



西安电子科技大学
XIDIAN UNIVERSITY

软件项目管理





内容

1. 软件项目及其特点

- ✓ 软件项目及其管理的特殊性

2. 软件项目管理内容

- ✓ 软件项目管理的对象和要素

3. 软件项目管理的方法

- ✓ 软件项目管理的常见方法



1.1 何为项目 (Project)?

□项目概念

- ✓ 项目是指为**创建**一个**唯一**的**产品**或者提供唯一的**服务**而进行的**努力**
- ✓ 项目是基于既定**资源与约束**，为实现既定**目标**而实施的**活动**，它是一份临时工作，目的是创造独特产品、服务或者结果

□典型项目示例

- ✓ 阿波罗登月项目
- ✓ Windows 7开发项目
- ✓ 三峡水利项目
- ✓ 载人飞船项目
- ✓

中国探月工程
2020.12.17
“绕落回” 收官

项目也称工程，存在约束，实施活动，提供产品和服务

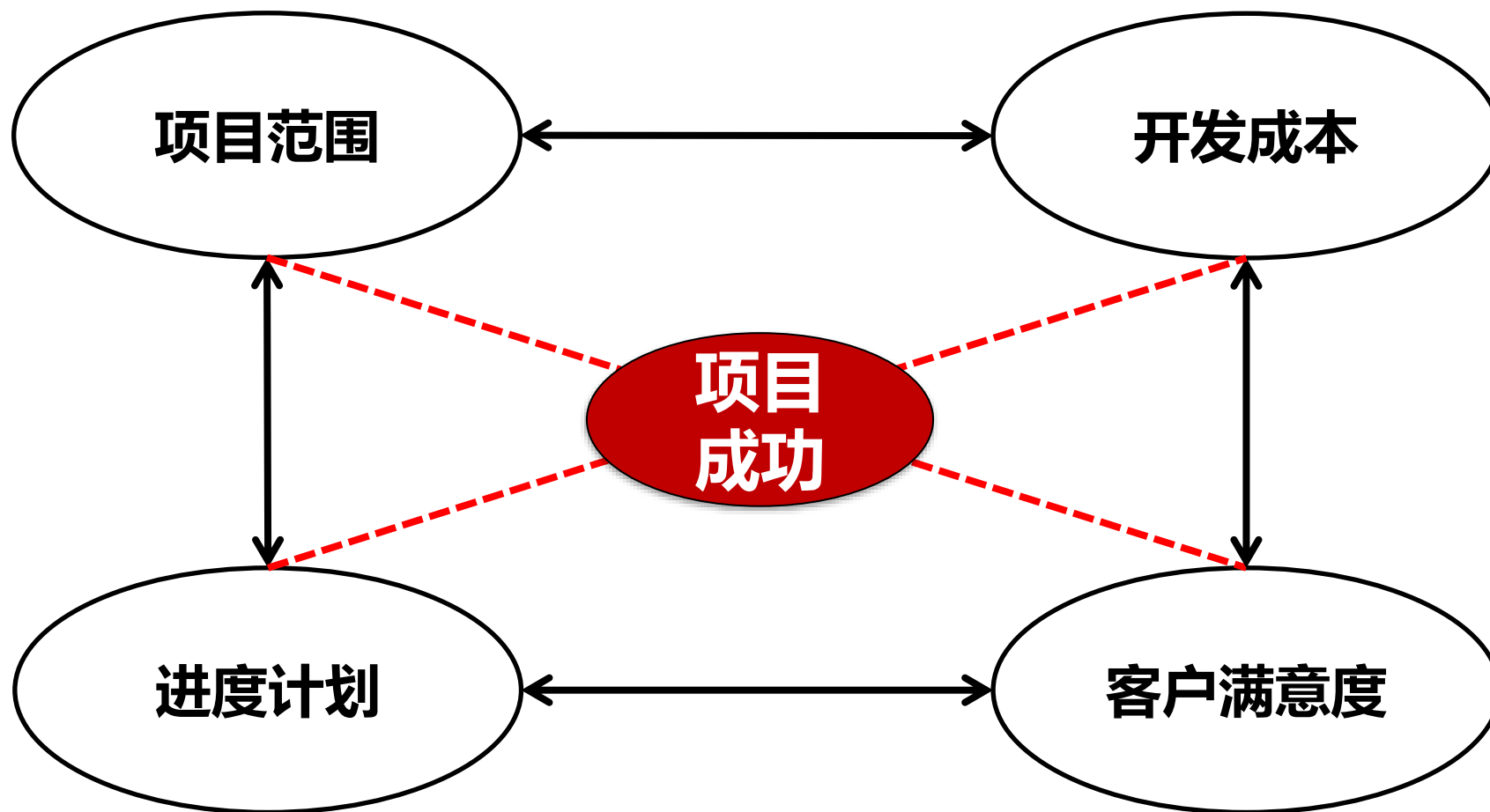
1.2 项目特点

- **目标性:** 获得预期的结果
- **进度性:** 在限定期间完成
- **约束性:** 具有有限的资源（如人员、经费、工具等）
- **多方性:** 涉及多个不同人与组织
- **独立性:** 项目间无重复性
- **不确定性:** 项目的实施及其结果不确定性

中国探月工程



影响项目成功的因素



项目的开展和实施受多要素影响，其结果具有不确定性

1.3 软件项目及其特点

□何为软件项目

✓针对**软件**这一特定**产品**和**服务**的项目努力开展 “**软件开发活动**”

□软件项目的特点

- ✓**对象**：作为逻辑产品的软件
- ✓**过程**：不以制造为主，没有重复生产过程
- ✓**属性**：实施要素难以度量和估算，如成本、进度、质量
- ✓**复杂性**：作为逻辑产品的复杂性非常高
- ✓**易变性**：软件需求通常难以确定且经常变化

示例：软件项目

- 火炮的火控软件
- 一体化指挥信息系统
- 卫星图像处理软件
- 导弹飞控软件
- 机载软件系统
- 微信软件
- 12306软件
-



对象、属性、过程、复杂性、易变性等

1. 软件项目及其特点

✓ 软件项目管理的特殊性

2. 软件项目管理内容

✓ 软件项目管理的对象和要素

3. 软件项目管理的方法和标准

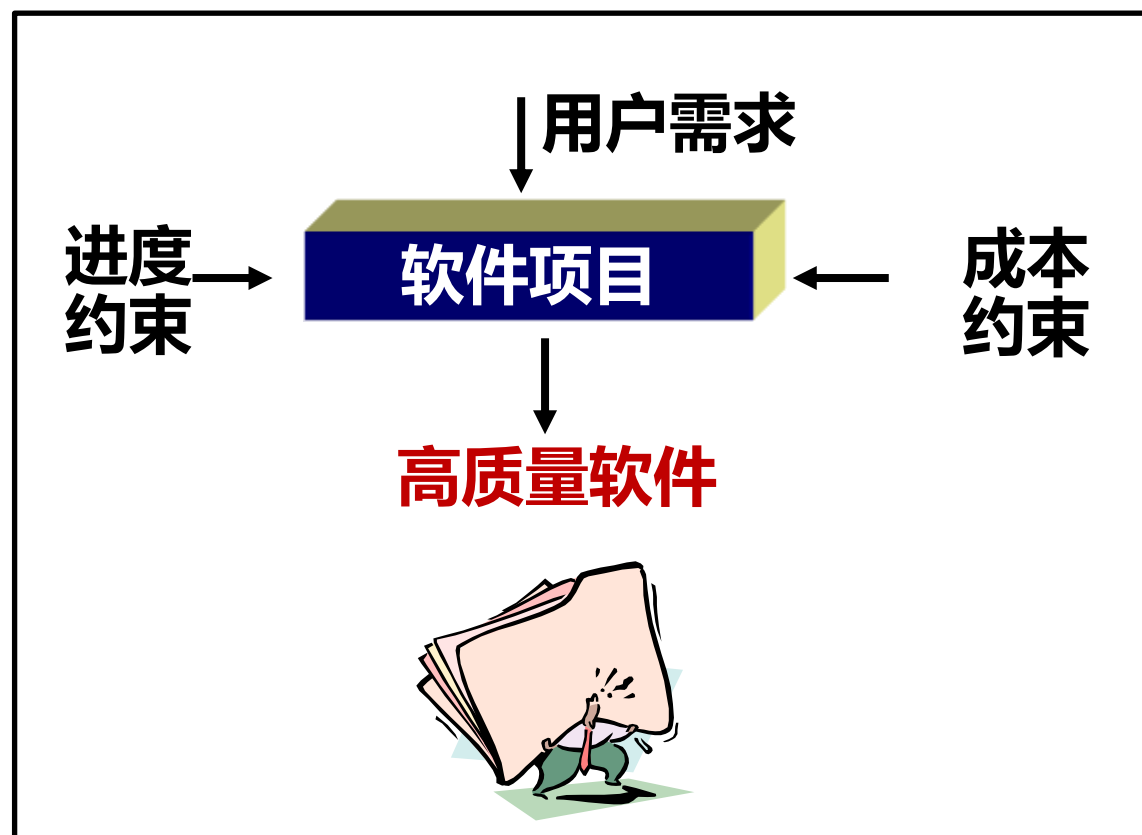
✓ 软件项目管理的常见方法和已有标准



2.1 软件项目的任务

□按照预定的**进度、成本和质量**，开发出**满足用户要求**的软件产品

- ✓用户需求
- ✓确保质量
- ✓成本限制
- ✓进度限制



- ✓ 目标性
- ✓ 进度性
- ✓ 约束性
- ✓ 多方性
- ✓ 独立性
- ✓ 不确定性

软件项目实施方法 – 工程化

- 按照过程
- 遵循原则
- 注重质量
- 借助工具
- 适时评审
- 开展测试
- 撰写文档
- 遵照规范
-

用软件工程方法来实施软件项目

- 军用软件工程过程
- 军用软件产品规范
- 军用软件质量保证方法
-

2.2 软件项目涉及的对象

项目经理、需求分析人员、软件
设计人员、程序员、测试人员、
质量保证人员等

人(参与项目
开发的人员)

模型、文档、代码
、数据、用例

物(软件制品)

软件项目
管理

需求分析、软件设计、编码实
现、软件测试、软件维护等

过程(软件开发过程)

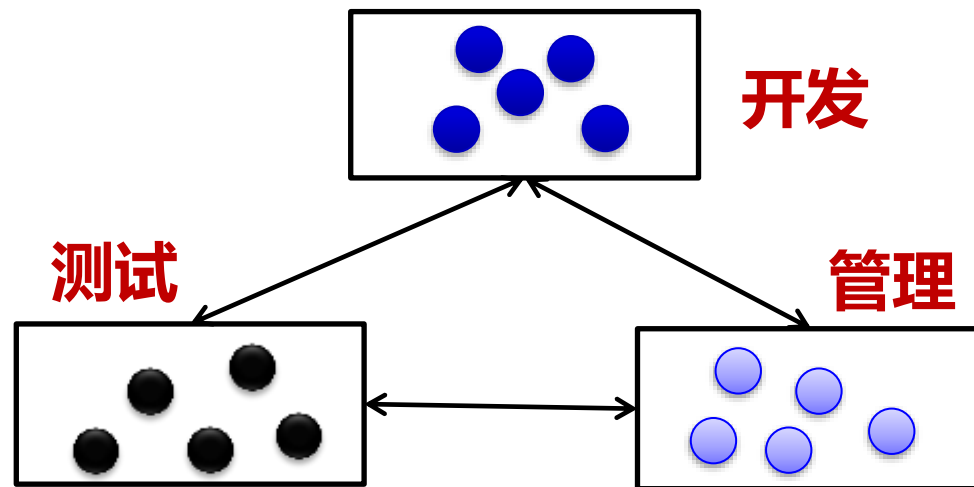
软件项目要管理好人、物和过程三类对象

示例：软件项目的人员及团队

□ Windows 7项目

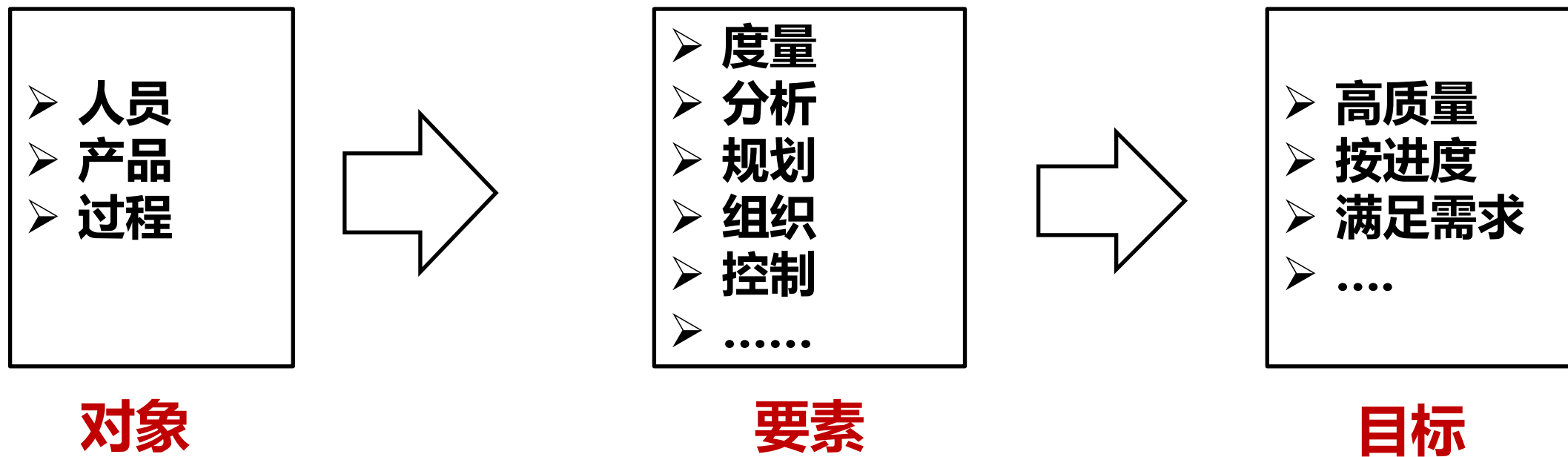
- ✓ 25个功能**团队**
- ✓ 每个团队由40名开发人员，40名测试人员，20名项目经理组成
- ✓ 整个项目总体约500个项目经理，1000名开发人员，1000名测试人员

**加强人员之间的交流和合作
非常重要！**



2.3 何为软件项目管理?

□对软件项目所涉及的过程、人员、产品、成本和进度等要素进行度量、分析、规划、组织和控制的过程，以确保软件项目按照预定的成本、进度、质量要求顺利完成



软件项目管理的对象

□过程管理

✓过程：怎么做(How)

□人员管理

✓人员：谁来做(Who)

□产品管理

✓产品：结果(What)

软件项目管理要管好三类对象
：过程、人员和产品

2.4 软件项目的管理要素

□ 管理软件过程

- ✓ 明确软件开发活动及过程：过程定义
- ✓ 估算软件项目工作量成本：软件度量
- ✓ 制定计划、跟踪过程、风险控制等

□ 管理软件产品

- ✓ 明确有哪些产品，呈什么形式(规范文档)
- ✓ 质量保证、配置管理、需求管理，风险控制

□ 管理项目人员

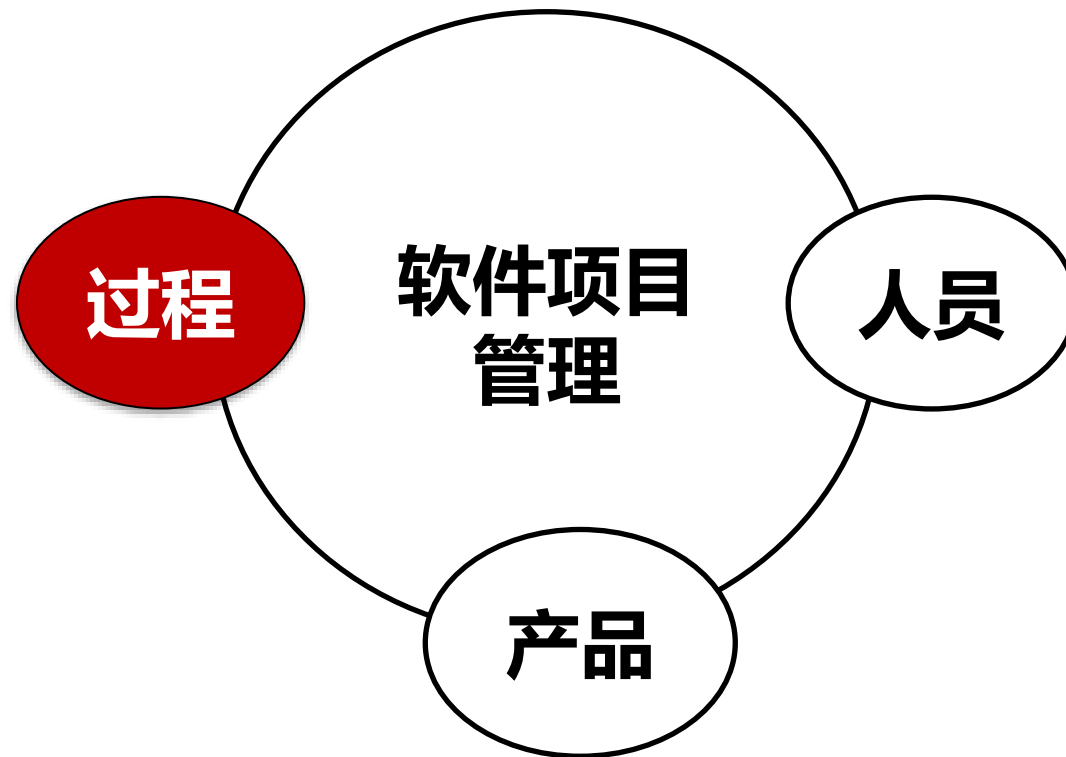
- ✓ 组建开发团队、调动积极性和激情
- ✓ 团队建设与沟通、机制设计、风险控制

软件项目管理要素

管理对象	人员	过程	产品
管理内容	团队建设和管理 纪律和激励机制	过程定义 软件度量 项目计划 项目跟踪	软件质量管理 软件配置管理 软件需求管理
	风险管理		

2.5 过程管理

- 过程定义
- 软件度量
- 项目计划
- 项目跟踪
- 风险管理



2.5.1 过程定义

□任务

- ✓定义、文档化软件开发过程、明确软件开发活动，得到一个**良定义、全面、灵活、简洁和可供剪裁的软件开发过程**

□需要关注的问题

- ✓如何根据要求定义软件开发过程？
- ✓如何对它文档化软件过程？
- ✓如何确保软件开发过程的有效性(包含必须的活动)、简洁性(舍弃不必要的过程和活动)和灵活性(根据具体项目情况进行剪裁)？
- ✓如何根据项目的特点剪裁过程？

定义和明确过程是软件开发的前提

示例：根据软件项目的特点来定义过程

□明确需求、严格的质量管控

- ✓以文档为中心的重型过程
- ✓瀑布模型、迭代和增量模型、螺旋模型等
- ✓大部分军用软件项目采用该方法

□变化和未确定的需求、快速应对变化、快速的交付

- ✓敏捷方法
- ✓互联网软件采用该方法

2.5.2 软件度量

□任务

- ✓对软件项目的过程、产品、资源的属性的**定量描述**，如软件项目的规模、成本、工作量、质量等
- ✓目的是为了对软件项目进行更好管理，如制定计划、质量保证等

□问题

- ✓需要对哪些方面进行度量？
- ✓如何进行度量？
- ✓如何将度量的结果指导软件项目的管理？
- ✓有哪些工具和模型可辅助进行软件度量.....

**软件项目管理需
定量的数据支持**

软件开发项目需要定量数据

□ 签订合同之时

- ✓ 客户和开发商都关心项目规模和合同额

□ 项目策划之时

- ✓ 项目组关心开发时间、投入成本和人力

□ 开发过程之中

- ✓ 项目组关注开发进度
- ✓ 用户和开发人员关注产品质量

□ 开发完成之后

- ✓ 项目实际投入和开销
- ✓ 项目的利润

□ **软件度量(Metrics)**是指对软件产品、软件开发过程或者资源的**简单属性**的定量描述

- ✓ **产品**：软件开发过程中所生成的各种文档和程序
- ✓ **过程**：与软件开发有关的各种活动，如软件设计等
- ✓ **资源**：软件开发过程中所需支持，如人员、费用等

□ **注意点**

- ✓ **定量描述**而不是定性描述
- ✓ **简单属性**：无需参照其它属性便可直接获得定量描述

□ **简单属性示例：代码行数、文档页数、操作符个数**

- ✓ 例子：**软件系统的代码行数目为1132 KLOC

□软件测量(Measure)

- ✓对软件产品、软件开发过程和资源**复杂属性**的**定量描述**，它是**简单属性度量值函数**，用于**事后或实时状态**，如软件可靠性

□注意点

- ✓**定量描述**而不是定性描述
- ✓**复杂属性**-不可直接获得、需参照其它属性的度量值
- ✓**实时或者事后状态**，用于对历史进行评估

□复杂属性示例

- ✓依据软件测试发现的错误数目，测量软件的质量、可靠性
- ✓例子：软件可靠性为0.95

□估算(Estimation)

- ✓对软件产品、软件开发过程和资源**复杂属性**的**定量描述**，它是简单属性度量值的函数，软件估算**用于事前**

□注意点

- ✓**定量描述**，而不是定性描述
- ✓**复杂属性**，不可直接获得、需参照其它属性的度量值
- ✓**事前状态**，可采用经验公式

□示例

- ✓如某个软件的开发成本为125万元

软件度量的对象

	产品	过程	资源	关注对象	难易程度
内部属性	(1)代码长度 (2)程序功能 (3)重用性 (4)模块的耦合和内聚度	(1)工作量 (2)计划和进度	(1)人 (2)软硬件环境 (3)方法 (4)经验	软件开发人员和项目管理人员	相对比较容易
外部属性	(1)可靠性 (2)可理解性 (3)质量 (4)可维护性 (5)可移植性	(1)成本 (2)可控制性	(1)成本 (2)时间	用户和软件项目管理人员	相对比较困难, 由内部属性决定

示例：面向规模的软件度量

□ 软件项目的规模是影响软件项目成本和工作量的主要因素

□ 用**软件代码行数**来表示软件项目规模

✓ **生产率** $PM = L / E$

L表示代码总量(单位：KLOC)，E表示软件工作量(单位：人月)

✓ 每千行代码的**平均成本** $CKL = S / L$

S为软件项目总开销

✓ **文档与代码比** $DI = Pd / L$

Pd表示文档页数

✓ **代码出错率** $EQRI = Ne / L$

Ne表示代码出错的数目



面向功能点的估算

□功能点计算公式

$$FP = CT * (0.65 + 0.01 * \sum F_i)$$

表 16.2 CT 值的加权计算

参数	加权因子			最终值
	简单	一般	复杂	
用户输入数	×3	×4	×6	CT ₁
用户输出数	×4	×5	×7	CT ₂
用户查询数	×3	×4	×6	CT ₃
文件数	×7	×10	×15	CT ₄
外部界面数	×5	×7	×10	CT ₅
CT =				Σ (CT _j) j=1..5

表 16.5 软件系统的功能点和代码行之间的对应关系

序号	程序设计语言	代码行/功能点
1	汇编语言	320
2	C	150
3	COBOL	105
4	FORTRAN	105
5	Pascal	91
6	ADA	71
7	PL/I	65
8	PROLOG/LISP	64
9	Smalltalk	21
10	代码生成器	15

表 16.3 F_i 及取值

F _i	复杂因素	F _i 的取值 (0,1,2,3,4,5) 0-没有影响 1-偶有影响 2-轻微影响 3-平均影响 4-较大影响 5-严重影响
F ₁	系统是否需要可靠的备份和复原	
F ₂	系统是否需要数据通信	
F ₃	系统是否有分布式处理功能	
F ₄	性能是否为临界状态	
F ₅	系统是否在一个实用的操作系统下运行	
F ₆	系统是否需要联机数据项	
F ₇	联机数据项是否在多界面或多操作之间进行切换	
F ₈	是否需要联机更新主文件	
F ₉	输入、输出、查询和文件是否复杂	
F ₁₀	内部处理是否复杂	
F ₁₁	代码是否需要被设计成可重用	
F ₁₂	设计中是否需要包括转换和安装	
F ₁₃	系统的设计是否支持不同组织的多次安装	
F ₁₄	应用的设计是否方便用户修改和使用	



示例：面向规模的软件度量

□基于经验模型的估算

- ✓ Barry Boehm 1981年提出了COCOMO模型，用于软件项目的规模、成本和进度的估算
- ✓ **构造性成本模型** CoCoMo (Constructive Cost Model)
- ✓ 基本CoCoMo模型： $E = a * (kLOC)^b$ 。其中E是软件系统的工作量(单位：人月)，a和b是经验常数
- ✓ $D = c * E^d$ 。其中D是开发时间（单位：月），c和d是经验常数

软件类型	a	b	c	d	适用范围
组织型	2.4	1.05	2.5	0.38	各类应用程序
半独立型	3.0	1.12	2.5	0.35	各类实用程序、编译程序等
嵌入型	3.6	1.20	2.5	0.32	各类实时软件、OS、控制程序等

2.5.3 项目计划

□任务

- ✓ **制定和文档化软件项目计划**，确保软件开发计划是可行、科学、符合实际的

□问题

- ✓ 要对软件开发过程中的那些方面制定计划？
- ✓ 制定软件项目的计划的基础和依据是什么？
- ✓ 要考虑哪些方面的问题？
- ✓ 如何确保计划是科学的和可行的？（软件度量）
- ✓ 如何描述计划？利用哪些工具可辅助计划的制定？

**软件项目计划有
助于项目有序地
开展和实施**

软件开发也需要计划

□合同和策划阶段

- ✓ 制定软件开发计划
- ✓ 软件项目计划分发到各个开发成员
- ✓ 每个成员准确了解各自任务和工作以及实施进度要求

□项目实施过程

- ✓ 根据软件开发计划有序实施项目
- ✓ 发现项目实施中存在的问题：如进度滞后
- ✓ 出现偏差后调整软件项目计划

□项目完成之后

- ✓ 总结项目计划实施情况
- ✓ 分析原因，指导后续其他项目的计划制定

计划有助于项目的有序实施

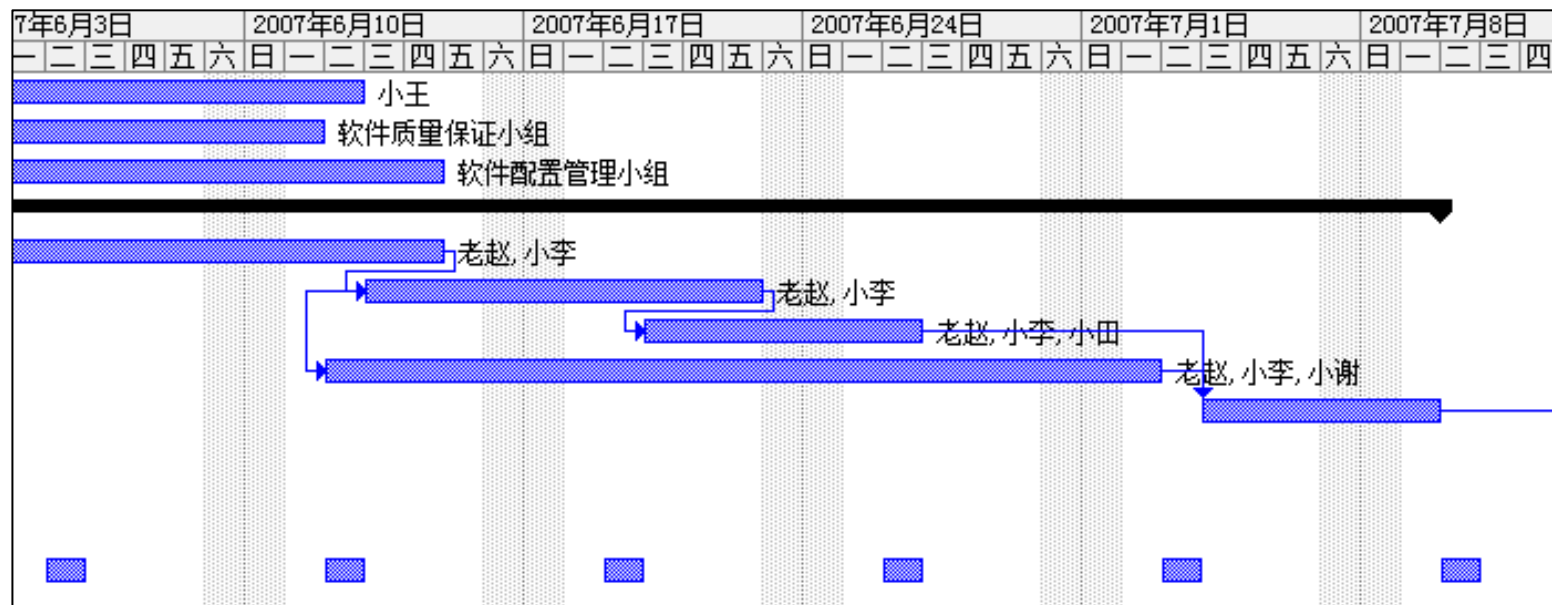
什么是软件项目计划?

□软件项目计划

- ✓对软件项目实施所涉及的活动、人员的安排、任务的划分、开发进度、资源的分配和使用等方面作出的**预先规划**

□计划内容

- ✓软件开发**进度计划**
- ✓人员任务**分工计划**
- ✓资源分配**使用计划**
- ✓.....



制定软件项目计划的基础和依据

□软件开发过程

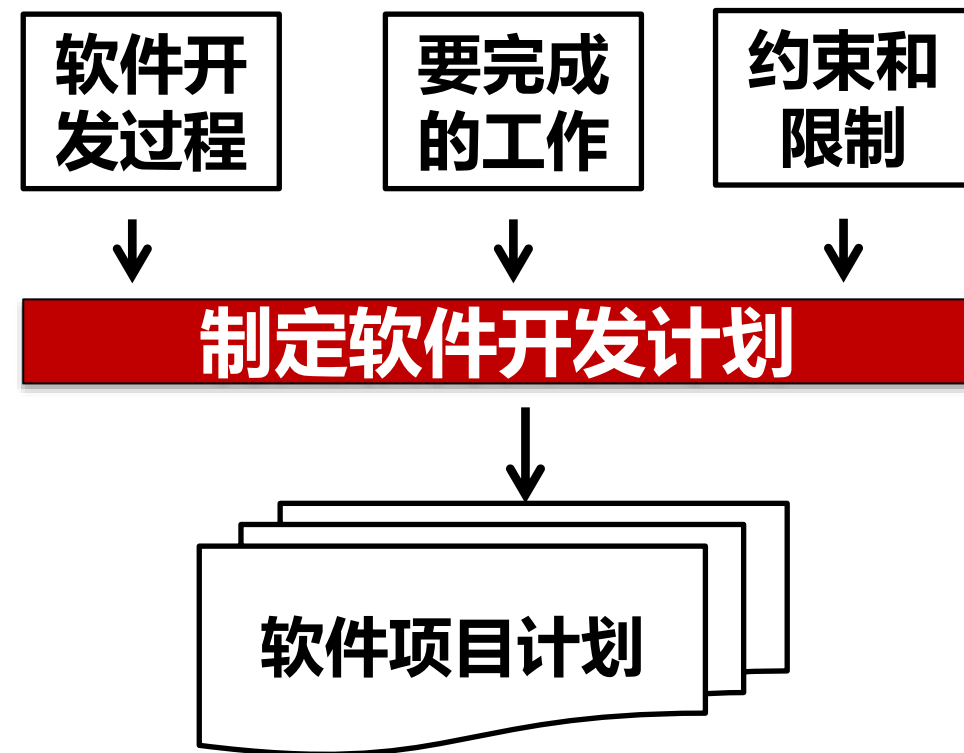
- ✓ 选择什么样的软件开发过程
- ✓ 支持敏捷方法过程：迭代模型

□软件项目任务

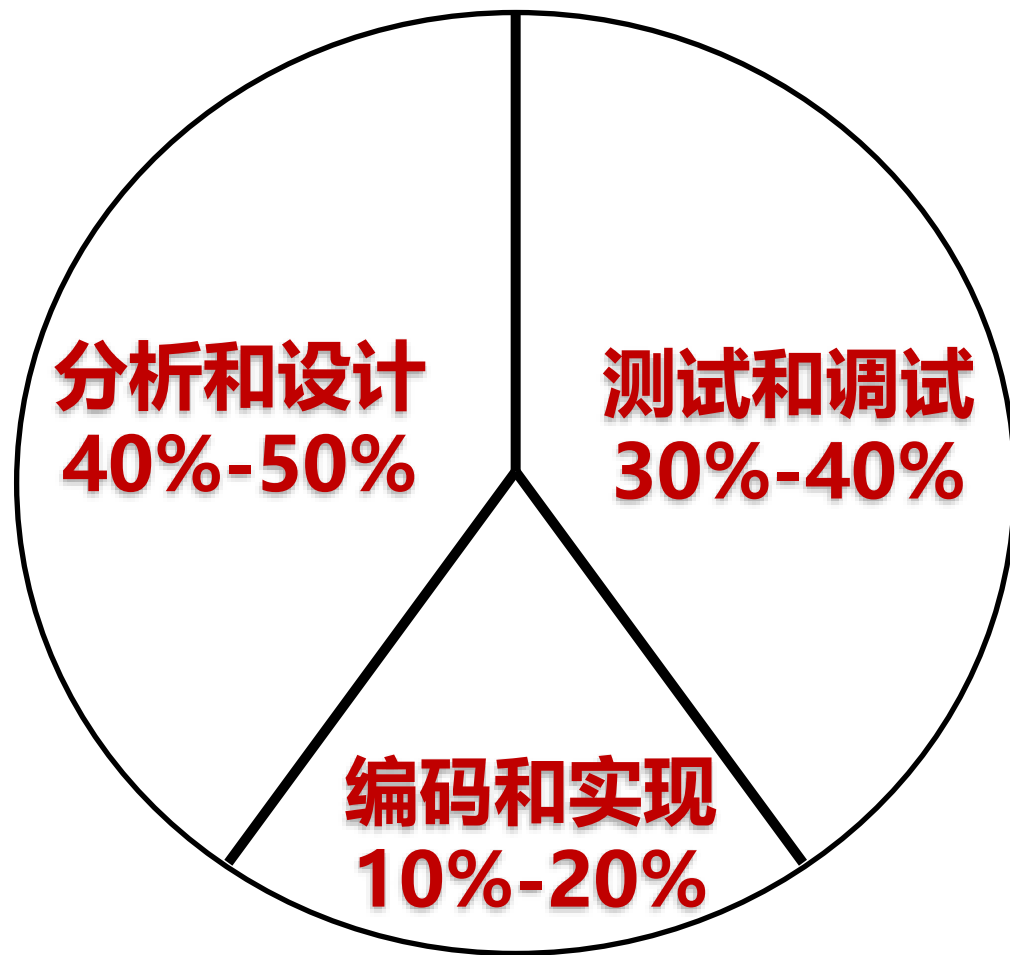
- ✓ 工作说明和软件需求
- ✓ 历史数据和估算模型
- ✓ 估算工作量和成本

□项目约束限制

- ✓ 项目投入资源及其质量
- ✓ 资源（经费、时间、人员、...）



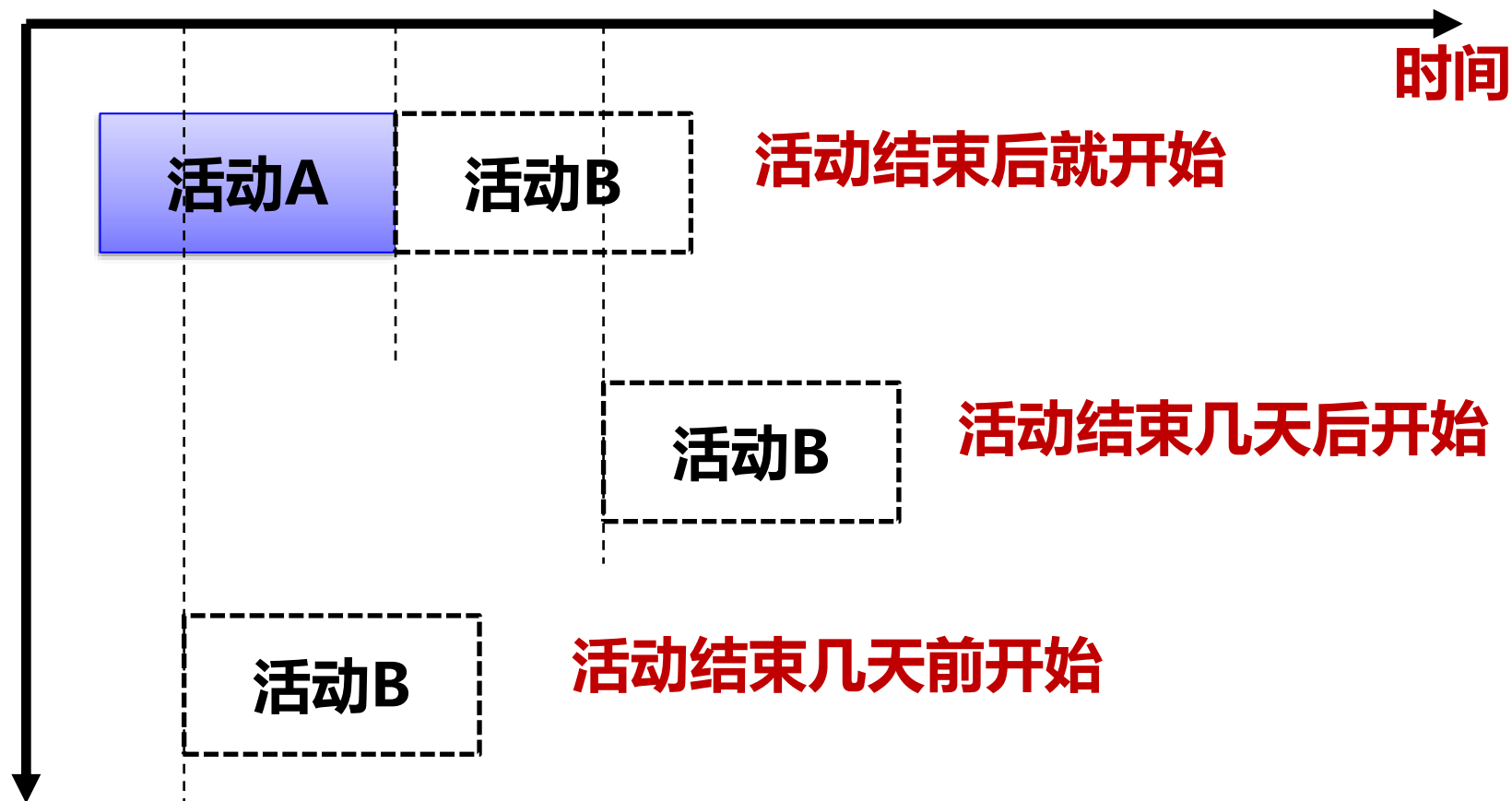
软件开发工作量的大致分布





软件开发活动之间的关系 (1/3)

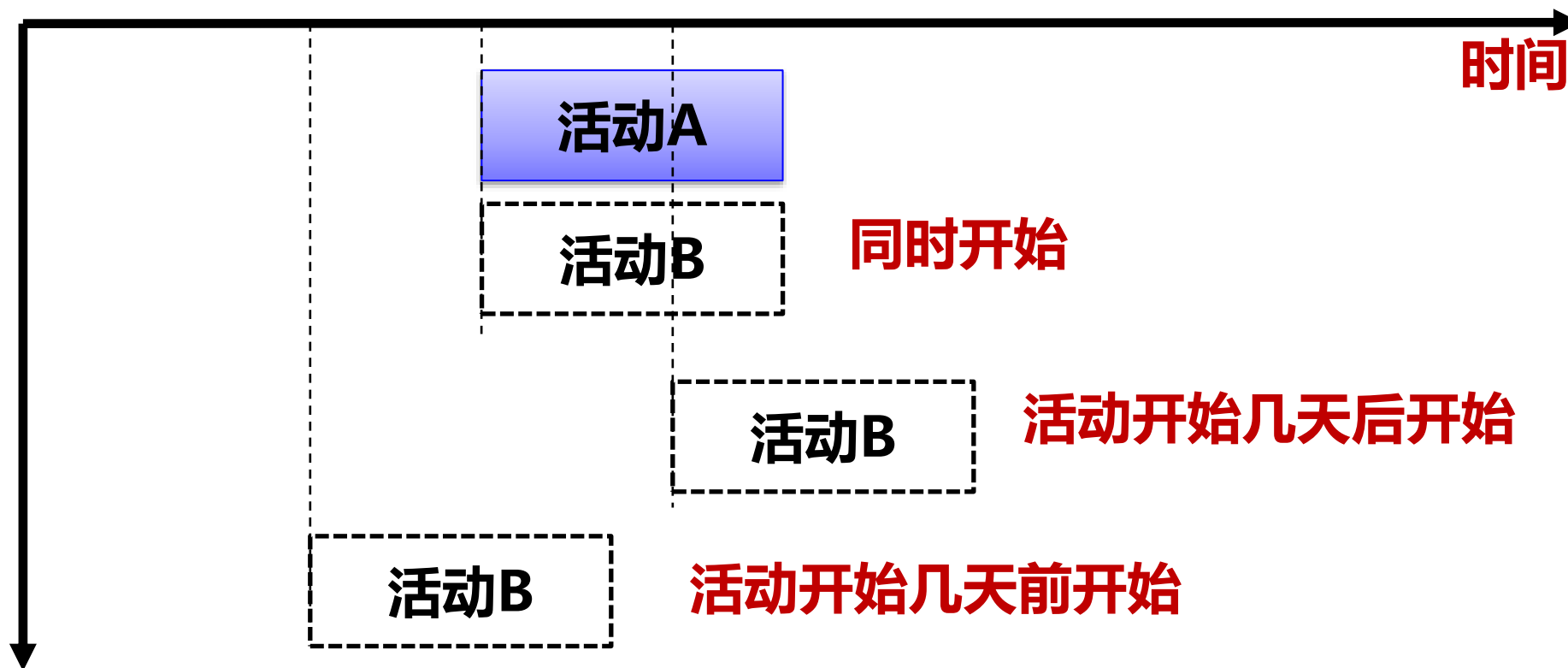
□结束到开始





软件开发活动之间的关系 (2/3)

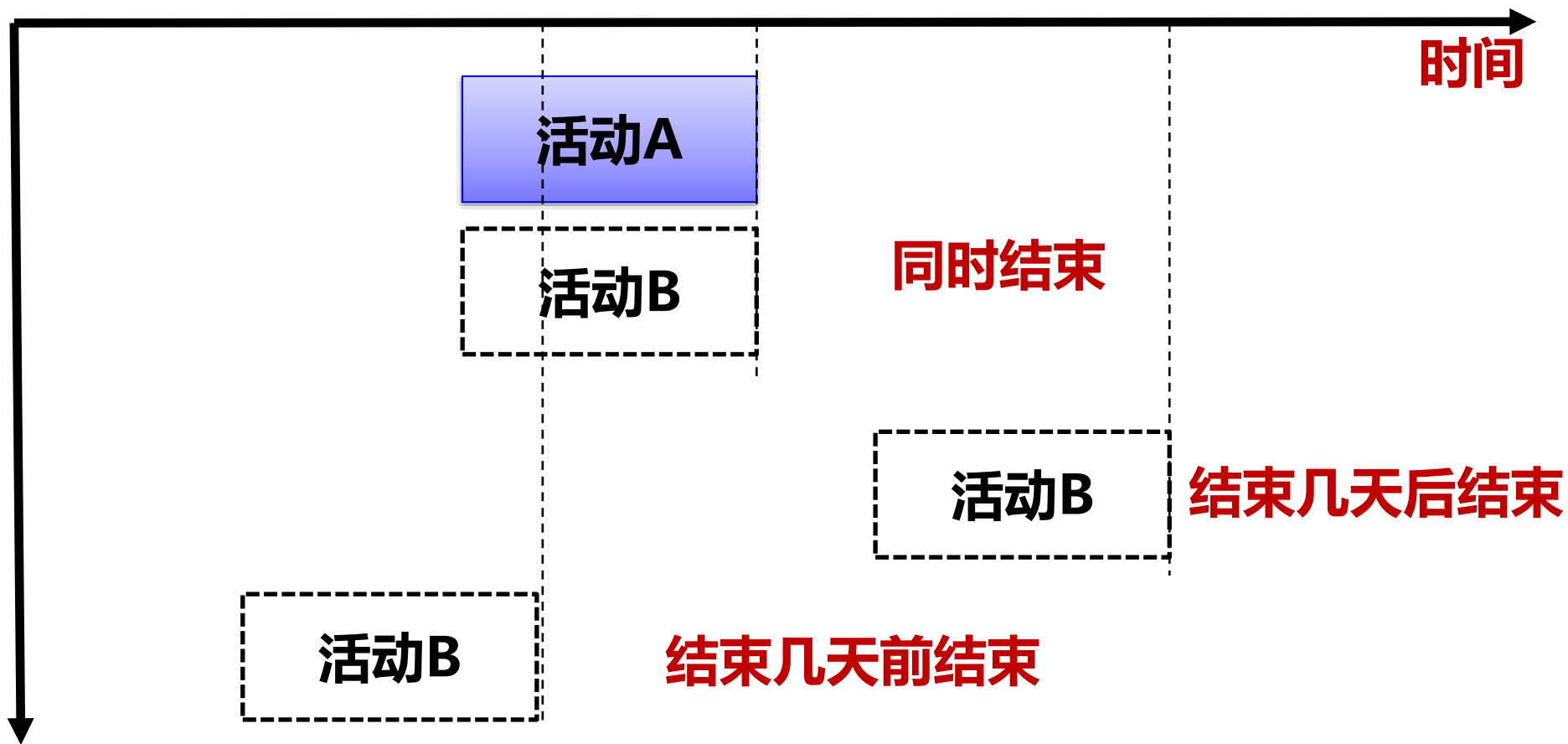
□开始到开始



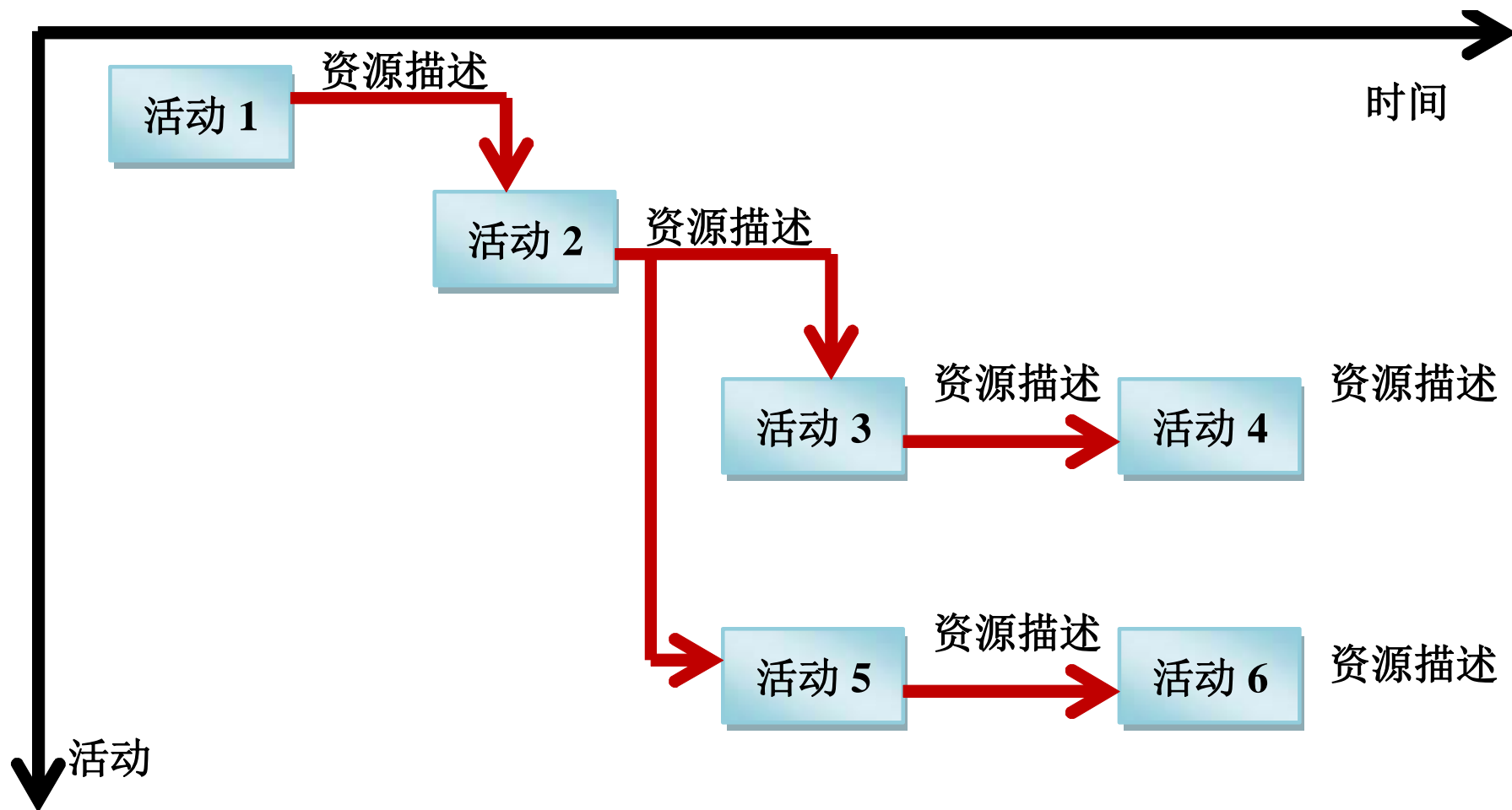


软件开发活动之间的关系 (3/3)

□结束到结束



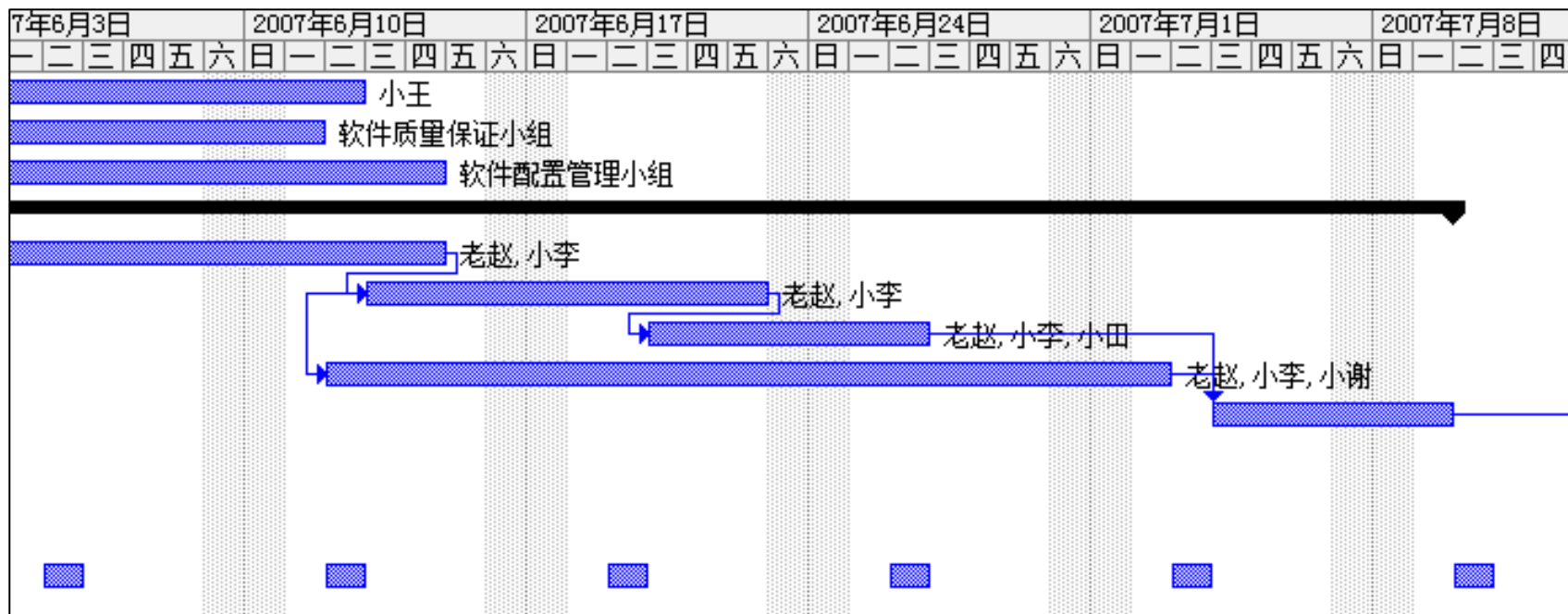
基于甘特图表示的软件项目计划



描述项目进度计划-甘特图

□甘特图

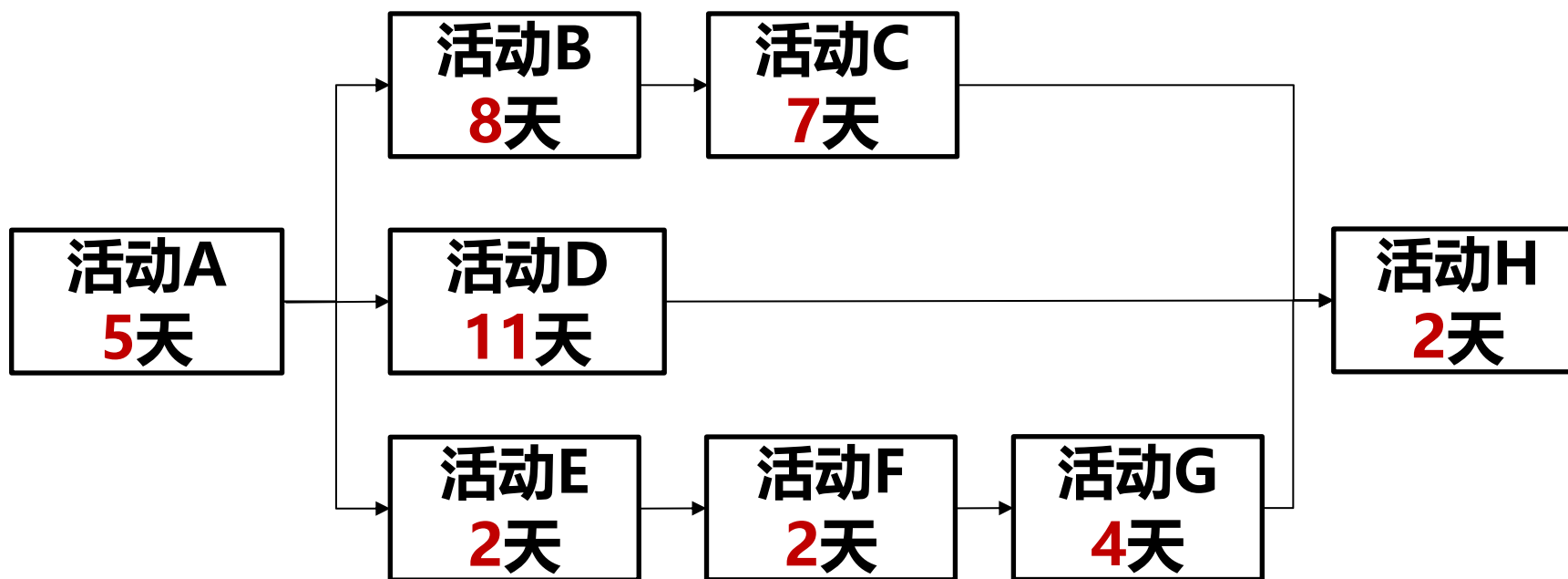
- ✓左部工作表，以文字方式显示任务信息，如任务名称，开始和结束日期等
- ✓右边的图表，以条形图方式显示任务信息





关键路径分析

- **关键路径**是指软件项目进度计划中从起始活动开始到结束活动为止，具有**最长长度**的路径。长度是指软件开发时间
- 关键路径上的活动实施进度直接影响整个项目的开发进度；如果关键路径上的进度受到影响，整个项目的开发进度肯定会受到影响



活动责任矩阵 (1/2)

□ 用于定义与软件开发活动执行、评审和批准相关的人员和角色

□ 软件开发活动 - 角色责任矩阵表

软件开发活动 角色	执行	负责	评审	批准
需求分析	需求分析小组	需求分析小组组长	用户方代表 需求分析小组 软件设计小组 质量保证小组 软件测试小组	软件项目负责人 用户方负责人

活动责任矩阵 (2/2)

□角色 - 人员责任矩阵表

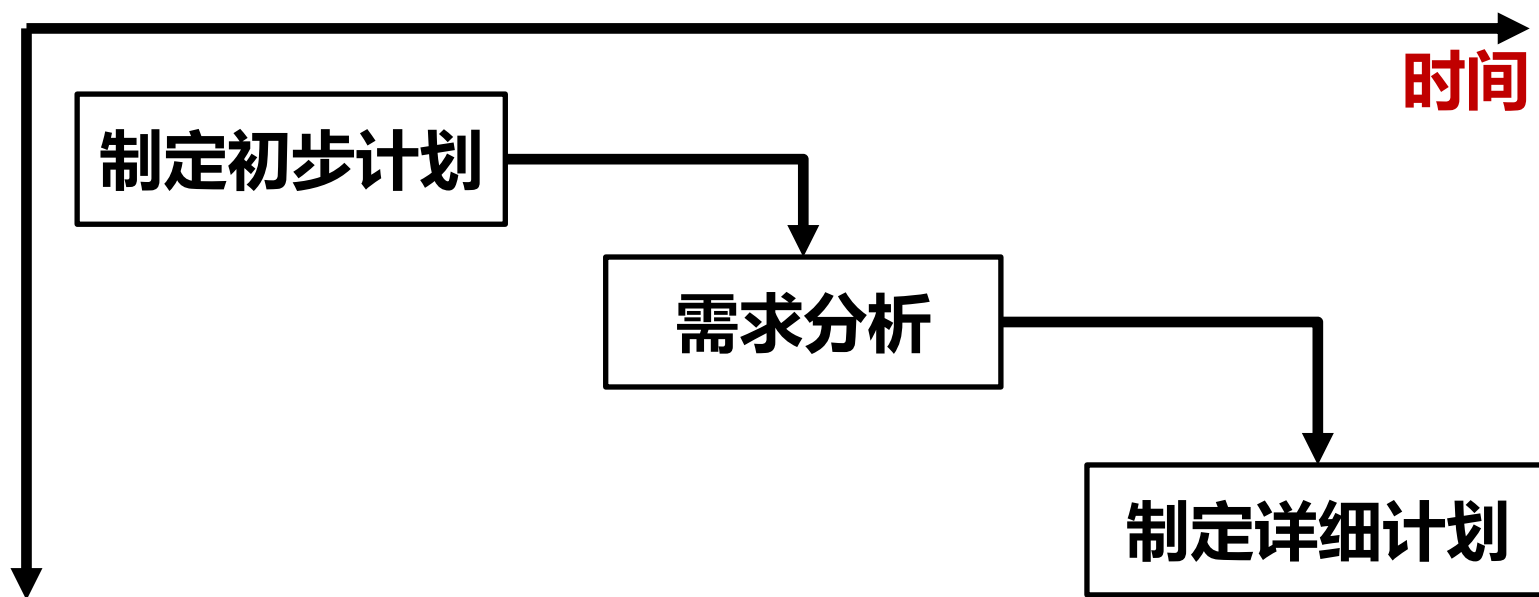
角色	人员
需求分析小组	小张、小李、小王
需求分析负责人	小张
软件项目负责人	小宋
用户方代表	小张
用户方负责人	小董



制定软件项目计划的时机

□软件项目计划一般是在软件项目**实施之初制定**

- ✓项目开始之初，制定一个初步的软件项目计划，用于指导后续短期的软件开发工作，如需求分析工作
- ✓软件需求分析完成之时，制定详细的软件项目计划，用于指导后续长期的软件开发工作



参与、承诺和分发软件项目计划

□共同参与

- ✓一起讨论、明确任务、估算周期、确认进度

□通过会议形式参与制定

- ✓便于加强交流

□评审和承诺

- ✓共同评审
- ✓承诺计划内容

□分发和宣传

- ✓分发给所有的项目组成员
- ✓让他们了解软件项目计划

2.5.4 项目跟踪

□任务

- ✓跟踪和及时调整软件项目开发计划，**提供项目情况的可视性**，保证计划的适时调整

□问题

- ✓要对哪些方面进行跟踪？
- ✓如何对软件项目进行跟踪？
- ✓当无法按预定计划实施时如何调整计划？
- ✓当跟踪发现问题时如何处理
- ✓如何提供工具辅助对软件项目进行跟踪.....

项目跟踪有助于发现项目实施中存在的偏差：实际与计划不符



为什么要进行软件项目跟踪

□ 随时掌握软件项目的实际开发情况，提供软件项目的实施提供可视性

✓ 哪些地方出现了什么样的问题

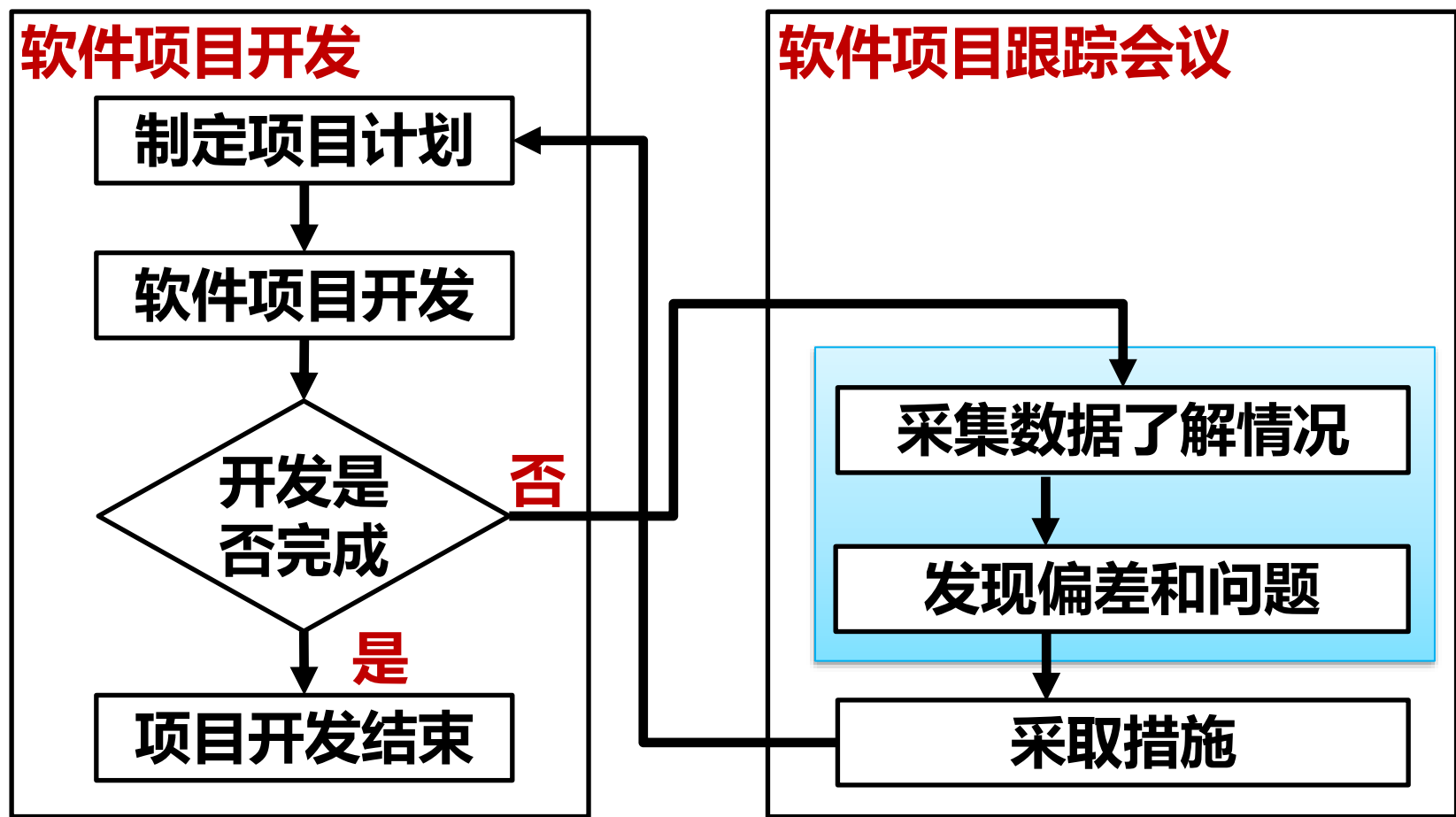
□ 跟踪的对象

✓ **项目问题和风险**，软件项目在实施过程中会出现各种各样的问题和风险

✓ **软件项目进展**，软件项目的实际进展与软件项目计划二者之间会产生偏差

项目跟踪的步骤和方法

□ 周期性的项目跟踪会议



2.5.5 风险管理

□任务

- ✓对软件开发过程中各种**风险进行分析、预测、评估、监控**的过程

□问题

- ✓什么是软件开发风险？
- ✓软件开发可能会有哪些风险？
- ✓如何客观地预测风险？
- ✓如何评估风险带来的影响？
- ✓如何避免和消除风险？
- ✓如何提供工具支持风险分析？

**风险管理有助于
发现软件开发过
程中存在的风险
： 实施中的问题**

□软件风险

- ✓使软件项目的实施受到影响和损失、甚至导致失败的、可能会发生的事件
- ✓例如，人员的临时流失，计划过于乐观，设计的低劣

□软件风险特点

- ✓事先难以确定
- ✓带来损失，影响项目实施，甚至会导致项目失败

示例：需求风险

- 需求已经成为项目基准，但仍在变化
- 需求定义欠佳：不清晰、不准确、不一致
- 增加额外的需求

示例：产品风险

- ❑ 错误率高的模块，需要更多时间对它进行测试、设计和实现
- ❑ 矫正质量低下的不可接受的产品需要更多的时间对它进行测试、设计和实现
- ❑ 由于功能错误，导致需要重新进行设计和实现
- ❑ 开发额外不需要的功能延长了进度
- ❑ 要满足产品规模和速度要求，需要更多的时间
- ❑ 严格要求与现有系统兼容，需要更多的时间
- ❑ 要求软件重用，需要更多的时间
- ❑

示例：人员风险

- 招聘人员所需的时间比预期要长
- 作为人员参与工作的先决条件(如培训、其他项目的完成等)不能按时完成
- 开发人员与管理层关系不佳导致决策迟缓、影响全局
- 项目组成员没有全身心地投入到项目中，因而无法达到所需的产品功能和性能需求
- 缺乏激励措施、士气低下，降低生产能力
- 缺乏必要的规范，增加工作失误，重复工作，降低工作质量
- 缺乏工作基础(语言、经验、工具等)
- 项目结束前，项目组成员离开项目组



风险管理模式

□ 危机管理

- ✓ 类似于救火模式，听任软件风险的发生，及至造成麻烦后才着手进行处理

□ 失败处理

- ✓ 听任软件风险的发生和演化，只是在风险发生之后才采取应对措施

□ 风险缓解

- ✓ 识别软件风险，事先制定好风险发生后的补救措施，但不做任何防范措施

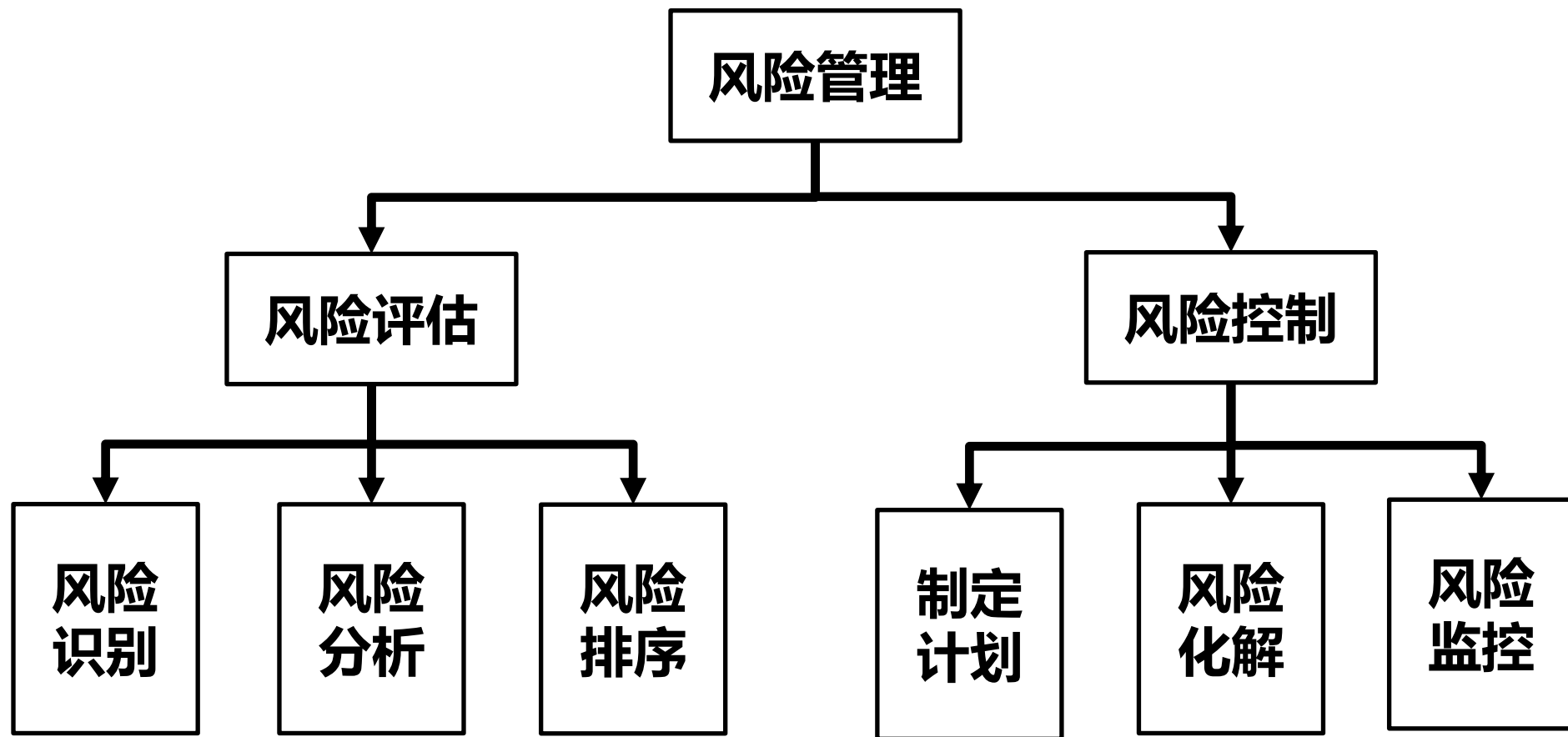
□ 风险预防

- ✓ 预先识别和分析哪些不好事件可能会发生，制定好了万一发生的应对措施，同时采取措施防止它发生

□ 消灭根源

- ✓ 不仅要识别出软件开发过程中各种潜在的软件风险，而且还要分析导致这些软件风险发生的主要因素，并采取积极的措施消除软件风险产生的根源

风险管理的组成





风险管理 (1/4)

□风险识别

- ✓识别软件项目可能存在的各种潜在软件风险

□风险分析

- ✓评估各项软件风险发生概率、可能造成损失及软件风险危险度

编号	风险名称	风险概率	损失 (人周)	危险度(人周)
1	软件项目规模的估算结果过于乐观	0.7	8	5.6
2	软件产品的交付日期提前	0.2	4	0.8
3	用户增加了额外的需求	0.8	5	4.0
4	需求分析工程师不能按时到位	0.9	2	1.8
5	需求分析所需的软件工具尚未到位	0.5	3	1.5



风险管理 (2/4)

□风险优先级

✓ 据软件风险的危险数，可以对软件风险的优先级进行排序

编号	风险名称	风险概率	损失 (人周)	危险度(人周)
1	软件项目规模的估算结果过于乐观	0.7	8	5.6
6	由于业务繁忙，用户没有足够多的时间配合需求分析小组开展需求调查工作	0.7	6	4.2
3	用户增加了额外的需求	0.8	5	4.0
4	需求分析人员不能按时到位	0.9	2	1.8
5	需求分析分析所需的软件工作尚未到位	0.5	3	1.5
2	软件产品的交付日期提前	0.2	4	0.8



风险管理 (3/4)

□制定风险管理计划

软件风险管理计划	
风险编号	2
风险名称	小刘离开项目组
风险发生的对象	小刘
风险发生的原因	未知
风险可能发生的时机	二周后
消除风险的措施	由软件项目负责人小王和小刘交互， 询问离开软件项目组的真正原因，并 及时向高层反映情况
风险发生后的应对措施	让小陈接替小刘的工作



风险管理 (4/4)

□风险化解方式

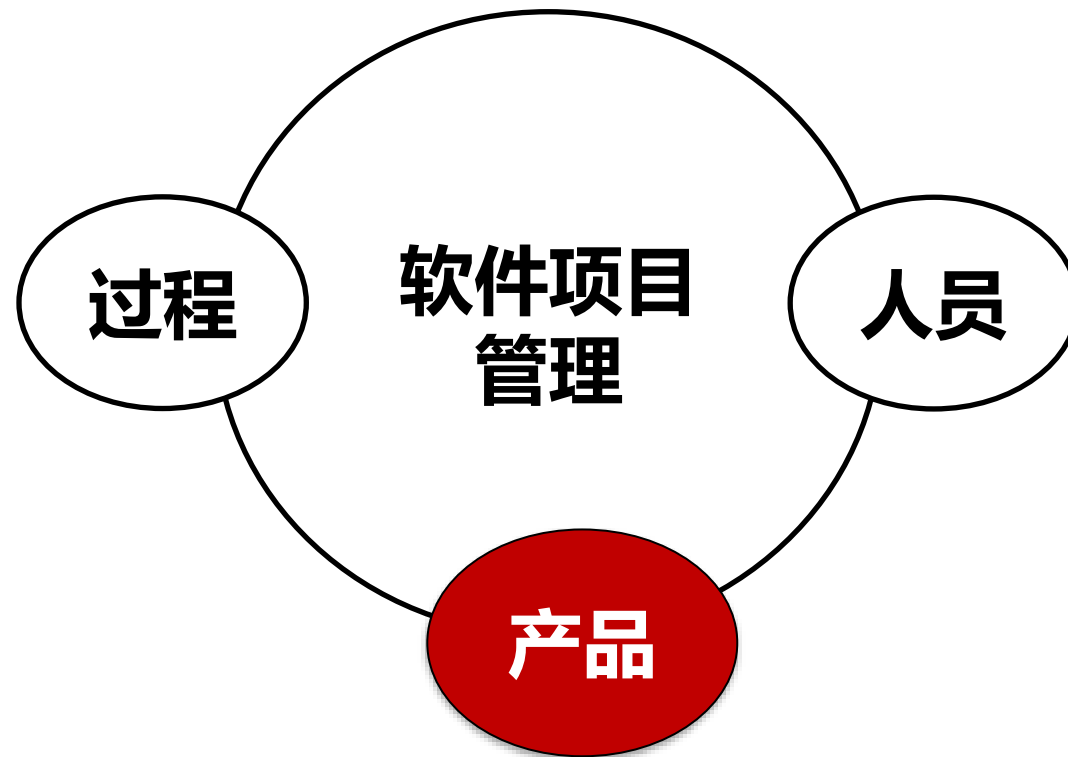
- ✓避免风险、转移风险、消除发生软件风险的根源

□风险监控

- ✓对软件风险的化解程度及其变化（如发生概率、可能导致的损失和危险度）进行检查和监控，并记录收集到的有关软件风险信息，以促进对软件风险的持续管理

2.6 产品管理

- 软件质量保证
- 软件配置管理
- 软件需求管理
- 风险管理



2.6.1 软件质量保证

□任务

- ✓为软件产品的**质量提供某种可视性**，知道哪些地方有质量问题，便于改进方法和措施，提高软件产品的质量

□问题

- ✓高质量的软件体现在哪些方面？
- ✓如何发现和保证软件产品的质量？
- ✓如何制定软件产品质量保证计划？
- ✓如何有效落实和实施计划？
- ✓如何提供工具支持软件质量保证？

**软件质量保证是
软件项目管理的重要目标**

□什么是软件质量保证

- ✓为管理层提供为获知产品**质量信息**所需的数据，从而获得产品质量是否符合预定目标的认识和信息

□软件质量保证的目的

- ✓为软件产品的质量提供**可视性**
- ✓知道哪些地方有质量问题
- ✓便于改进方法和措施

□示例

- ✓对软件进行测试，发现有哪些错误
- ✓掌握这些错误的性质(严重、一般)

软件质量保证的内容

□掌握软件产品质量

- ✓软件测试

□提交软件质量报告

- ✓软件测试报告，说明质量问题

□汇报项目组和管理层

- ✓例行的质量回报，便于改进管理和技术手段

从哪些方面关注软件质量

□ 软件产品

- ✓ 软件需求规格说明书
- ✓ 软件设计规格说明书
- ✓ 源程序代码

□ 开发活动

- ✓ 需求分析
- ✓ 软件设计
- ✓ 编码

□ 标准和规程

- ✓ 代码编写标准
- ✓ 文档规范

如何保证软件质量

□软件产品

- ✓文档类：审核，产生审核报告
- ✓代码类：测试，产生测试报告

□开发活动

- ✓审查：产生审查报告

□制定标准和规程

- ✓组织内部或者在项目开始时要制定软件开发的标准和规程

谁来执行和实施软件质量保证

- 软件项目质量保证小组(SQA小组)
- 独立于项目开发小组
- 具有比较大的权限

软件质量保证活动和手段

- 正确理解用户要求
- 制定相关标准规程
- 审查软件开发活动
- 审核软件工作产品
- 测试程序代码
- 记录各种偏差
- 记录不符合项，报告高级管理者

2.6.2 产品管理 – 软件配置管理

□必要性

- ✓对于同一软件制品，可能需要对它进行多次变更产生不同形式和版本，对软件制品的形式和版本进行配置管理

□任务

- ✓控制软件制品的标识、存储、变更和发放
- ✓记录、报告软件制品的状态
- ✓验证软件制品的正确性和一致性
- ✓对上述工作进行审计

为什么需要SCM?

- 软件产品的**易改性与可控性**
- 修改很可能**引入新的错误**, 使结构变坏
- **牵一发而动全身**(影响域)
- 团队开发时, 多人**并发存取**需加控制(存取控制)
- 多应用开发时, 同一软件的不同版本可能对应于不同应用, 对此需加控制(版本控制)
- 应对软件更动状态予以**追踪**, 并及时向有关人员通报状态情况
- 如果软件产品不能自始至终地保持清晰、互相一致, 造成混乱、丢失, 那么该软件系统会因无法使用而不得不报废

□软件生命周期内产生、需进行配置管理的工作产品

- ✓文档：SRS/SDS/...
- ✓代码：源代码、中间代码、可执行代码、...
- ✓数据：配置数据、数据库、数据文件、...
- ✓标准和规约：编码规范、...

□命名和编号

✓如SRS

□属性

✓版本(1.0)

✓类型(文档)

□关系

✓与软件概要设计文档、数据设计文档关联

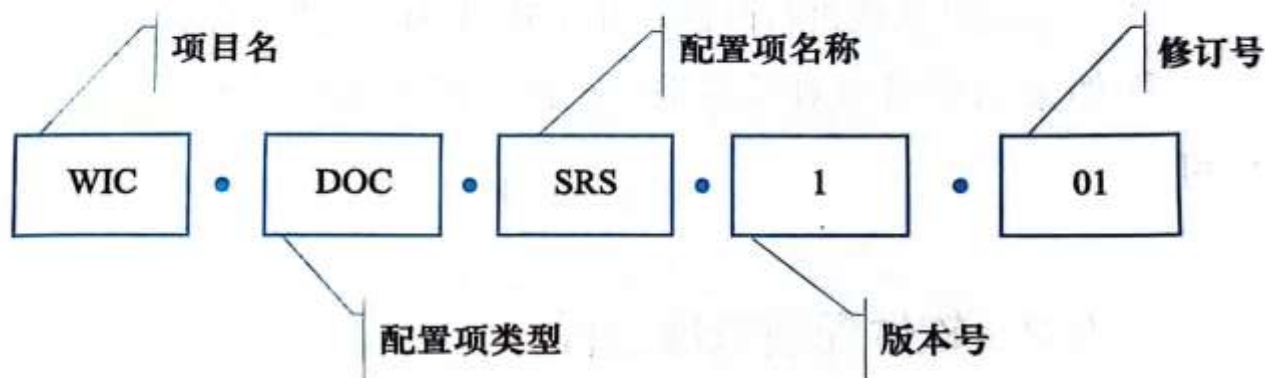


图 16.16 配置项标识模板示意

□ 已经通过正式复审和批准的**软件产品、标准或规约**,

- ✓ 它们可以作为进一步开发的基础
- ✓ 只能通过正式的变化控制过程才允许对它们进行变更

□ 基线示例

- ✓ 经过评审后的，发现的问题已经得到纠正，用户和项目组双方认可并且正式批准，就可纳入基线

- 纠错、改进、完善、扩充等工作导致同一配置项有多个版本；
- 在同时进行多个软件项目开发，同一配置项可能需要多个不同的版本；
- 软件配置管理者应提供有效的手段区分和描述配置项的多个不同版本，以及版本之间的关系和演化。

□ 不受控制的变化会导致软件制品混乱。对变更应遵循以下过程：

- ✓ 提出书面变更申请；
- ✓ 评估变更申请；
- ✓ 提取配置项；
- ✓ 修改配置项；
- ✓ 质量保证；
- ✓ 纳入基线。

2.6.3 软件需求管理

□任务

- ✓获取、文档化和评审用户需求，对用户需求变更进行控制和管理

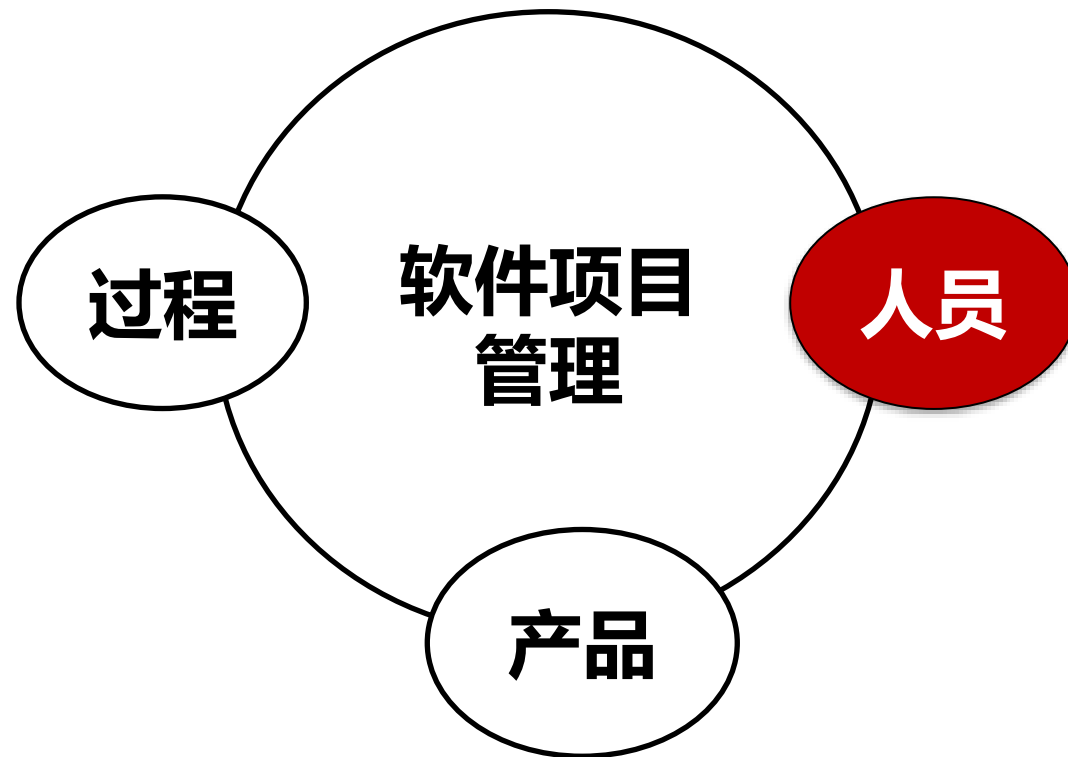
□问题

- ✓如何获取需求？
- ✓如何撰写软件需求规格说明书？
- ✓如何对需求进行评审以发现问题？
- ✓如何控制需求的变更？
- ✓如何提供工具支持需求管理.....

**软件需求管理为
应对需求变化提
供有效措施**

2.7 人员管理

- 软件项目团队
- 纪律激励机制



2.7.1 软件项目团队

□任务

- ✓ 确定开发**团体的结构、明确人员的角色和任务、加强人员间的交流与合作**，结构合理、任务明确、团结协作、交流顺畅

□问题

- ✓ 如何根据开发组织、软件项目和开发人员的特点来组建项目团队？
- ✓ 如何采取有效的措施来加强和促进人员之间的交流、沟通和合作？
- ✓ 如何提高团队的合作精神？
- ✓

2.7.2 纪律和激励机制

□任务

- ✓制定和实施纪律约束团体，通过激励机制**激发人员的激情，严格的纪律**，饱满的激情

□问题

- ✓如何制定有效的纪律确保项目得以顺利的实施
- ✓如何制定措施激励员工的积极性和热情
- ✓.....

管理在软件项目中的重要性

- **70%**的软件项目由于**管理不善**导致难以控制进度、成本和质量，三分之一左右的软件项目在时间和成本上超出额定限度**125%以上**
- **管理**是影响软件项目成功实施的**全局性因素**，而**技术**仅仅是**局部因素**
- 如果软件开发组织不能对软件项目进行有效管理，就难以充分发挥软件开发方法和工具的**潜力**，也无法高效率地开发出高质量的软件产品
- 历史上由于管理不善而导致软件项目失败的例子比比皆是，如美国国税局税收现代化系统、美国银行的MasterNet系统等

1. 软件项目及其特点

✓ 软件项目管理的特殊性

2. 软件项目管理内容

✓ 软件项目管理的对象和要素



3. 软件项目管理的方法和标准

✓ 软件项目管理的常见方法和已有标准

3.1 与软件项目管理相关的国军标

□ **GJB500A - 军用软件研制成熟度模型**

□ **GJB9001B – 2009，对承担军用产品、研制、生产、试验和维修任务的组织是必须执行的质量管理标准**

- ✓ 提出了武器装备“六性”的概念，即可靠性、维修性、保障性、安全性、测试性、环境适应性
- ✓ 强化了风险管理和软件工程管理要求

3.2 研制过程：以文档为中心的重型管理方式

- 需求→研制总要求提出各种功能和性能指标
- 设计→专家评审
- 研制→过程管理
- 验收→测试和评审

现行管理制度下的一些困局

- 产品陈旧，总是赶不上潮流，并且越来越远
- 队伍不能持续
- 老队伍总是被新型号淘汰
- 文档发挥的作用不大
- 过程复杂，研制周期长，无法快速提交产品
- 需求调整困难（需求不可能一次获取）
- 技术实现难以调整

3.3 软件项目的团队组织方法

- 由一帮志同道合的人员组成
- 一致、共同的集体目标
- 分工明确，相互合作，共同完成任务

- 跑步接力赛
- 软件项目开发
-

3.4 软件项目团队的运行模式(1/2)

□一窝蜂模式

- ✓无组织，一窝蜂，无序和随意
- ✓典型例子是小孩子游戏

□主治医生模式

- ✓主治医生主刀，其他人员协助
- ✓容易产生一人干活，其余打酱油

□社区模式

- ✓志愿者因为兴趣参加，没有报酬，众人拾柴火焰高
- ✓只烤火不拾柴，柴火质量低

□功能团队模式

- ✓具备不同能力的同事平等协作，共同完成功能
- ✓一个功能完成之后，这些人又重组织，完成其他功能
- ✓人员之间没有管理关系，小组内部交流频繁

□官僚模式

- ✓大领导--》小领导--》员工
- ✓存在明显的领导和管理关系
- ✓跨组织合作变得困难

3.5 团队中的合作

- 交流与沟通：面对面、基于互联网和平台
- 共同解决问题
- 协商和消解冲突
- 多从自己的角度考虑问题
- 多为他人提供帮助
- 有些工作(如联调)必须联合开展
-

基于敏捷和开源的组织 and 开发模式

- 迭代性的开发 – 分阶段持续提交可演示产品
- 以代码为中心 – 开源软件
- 以敏捷为手段 – 敏捷方法
- 必须系统模型 – UML模型
- 适当软件文档 – 文档规范

□软件项目的工程特征

- ✓进度、成本、质量等约束，涉及人员、过程、产品等管理对象

□软件项目的管理内容

- ✓包含人员管理、过程管理、产品管理等多个方面

□软件项目的常见管理方法

- ✓结合项目特点选择合适的过程模型、组织项目团队
- ✓管理好软件制品，确保质量，发现风险