《准生产申请表》，并附带《产品安装部署及运维手册》、《系统测试报告》，以邮件方式向运维工程师提交转准生产申请，并知会 PO。

	2. 运维工程师收到 PM 试运行申请后，运维团队根据《准生产环境上线 checklist 表》相关信息进行准生产准入检查，如果准入检查通过，则由 CM 创建验收版本，并输出《准生产环境上线 checklist 表》的确认结果，如果准入检查不通过，则返回项目团队修改。 3. 运维工程师根据《产品安装部署及运维手册》及准生产申请表等相关信息，部署产品验收版本。 4. 产品正式纳入准生产环境的在线监控后，运维团队根据在线监控记录监控期间发现的问题，统一提交 DevSutie 工具进行跟踪管理。并输出《准生产监控报告》。 5. 产品经理编制产品培训相关材料，组织实施工程师、商务 BD、市场营销等相关干系人开展产品培训，并收集整理培训后的问题与建议。 6. 产品经理依据验收标准组织事业部人员进行内部验收，执行验收期间发现的问题，统一提交 DevSutie 工具进行跟踪管理，如果验收通过，产品经理编制《内部验收报告》并对验收结果签字确认；如果验收不通过，则返回项目团队修改，必要时发起项目结项变更。 7. PM 编制产品市场交底书，并经过市场、实施、商务人员签字确认。 8. PM 组织项目团队进行项目总结，总结主要包含以下几点： 技术层面总结 管理层面总结 9. PM 检查并确保项目文档正确归档 SVN 库，根据项目结项审批流程指引，提交项目结项申请， 如果审批通过，PM 释放资源，并邮件知会相关领导；如果审批不通过，返回项目组修改。 10. CM 收到项目结项审批通过后，收回项目组成员配置库权限，并建立产品基线。
活动要求	1. 产品经理前期与项目组明确验收标准，并达成一致； 2. 产品经理确保验收的充分性。
补充说明	
方法与工具、IT	缺陷管理：DevSuite；文档归档：SVN
备注	。

2.3.2. 客户验收



活动名称	**客户验收**
活动目的	通过演示所有迭代完成的产品发布版本，确认是否符合客户要求，获得客户的认可和反馈
责任角色	PO
参与角色	实施工程师、PM、CUSTOMER
输入	产品发布版本
输出交付件	客户验收报告
活动描述	1. PO 负责组织产品验收，指定项目实施工程师。 2. 实施工程师提前和客户确认部署计划，提前部署好客户验收现场环境。 3. 实施工程师向产品管理人员申请产品出库，然后将产品发布版本部署到客户环境。 4. 如果客户需要产品培训，产品经理/实施工程师可组织产品培训工作。 5. 客户进行正式的产品验收，输出《客户验收报告》，如果有必须要修改的缺陷，PO 负责把缺陷转交到 PM，由 PM 安排人员进行修改、验证，并最终获得客户的确认。
活动要求	实施工程师需要和客户沟通，获得客户负责人的客户验收签字确认或邮件确认
补充说明	
方法与工具、IT	
备注	

2.4. 其他管理类活动

2.4.1. 站立会议

活动名称	站立会议
目　　的	增加团队凝聚力，产生积极的工作氛围；及时暴露风险和问题；促进团队内成员的沟通和协调。
活动输入	持续集成报告、各项状态信息等
输出交付件	NA
责任角色	PM
参与角色	PO、PM、SWE、TE、QA
活动描述	1. PM 组织会议，全员围绕任务墙/story 墙站立，开始会议。 2. 各成员依次发言，但是每次的发言顺序随机。 3. 发言中，各自更新 story 墙上 story 卡片的最新状态。 4. 对各问题明确协助人或跟踪解决人，仅对公共问题讨论解决方案。
活动要点	
方法与工具、IT	

2.4.2. 可视化管理

活动名称	可视化管理
目　　的	通过可视化管理可以清晰、简单和有效的组织和展示工作状况，及时暴露问题，减少沟通成本，激励团队。
活动输入	《用户故事列表》/DevSuite 工具信息、story 列表、站立会议、持续集成等信息
输出交付件	各状态刷新
责任角色	PM
参与角色	PO、PM、SWE、TE、QA
活动描述	1. 团队要在办公场所很容易看到故事墙、任务及问题进展状态。 2. 团队成员及时更新故事墙、任务及问题进展状态。
活动要求	1. 团队要在办公场所很容易看到故事墙、任务及问题进展状态。 2. 要简单，一目了然，便于更新。 3. 更新及时，状态客观。
补充说明	团队所关注的信息有多种多样，可视化管理也没有严格的定义和标准，根据实践，基本可以总结为以下几种方式，项目组可根据情况选择采用： 1. 故事墙/任务墙：关注软件开发组每个开发任务的进展和状态，实时展现团队的交付进展。 2. Burn Down Chart 燃烧图：展现剩余工作量和当前进展 3. 迟到展示区：以每周的粒度，展现每周站立会议的及时出勤情况。 4. 阻碍问题跟踪区：对阻碍项目的问题明确协助人或跟踪解决人，每日站立会议进行跟踪
方法与工具、IT	工具:白板　　项目管理工具:DevSuite