

# 项目启动

刘钦

# Outline

- 项目和项目管理
- 团队组织与管理
- 软件质量保障
- 软件配置管理
- 管理实践
- 项目实践

# 项目

- 项目是具有下列特征的一系列活动和任务[Kerzner2009]:
  - 具有一个明确的目标;
  - 有限定的开始和结束日期;
  - 有成本限制;
  - 消耗人力和非人力资源;
  - 多工种合作。

# 项目管理的目标

- 项目管理的目标是做到以下方面：
  - 在限定期限内；
  - 在一定的成本内；
  - 在要求的质量水平上；
  - 高效使用资源；
  - 获得客户认可。

# 过程组与活动

- 过程组：
  - 项目启动、项目计划、项目执行、项目跟踪与控制和项目收尾
- 活动
  - 计划制定、团队管理、成本控制、质量保障、度量、过程管理、进度跟踪与控制、风险管理、配置管理

# Outline

- 项目和项目管理
- 团队组织与管理
- 软件质量保障
- 软件配置管理
- 项目实践

# 团 队

- 一个协作良好的团队是任何项目成功的基础。
- 软件项目尤其依赖于有效的团队组织和管理：软件开发是一个以人为本的活动，人力资源是软件项目最大的资产。
- 有很多实践者认为比生产高质量产品更大的成功是在生产过程中建立一个凝聚的团队

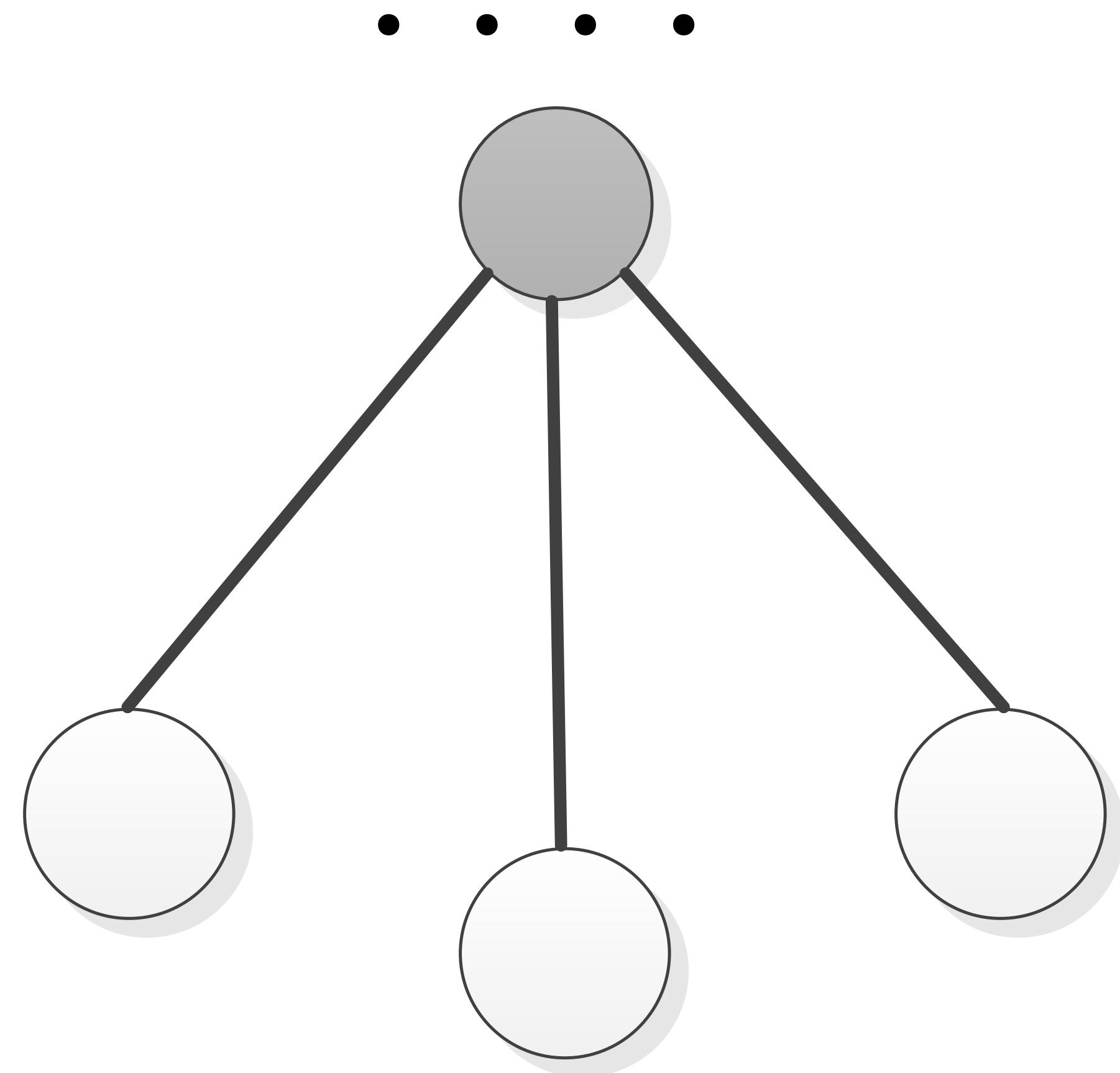
# 团队的特征

- [Katzenbach1993]将团队定义为：为了致的目的、绩效标准、方法而共担责任并且技能互补的少数人。
  - 团队成员要具备共同的目标。
  - 团队成员要共担责任。
  - 团队成员要技能互补。
  - 团队内部要有一个明确的结构。

# 团队结构

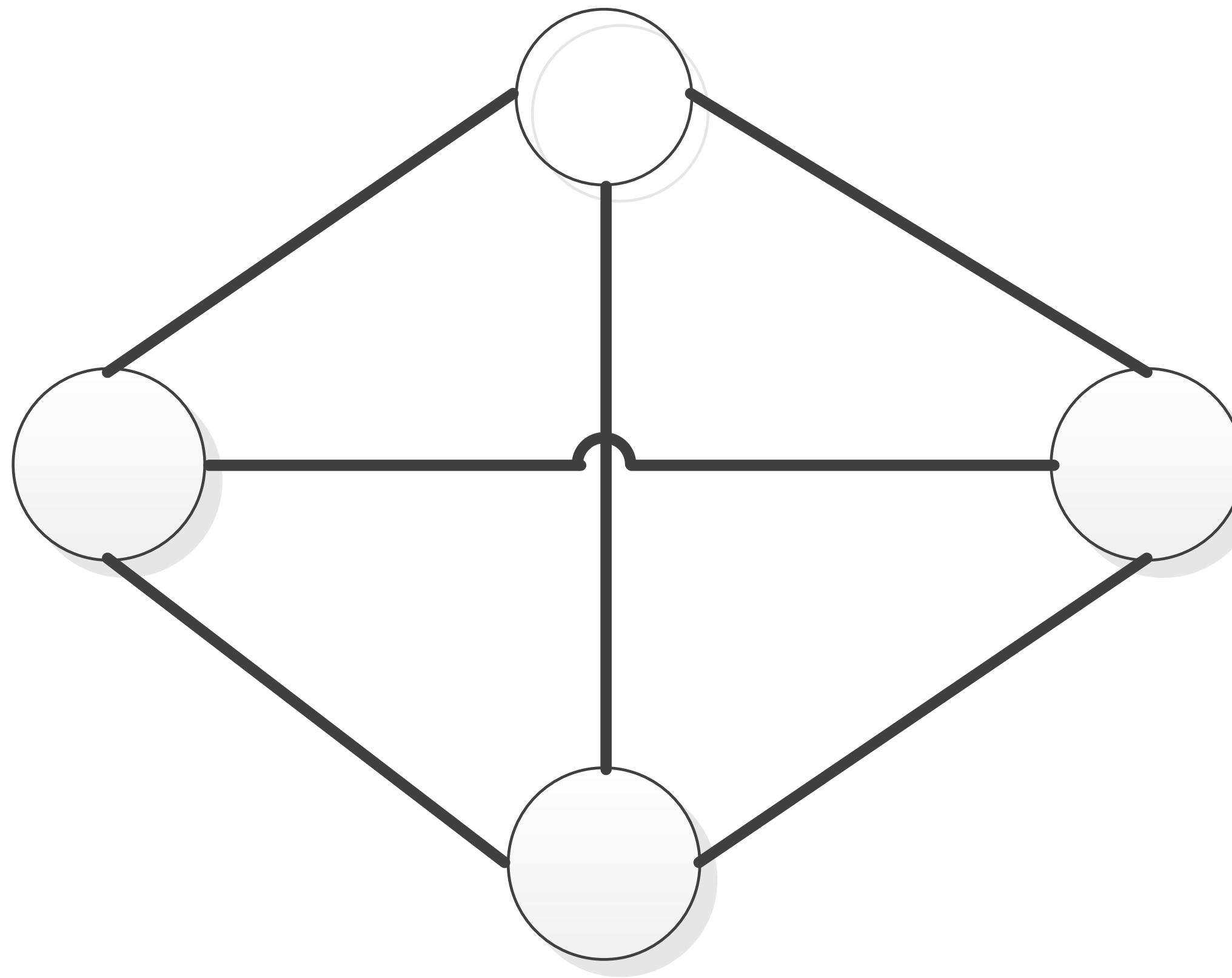
- 主程序员团队
- 民主团队
- 开放团队

效率比较高



# 主程序员团队

花更多时间在讨论上



**民主团队**



开放团队

# 团队建设

- 建立团队章程
- 持续成功
- 和谐沟通
- 避免团队杀手

# 某小组的团队章程示例

- 开会规则：
  - 1、开会时间由组员协商决定（电话协商），开会地点为509，一旦决定不允许请假、缺席。否则要求请其他小组成员喝冷饮，并组内批评。
  - 2、会前各小组成员要对所要讨论的议题有所准备，各小组成员均需发表个人看法，会议结果由表决决定，组长两票，组员各一票。会议期间需要有完整的记录。
  - 3、开会期间务须认真，不允许干任何于会议内容无关的事情。否则会内批评。
- 组内联系方式：
  - 开会协商：电话、当面协商； 会议缺席、迟到：电话联系； 文档交流：邮箱、svn； 娱乐交流：qq； 项目产物交流：svn；
- 工作时间安排：
  - 上午： 9: 00—11:30 ； 下午： 14:00—17:00 ； 晚上： 19:00—21:00.
  - 周六、周日的上午跟晚上休息。 工作日的晚上如果项目进度较快可以选择性休息，但一周不允许超过两次。
  - 如有无故旷课或早退现象需要写检讨并接受小组批评；如果早上迟到需要为大家带冷饮，迟到不允许超过40分钟，否则视为旷课。

顺序	开发人员	项目管理人员	普通人
1	成就感	责任感	成就感
2	发展机遇	成就感	受认可程度
3	工作乐趣	工作乐趣	工作乐趣
4	个人生活	受认可程度	责任感
5	成为技术主管的机会	发展机遇	领先
6	领先	与下属关系	工资
7	同事间人际关系	同事间人际关系	发展机遇
8	受认可程度	领先	与下属关系
9	工资	工资	地位
10	责任感	操控能力	操控能力
11	操控能力	公司政策和经营	同事间人际关系
12	工作保障	工作保障	成为技术主管的机会
13	与下属关系	成为技术主管的机会	公司政策和经营
14	公司政策和经营	地位	工作条件
15	工作条件	个人生活	个人生活
16	地位	工作条件	工作保障

**不同人员激励因素比较，源自[Boehm1981]**

# 避免团队杀手

- [DeMarco1999]认为组织和管理团队时要回避下列团队杀手：

- 防范式管理。
- 官僚主义。
- 地理分散。
- 时间分割。
- 产品质量的降低。
- 虚假的最后期限。
- 小圈子控制。

# Outline

- 项目和项目管理
- 团队组织与管理
- 软件质量保障
- 软件配置管理
- 项目实践

# 软件质量

- 软件工程师也要对软件产品的质量负责。
- 对软件质量的要求可能是显式的，也可能是隐式的。
- 人们通常会选用系统的某些质量要素进行量化处理，建立质量特征，这些特征被称为质量属性（Quality Attribute）。
- 为了根据质量属性描述和评价系统的整体质量，人们从很多质量属性的定义当中选择了一些能够相互配合、相互联系的特征集，它们被称为质量模型。

# 质量模型

- [IEEE1061-1992,1998] 和 [ISO/IEC 9126-1]

- 因素

- 功能性

- 可靠性

- 易用性

- 效率

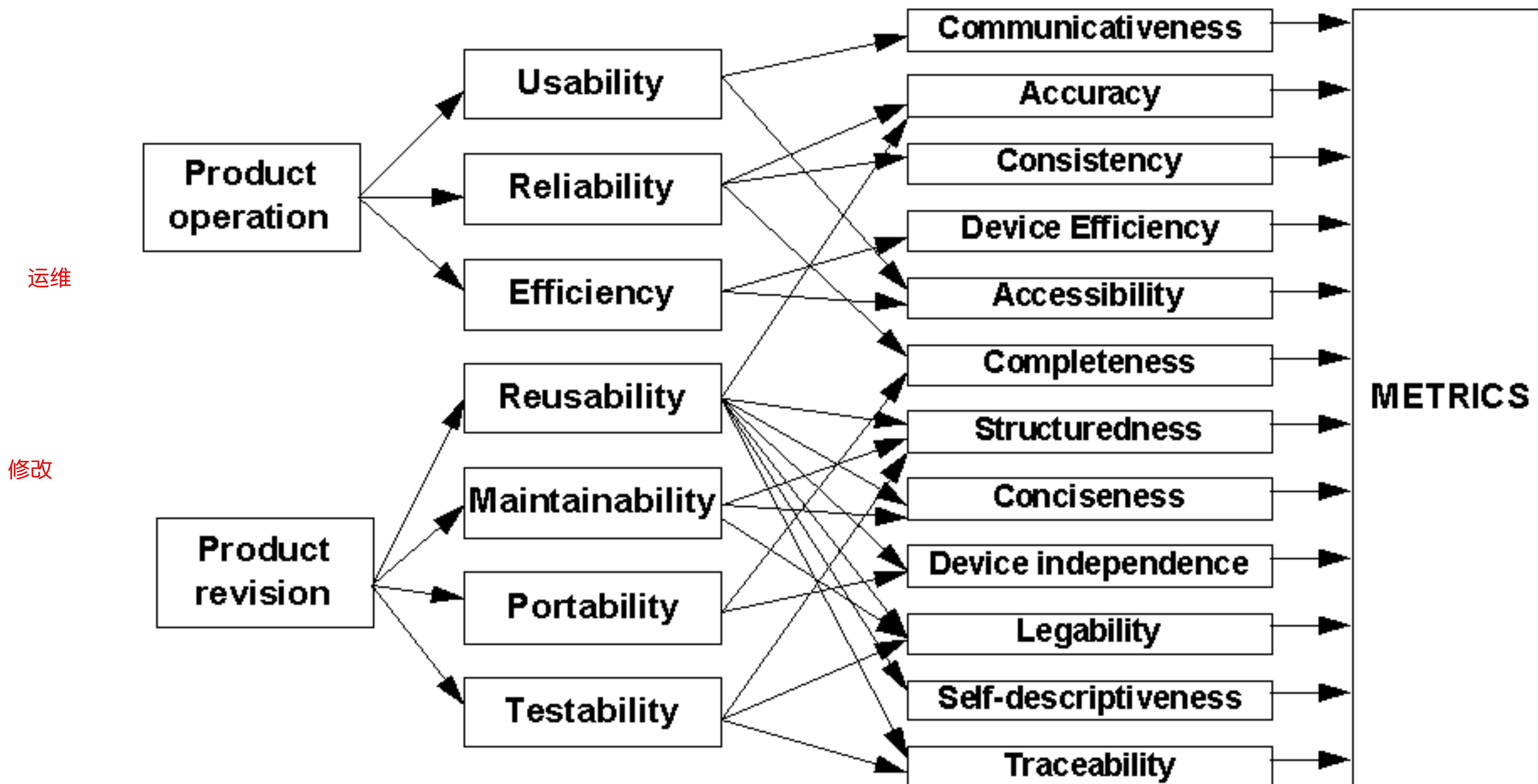
- 可维护性

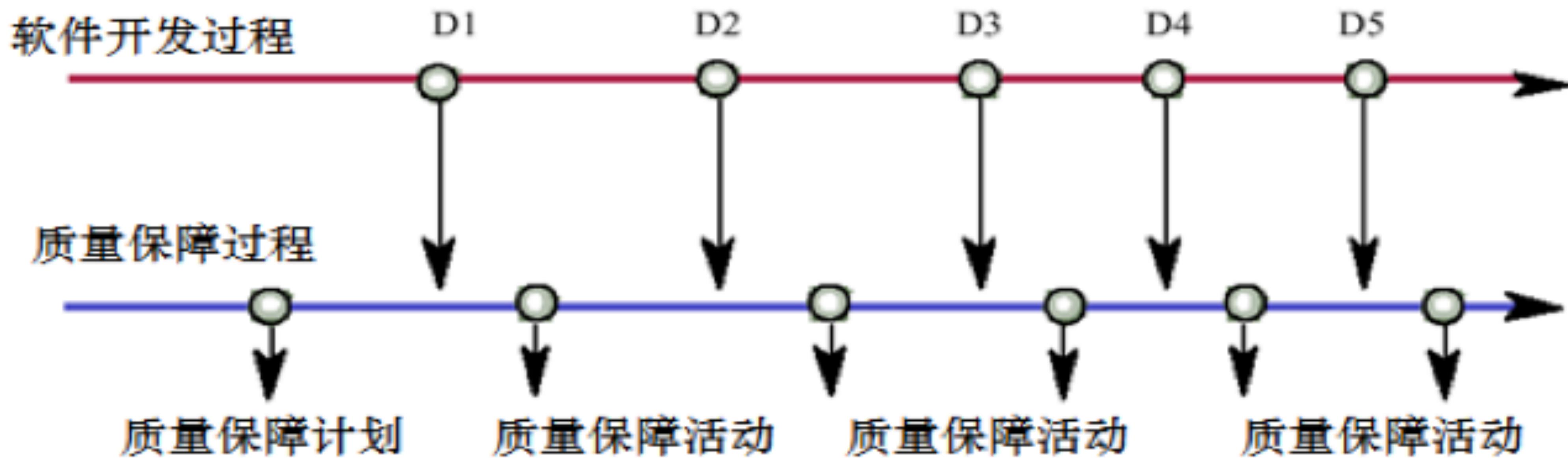
- 可移植性

# Use

# Factor

# Criteria





# 质量保障

## 里程碑

## 质量保障活动

需求开发

需求评审、需求度量

体系结构

体系结构评审、集成测试（持续集成）

详细设计

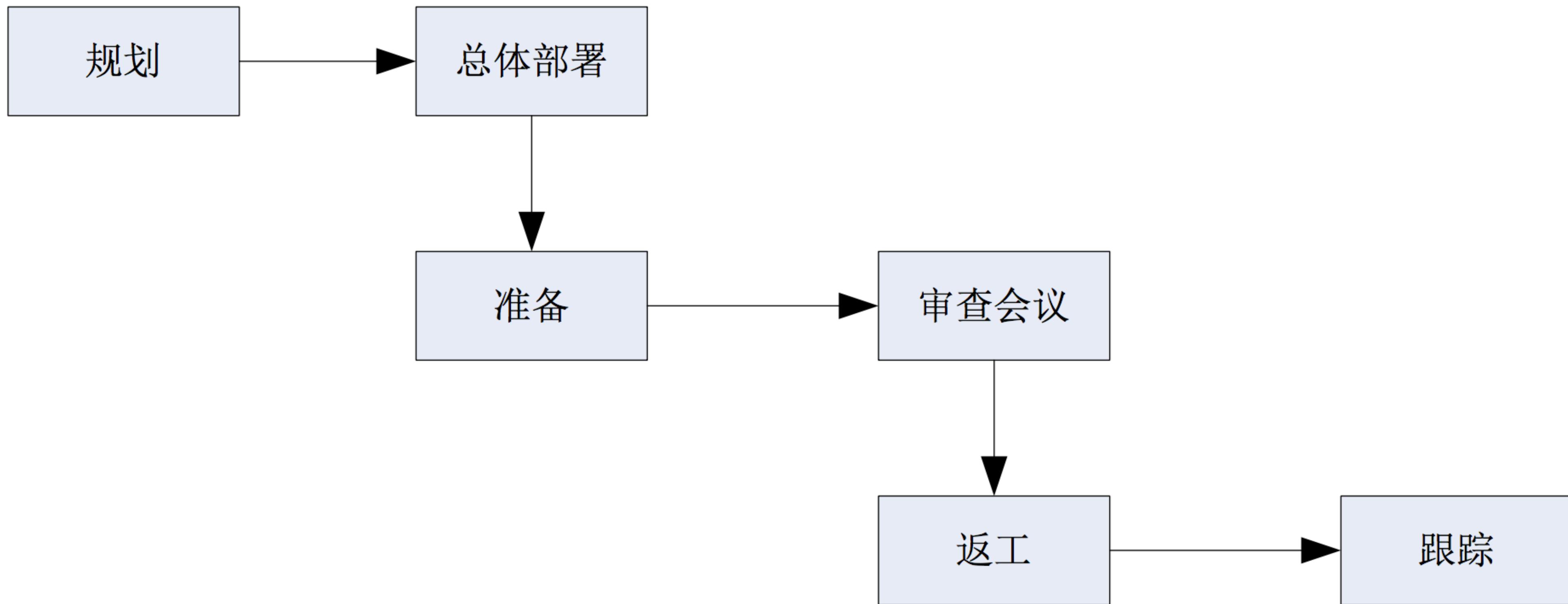
详细设计评审、设计度量、集成测试（持续集成）

实现

代码评审、代码度量、测试（测试驱动、持续集成）

测试

测试、测试度量



# 评审过程

# 评审

- 1. 在规划阶段（Planning），制定审查计划，决定审查会议的次数，安排每次审查会议的时间、地点、参与人员、审查内容等等。
- 2. 在总体部署阶段（Overview），向所有参与审查会议的人员描述待审查材料的内容、审查的目标以及一些假设，并分发文档。
- 3. 在准备阶段（Preparation），审查人员各自独立执行检查任务。在检查的过程当中，他们可能会被要求使用检查清单、场景等检查方法。检查中发现的问题会被记录下来，以准备开讨论或者提交给收集人员。
- 4. 在审查会议阶段（Inspection Meeting），通过会议讨论，识别、确认、分类发现的错误。
- 5. 在返工阶段（Rework），修改发现的缺陷。
- 6. 在跟踪阶段（Follow-up），要确认所有发现的问题都得到了解决，所有的错误都得到了修正。

# 质量度量

- 度量产生自统计控制（Statistical Control）思想。“你不能控制自己无法度量的东西”[DeMarco1998]。
- 测度（Measure）就是为了描述软件产品而提供的定量指标。
  - 代码行数
- 进行测度的活动被称为测量（Measurement）。
- 度量（Metric）是软件产品在特定属性上的量化测度程度。
  - 例如基于所有对象的代码行数测度可以建立平均代码行数、最大代码行数、最小代码行数

# Outline

- 项目和项目管理
- 团队组织与管理
- 软件质量保障
- 软件配置管理
- 项目实践

# 软件配置管理的动机

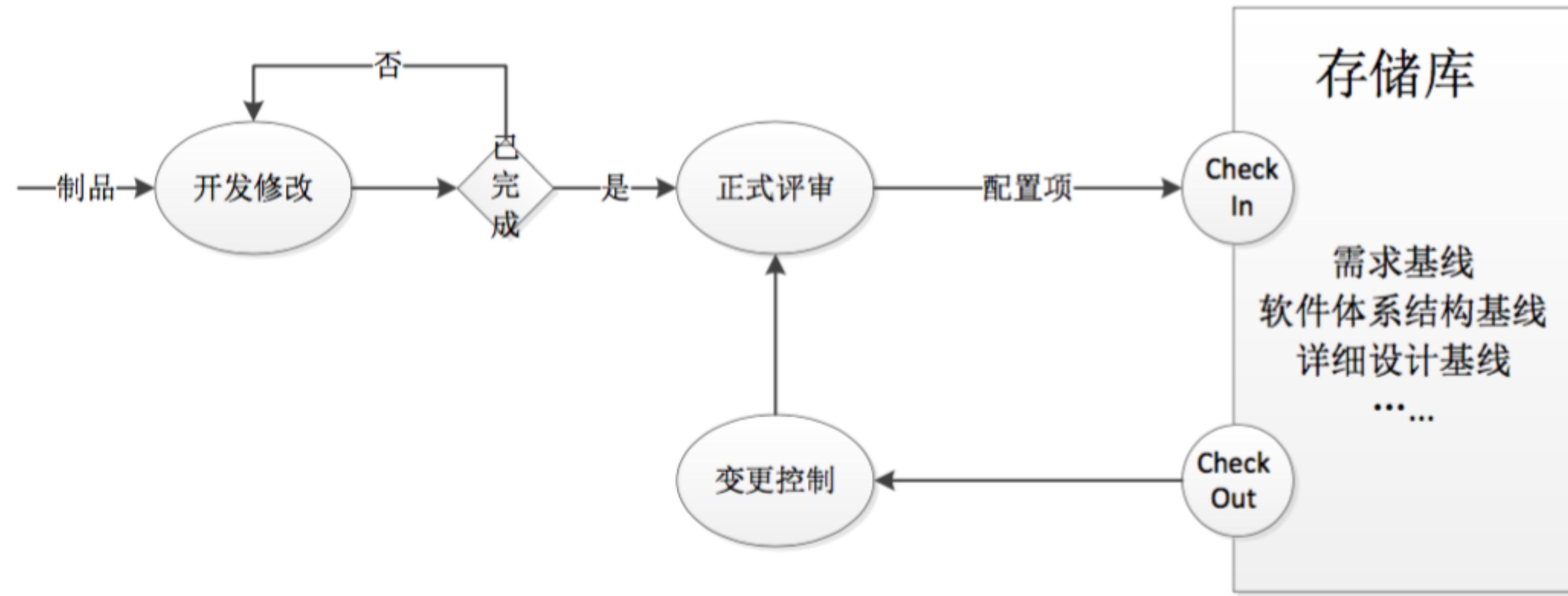
- 在软件开发活动中，除了最终产品之外，还会产生很多中间制品，例如需求规格说明、需求分析模型、软件体系结构设计模型、详细设计模型等。这些制品是不同阶段、不同角色、不同软件开发活动进行协同的基础。
- 在复杂软件系统开发中，产生的制品数量众多，以至于开发者需要维护一个清单才能清楚项目所处的状态，理解已经完成的工作和将要进行的工作。
- 某个制品发生变化带来的最大挑战是如何确保其使用者能够得到最新的制品，避免开发协同出现问题。

# 配置管理

- IEEE将配置管理定义为[IEEE610.12-1990]：“用技术的和管理的指导和监督方法，来标识和说明配置项的功能和物理特征，控制对这些特征的变更，记录和报告变更处理及其实现状态，并验证与规格需求的一致性”。

# 配置项

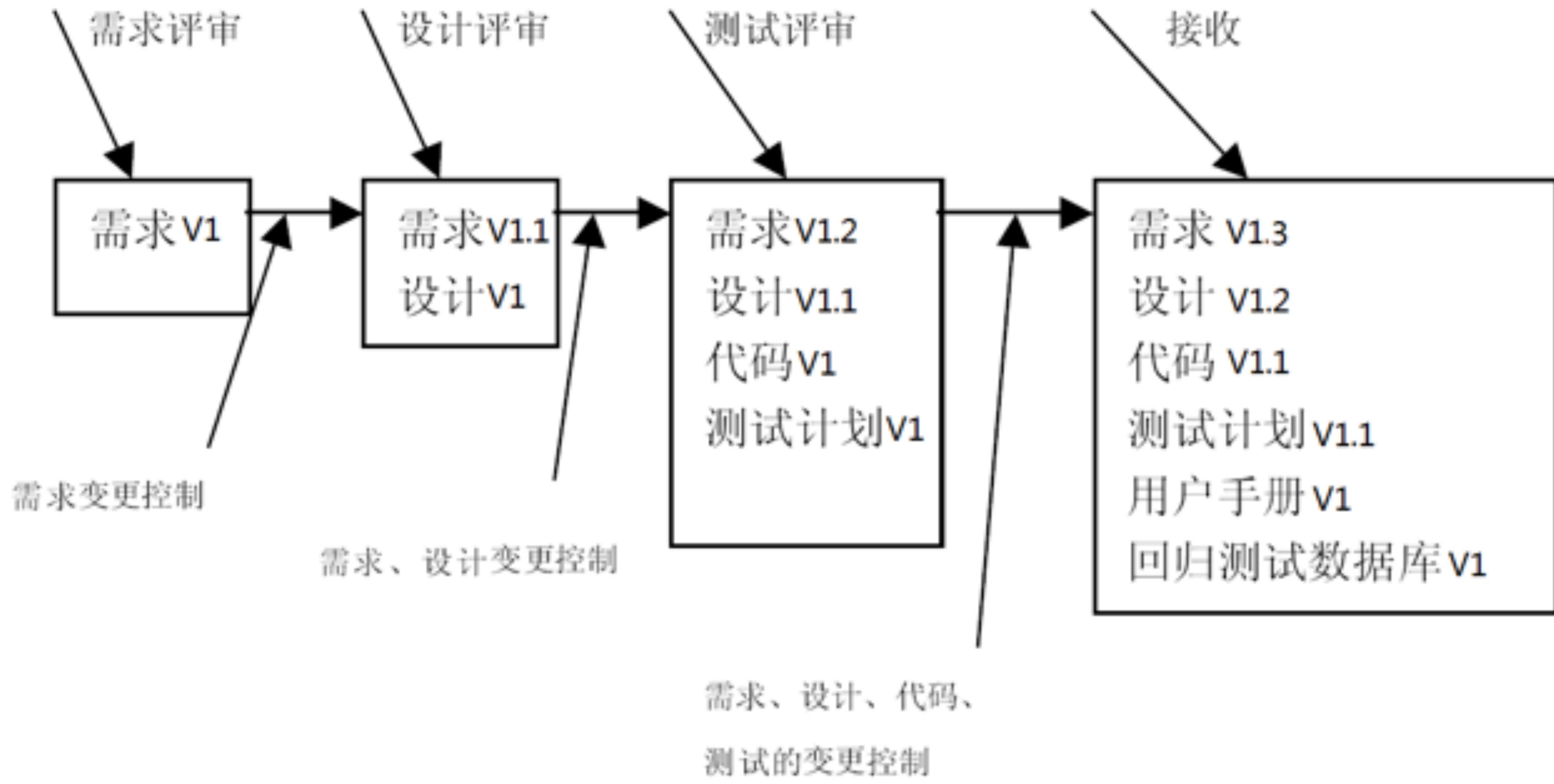
- IEEE将配置项定义为[IEEE610.12-1990]：“置于软件配置管理之下的软件配置的各种有关项目，包括各类管理文档、评审记录与文档、软件文档、源码及其可执行码、运行所需的系统软件和支持软件以及有关数据等”。



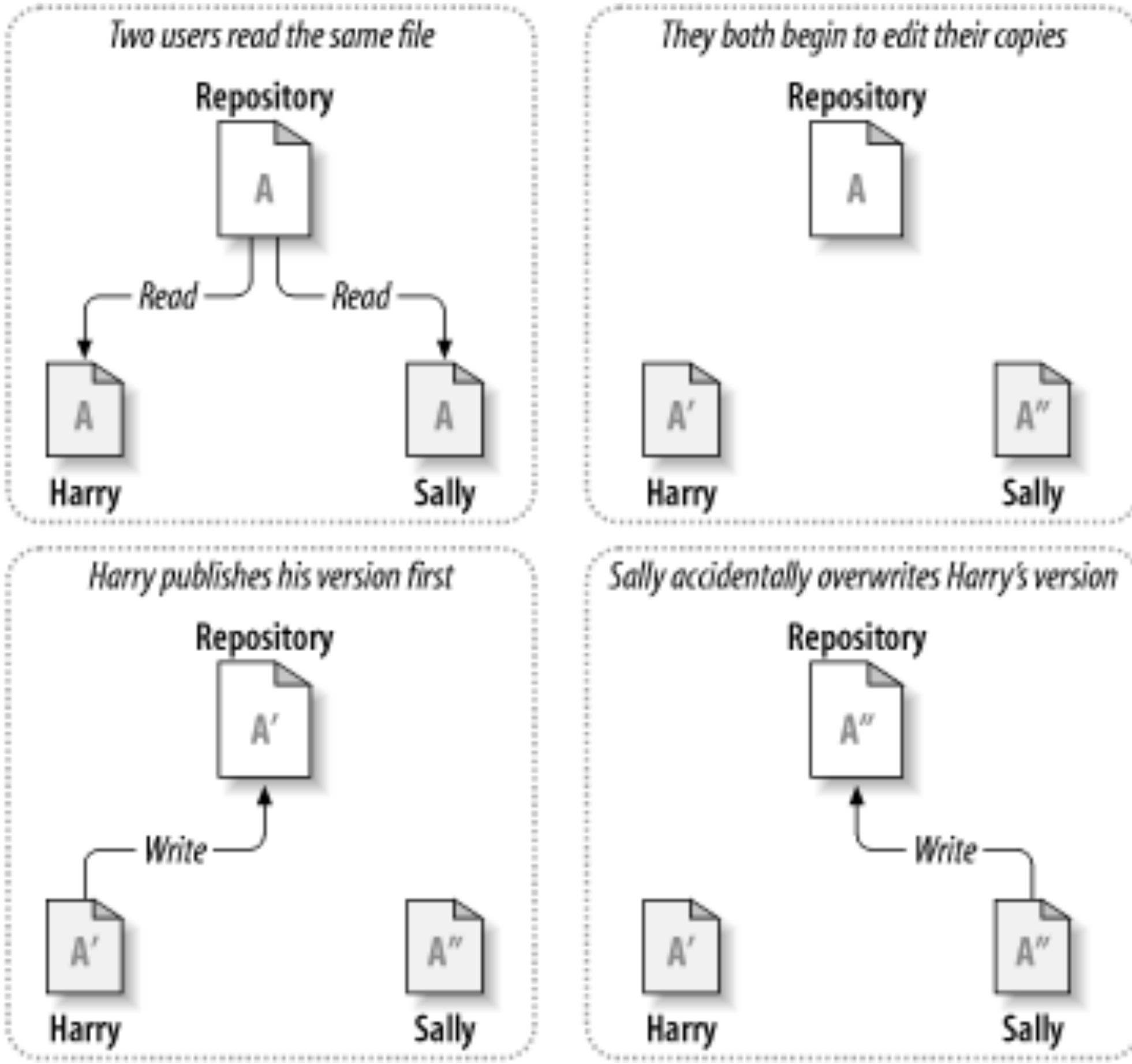
[IEEE610.12-1990]将**基线**定义为：已经经过正式评审的规格说明或产品，可以作为进一步开发的基础，并且只有通过正式的变更控制过程才能变更。

# 配置管理活动

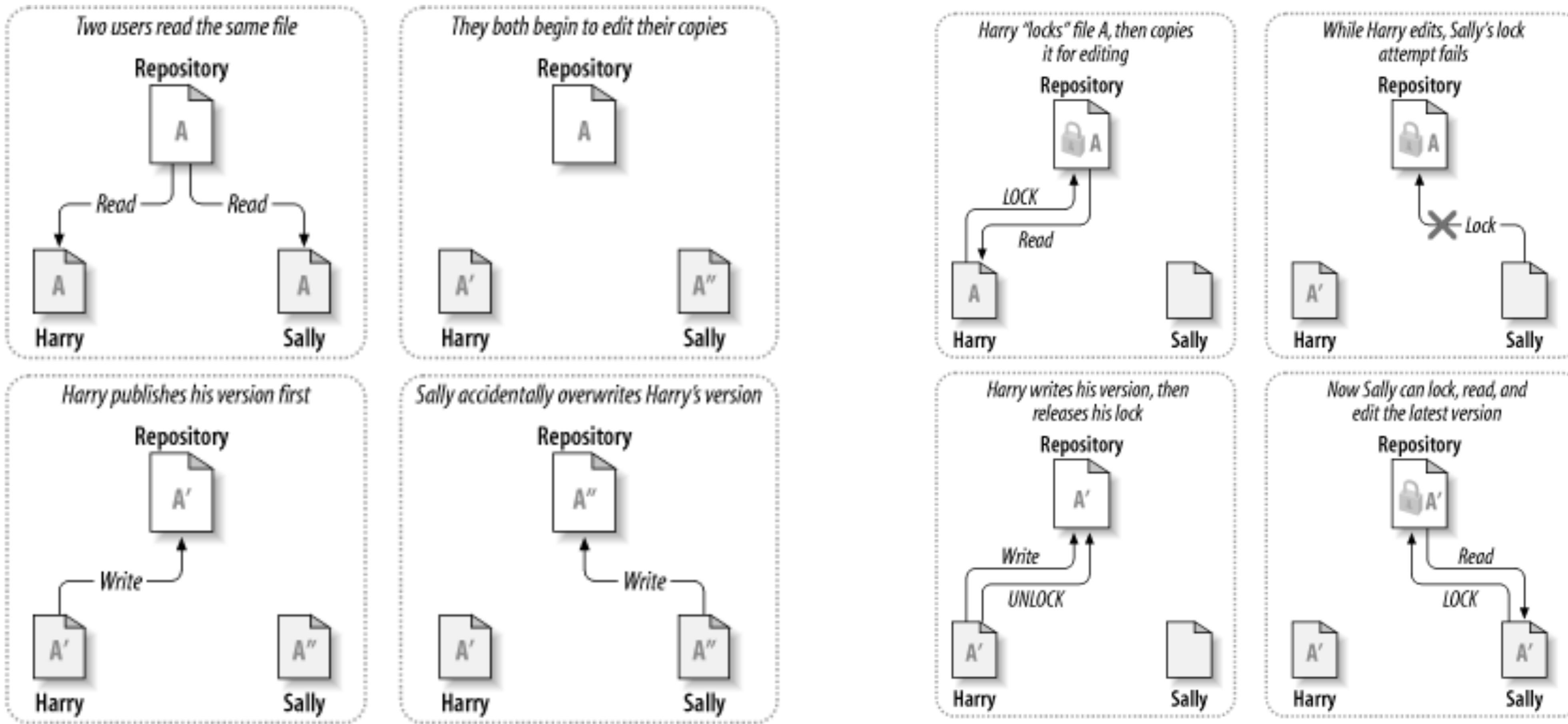
- 标识配置项
- 版本管理
- 变更控制
- 配置审计
- 状态报告
- 软件发布管理



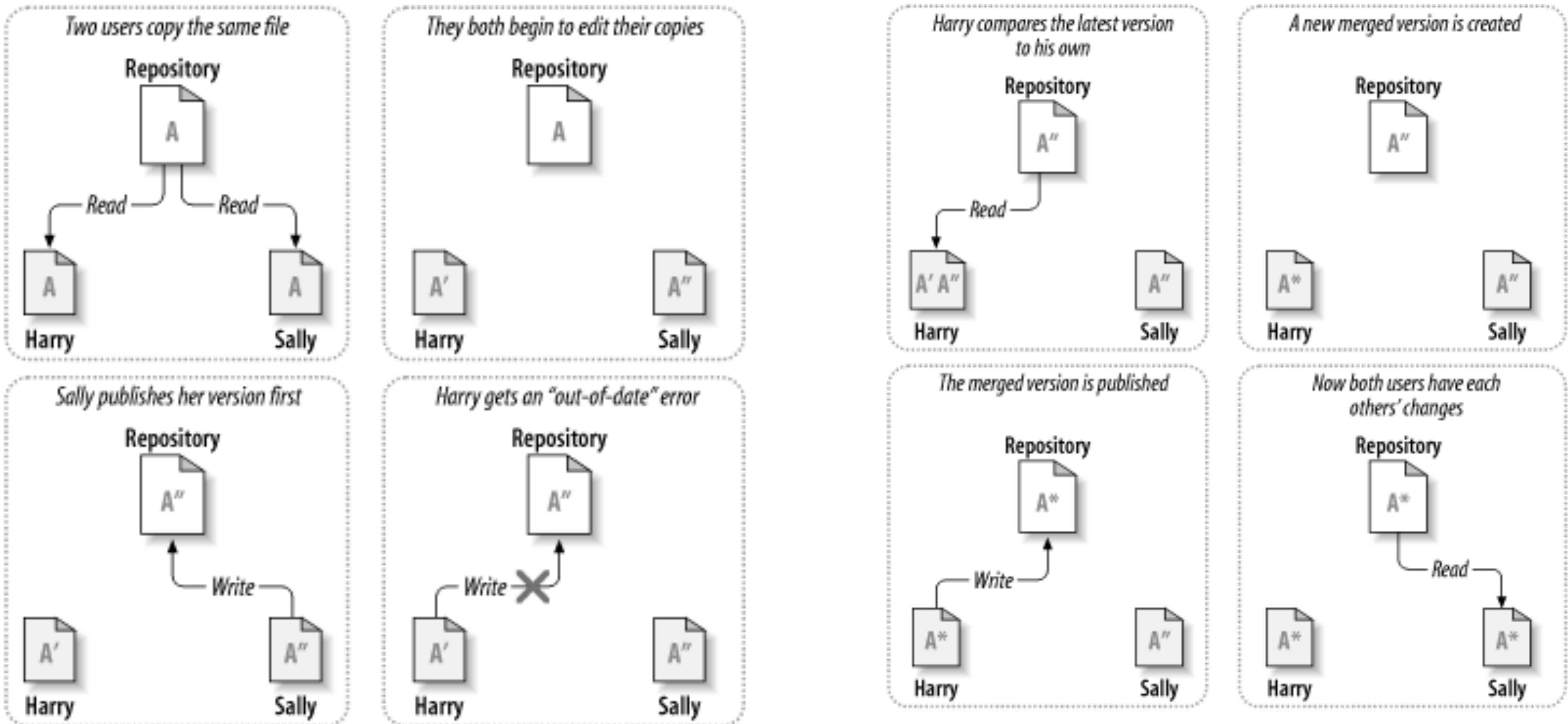
# 版本管理示意图



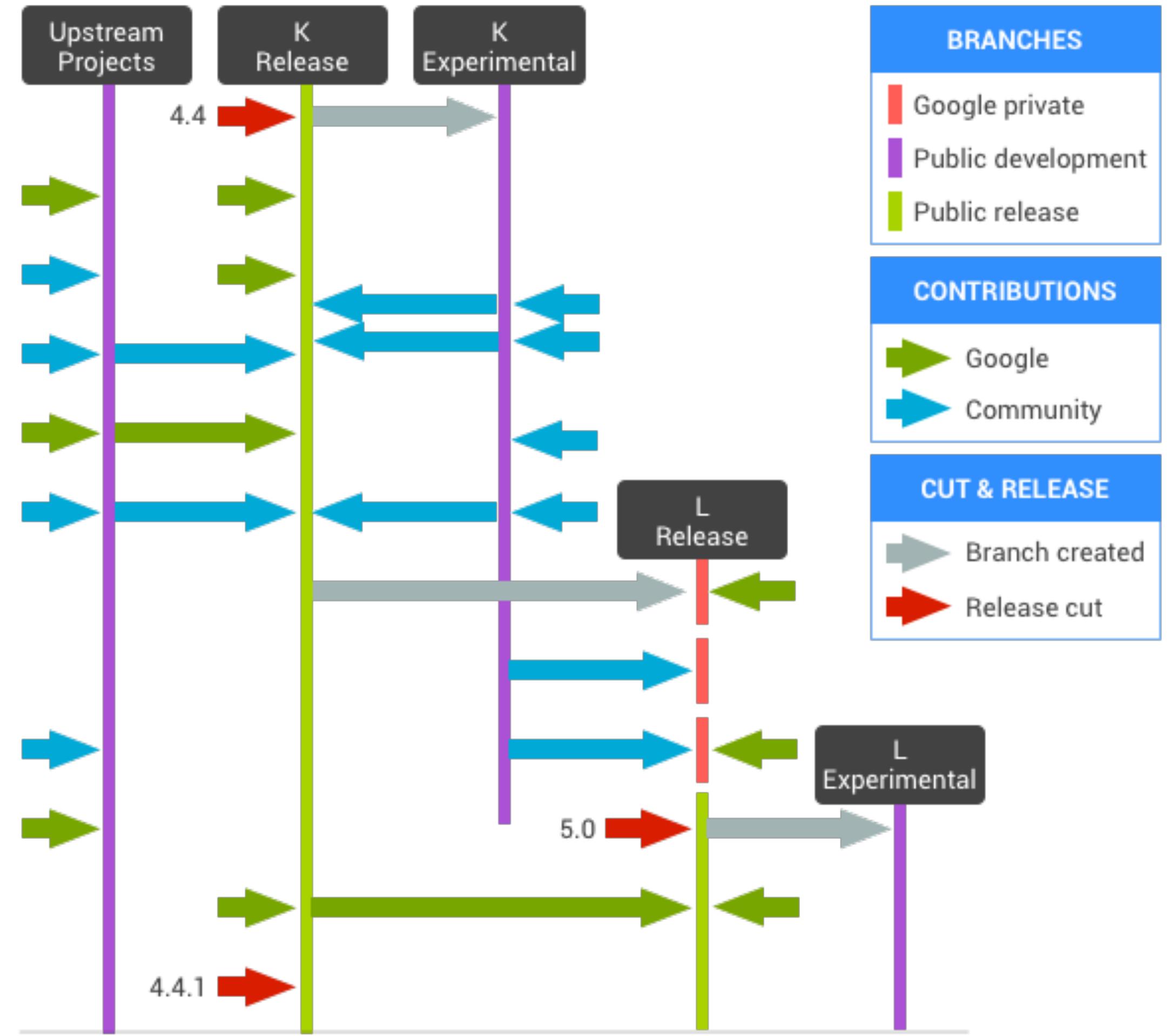
# 版本控制的问题



# lock-modify-unlock solution



# copy-modify-merge solution



# Android Open Source Project Code and Releases

# 分支管理常见策略

- 主分支 (Master)
- 开发分支 (Develop)
- 临时性分支
  - 功能 (Feature)
  - 预发布 (Release)
  - 修补bug (Fixbug)

Master



Tag  
0.1



Tag  
0.2



Tag  
0.3

Develop



Master



Tag  
0.1

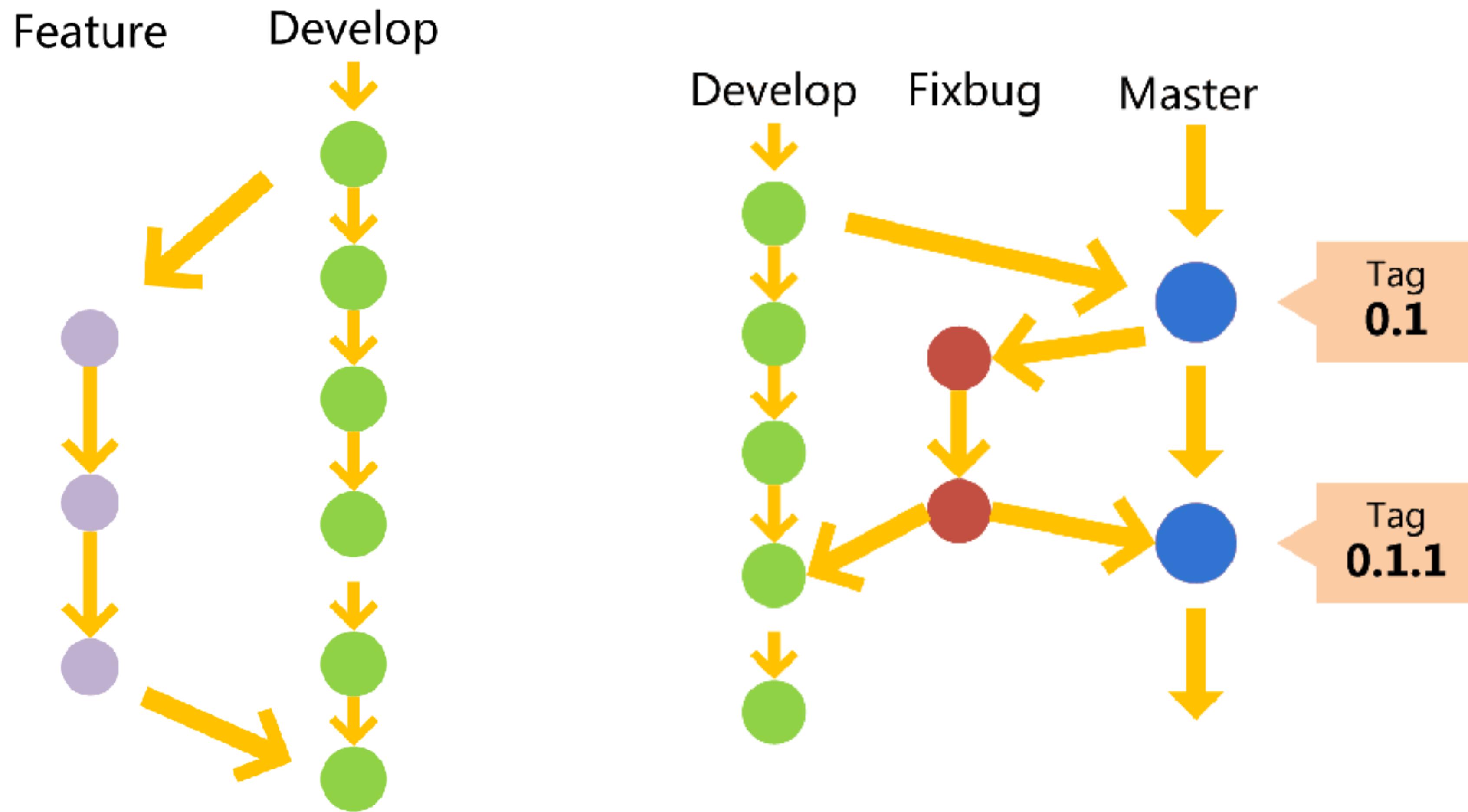


Tag  
0.2



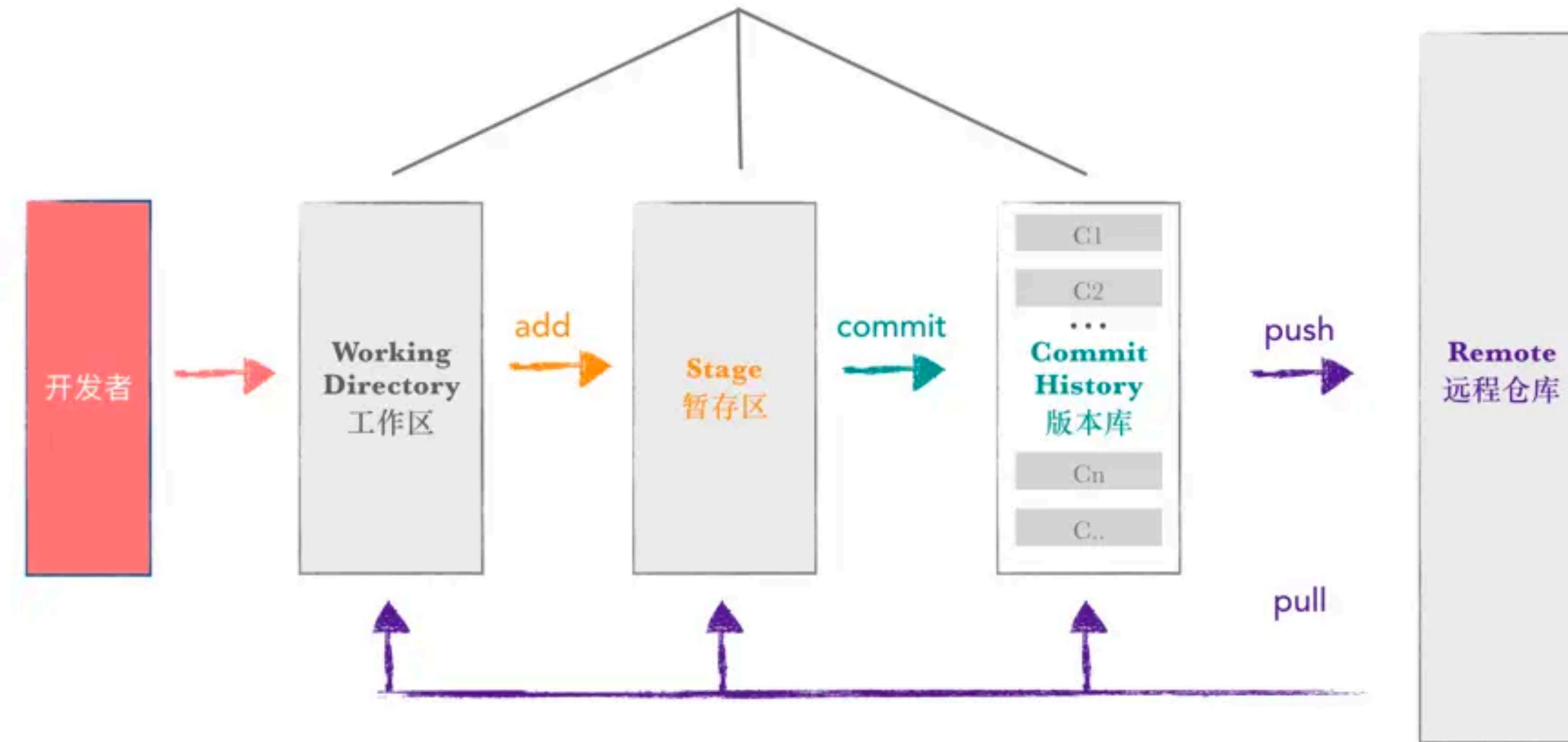
Tag  
0.3

# Master & Develop



# Feature & Fixbug

## Git 三大分区



# Git的分区

# 查看不同

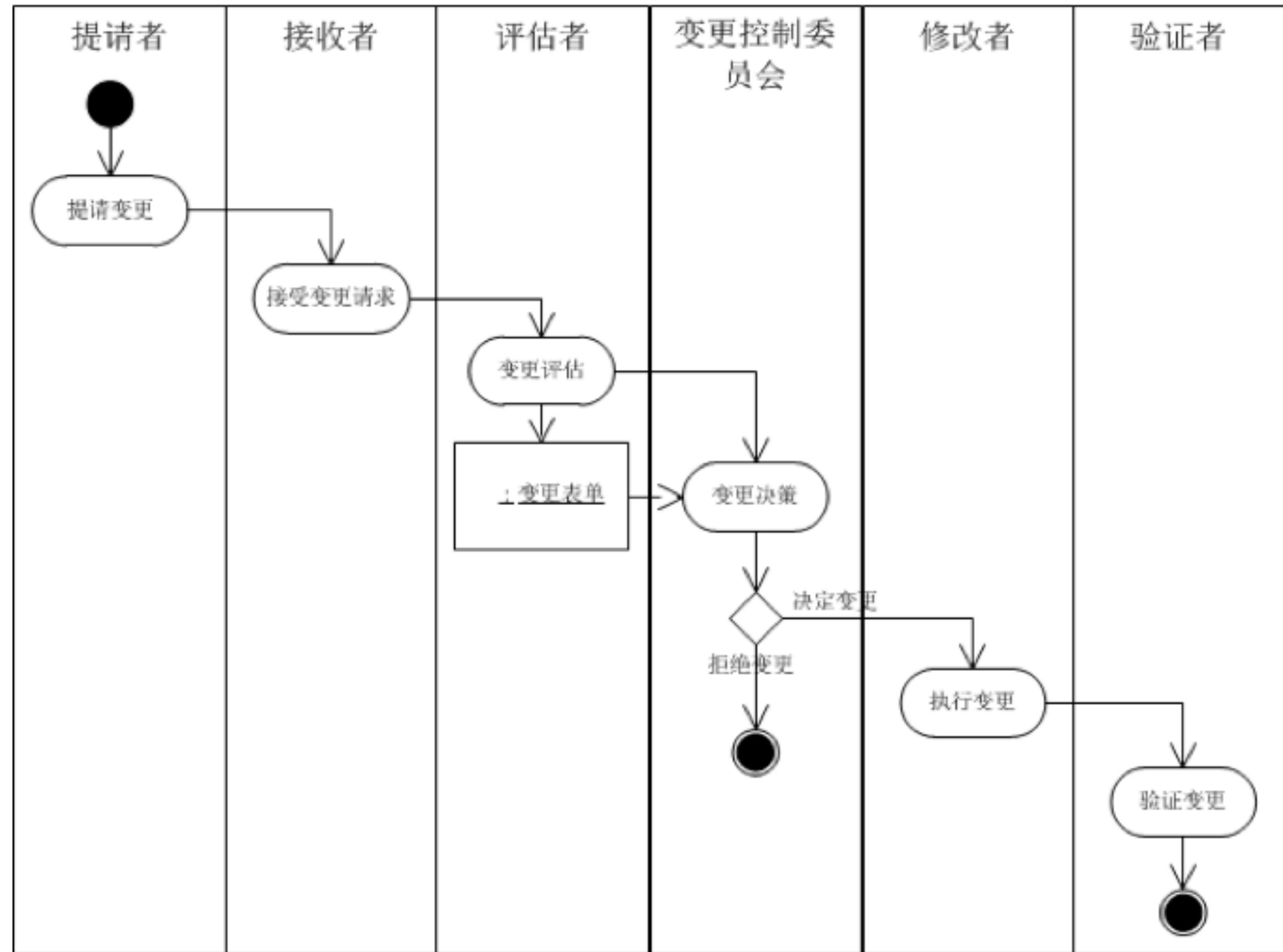
- git diff
  - 工作区 vs 暂存区
- git diff head
  - 工作区 vs 版本库
- git diff --cached
  - 暂存区 vs 版本库

# 为啥要暂存区

- 增加灵活性
  - 修改了4个文件，在不放弃任何修改的情况下，其中一个文件不想提交，如何操作？（没add : git add 已经add: git reset --soft）
  - 修改到一半的文件，突然间不需要或者放弃修改了，怎么恢复未修改前文件？（git checkout）
  - 代码写一半，被打断去做其他功能开发，未完成代码保存？（git stash）
  - 代码写一半，发现忘记切换分支了？（git stash & git checkout）
  - 代码需要回滚了？（git reset）

- 作者：段浅浅儿
- 链接：<https://www.jianshu.com/p/f79ae362ef9e>





# 变更控制

# 变更请求表单

项目名称:	请求编号:
提请人:	提请日期:
提请理由及优先级:	
变更请求描述:	
评估人:	评估日期:
评估优先级:	变更类型:
影响范围:	
工作量估算:	
变更评价:	
提交CCB日期:	CCB决策日期:
CCB决定:	
修改人:	修改日期:
修改结果:	
验证人:	验证日期:
验证结果:	
备注:	

# Outline

- 项目和项目管理
- 团队组织与管理
- 软件质量保障
- 软件配置管理
- 管理实践
- 项目实践

# 管理实践

- 经济为本
  - 分工协作
  - 目标驱动
  - 常来常往
  - 有张有驰
  - 不断总结
  -
- 《软件工程实践导论》

经济为本

# 软件项目的投入-1

- 人员的成本
  - 这是最重要的一部分投入。现在的软件从业人员的工资不断上涨，工资占整个软件成本的比例更是呈上升趋势。除了开发人员外，还要计算项目管理人员和其他相应支持人员的费用（不是全职的，按时间乘比例计算）。
- 工具的购买
  - 包括计算机及其周围配套设备等硬件，也包括开发工具、办公套件等软件。
- 培训的费用
  - 开发人员接受培训，获得开发项目所需技能的费用。
- 差旅费
  - 拜访客户，参加会议等的费用。

# 软件项目的投入-2

- 维护的费用
  - 定时的数据备份、系统监控、系统维修和升级等引起的费用。
- 生产停顿的损失
  - 因为项目调试引起正常工作业务停顿的损失。
- 市场和服务的费用
  - 推广软件产品所要的广告费用、参加展览会的费用等。
- 机会成本
  - 因为投资该项目，而不能投资别的项目或者放银行收取利息的机会成本。

# 软件项目的产出

- 节约商业活动成本。
  - 只要是和无新软件系统时候比较，将节省的时间和原材料折算成量化的数字。例如，开发了新的库存管理系统后，加快了流通并减少了库存浪费。
- 创新商机增加销售。
  - 指由于使用新软件带来的盈利，可能是软件产品本身的销售，也可能是软件项目带来的营业成长。
- 提高品牌含金量。
  - 提高质量和客户满意度，可以带来品牌含金量的提高。这比较虚一点，但也可以像企业的无形资产一样估算。

# 经济为本

- 为技术而技术是条死胡同
- 以经济原则指导软件项目的决策过程
- 按照产品规律来营销软件产品
- 以收益为依据规划设计产品

# 分工协作

# 横向比较

- 电影制作
- 急诊室
- 厨房
- 软件开发

# Harlan Mills建议的团队

- 首席程序员
- 副手
- 行政助理
- 编辑
- 秘书两名
- 程序管理员
- 工具专家
- 测试员
- 语言专家

# 三驾马车

- 产品经理
- 项目经理
- 技术经理

# 项目管理的角色

- 非正式的角色
- 明确定义的角色 (PM)
  - 领导整个团队以了解项目工作 (计划, 进度安排以及收集需求)
  - 带领项目进行设计和开发工作 (沟通、决策以及中期策略)
  - 以及驱动完成整个项目 (领导力、风险管理以及终局策略)

# 目标驱动

# 目标的SMART原则

- specific
- measurable
- achievable
- reasonable
- time

常来常往

# 观念

- 相互尊重
- 管道流畅
- 开放透明
- 坦诚真实

**有张有弛**

# 方式

- Kick-off Meeting
- 发布聚会
- 休假
- 技术培训
- 人员培训和被培训
- 换个项目

**不断总结**

**关键是“不断”**

# 给管理新手的建议

- 压力和分心
- 不要将流程与目标相混淆
- 适度参与
- 利用好自己的观点
- 创造独特的价值

# Paul Beckett给管理新手的建议

- 你所知道的比你想象的要多。
- 让人喜欢是好事，但不用刻意讨好。
- 人们并不介意自己被管理。
- 如果你想唤起别人对某件事的热情，自己先表现出热情。
- 需要坦白的时候就直言不讳。
- 你需要不停地进步。

# Outline

- 项目和项目管理
- 团队组织与管理
- 软件质量保障
- 软件配置管理
- 项目实践

# Outline

- 项目和项目管理
- 团队组织与管理
- 软件质量保障
- 软件配置管理
- 项目实践

# 项目实践

- 1. 为实践项目组建你的团队：
  - 选择技能互补的成员组成团队，明确分工；
  - 根据成员特点，选择团队结构；（建议使用民主团队）
  - 建立团队章程；
  - 明确团队的交流沟通手段。

- 2. 配置管理
  - 所有产物都通过Gitlab来管理
  - 建立Group
  - 文档采用MD文件

- 3. 项目结构

- 项目名称ERPnju

- src

- main

- java

- resources

- test

- java

- resources

- doc

- md

- img

# 项目开发管理工具

- 协同管理
  - trello/teambition/worktile/github project
- 协同画图
  - processon
- 文档管理
  - 语雀
- UML画图
  - astah
- 原型设计
  - 墨刀
- adobe xd
- 开发
  - git + jenkins + maven + mysql
- 测试
  - JUnit
  - selenium



# CI / CD

# Q&A

# 预习测验 -- 5分钟

- 1. 典型团队结构有哪些?
- 2. 什么是“配置项”?

# 预习测验 -- 5分钟

- 1. 项目管理的目标是什么？
- 2. 变更控制的过程是什么？