CS10102302
软件开发管理

赵君峤 副教授

计算机科学与技术系 电子与信息工程学院

同济大学

教学提纲

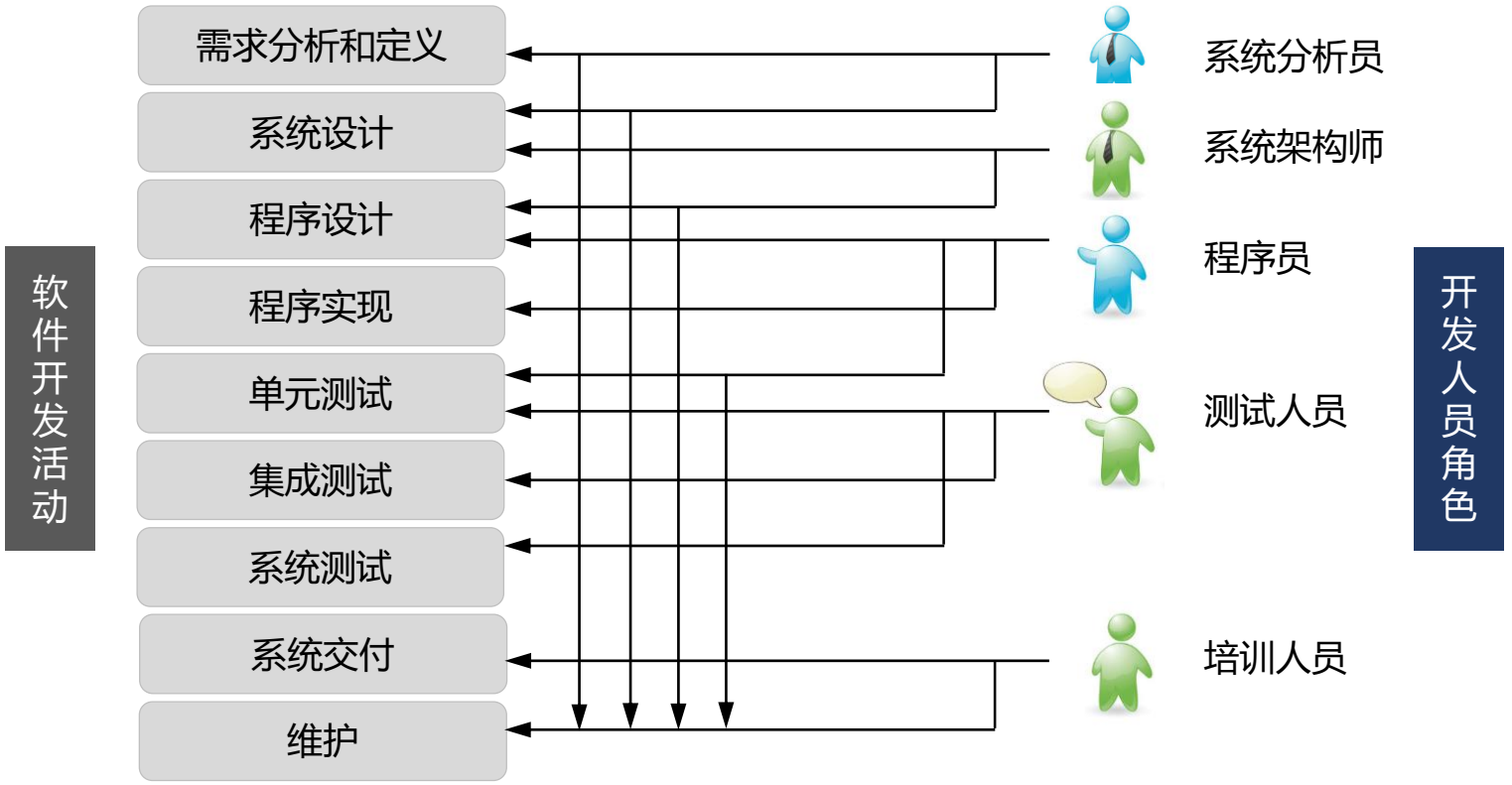
1. 项目人员和团队组织
2. 项目沟通管理
3. 软件项目估算
4. 项目计划与进展跟踪
5. 软件配置管理

教学提纲

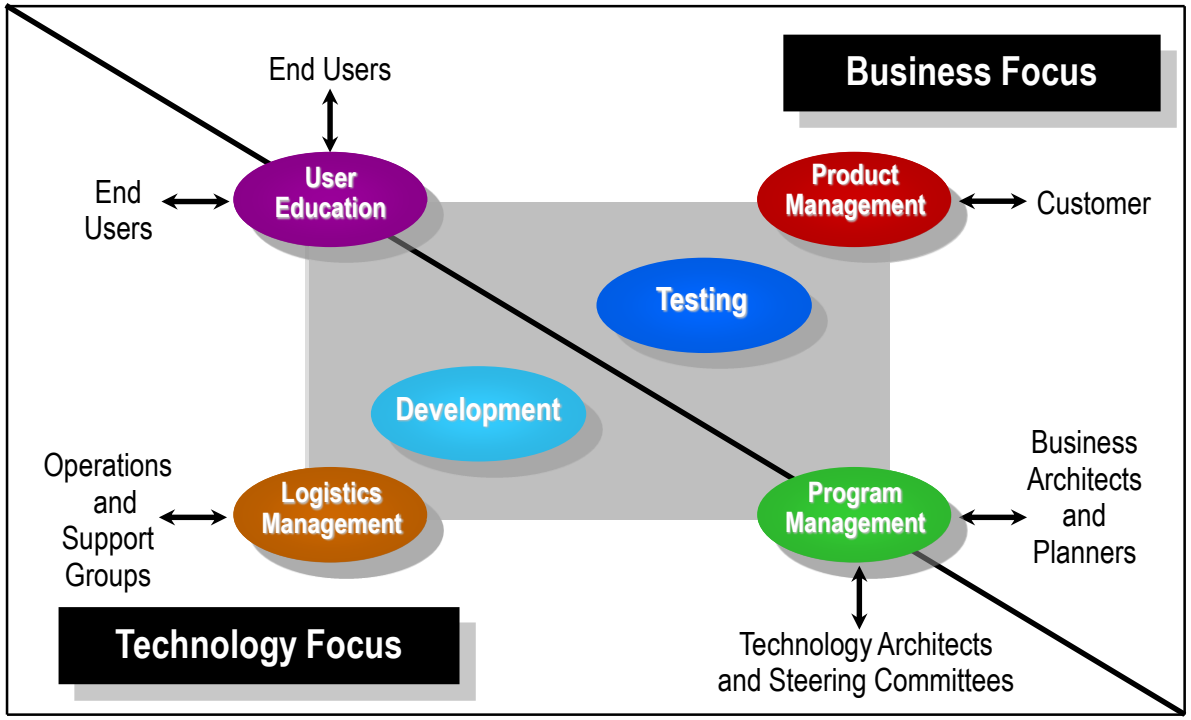
1 项目人员和团队组织

- 开发团队的角色分工
- 人员的选择
- 团队组织结构
- 团队建设与管理

开发团队的角色分工


软件开发活动
需求分析和定义
系统设计
程序设计
程序实现
单元测试
集成测试
系统测试
系统交付
维护
开发人员角色
系统分析员
系统架构师
程序员
测试人员
培训人员

微软开发团队的角色分工



人员的选择

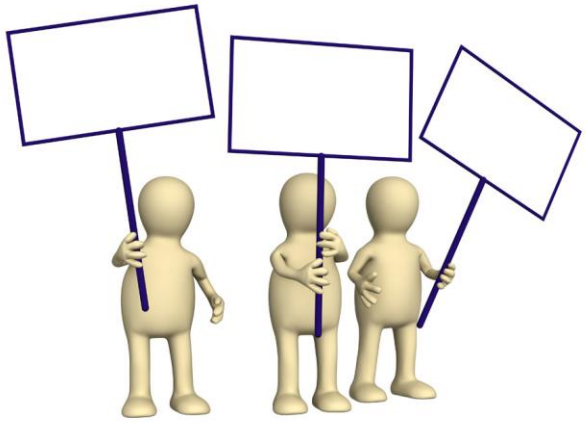
案例描述

假设你是一个报警系统开发公司的项目经理，该公司希望进入帮助独立生活的老年人和残疾人的技术服务市场。你需要组建一个6人的开发团队，他们可以基于公司现有的报警处理技术开发新产品。

显然，你的首要任务是从公司内部或者外部选择合适的团队成员。

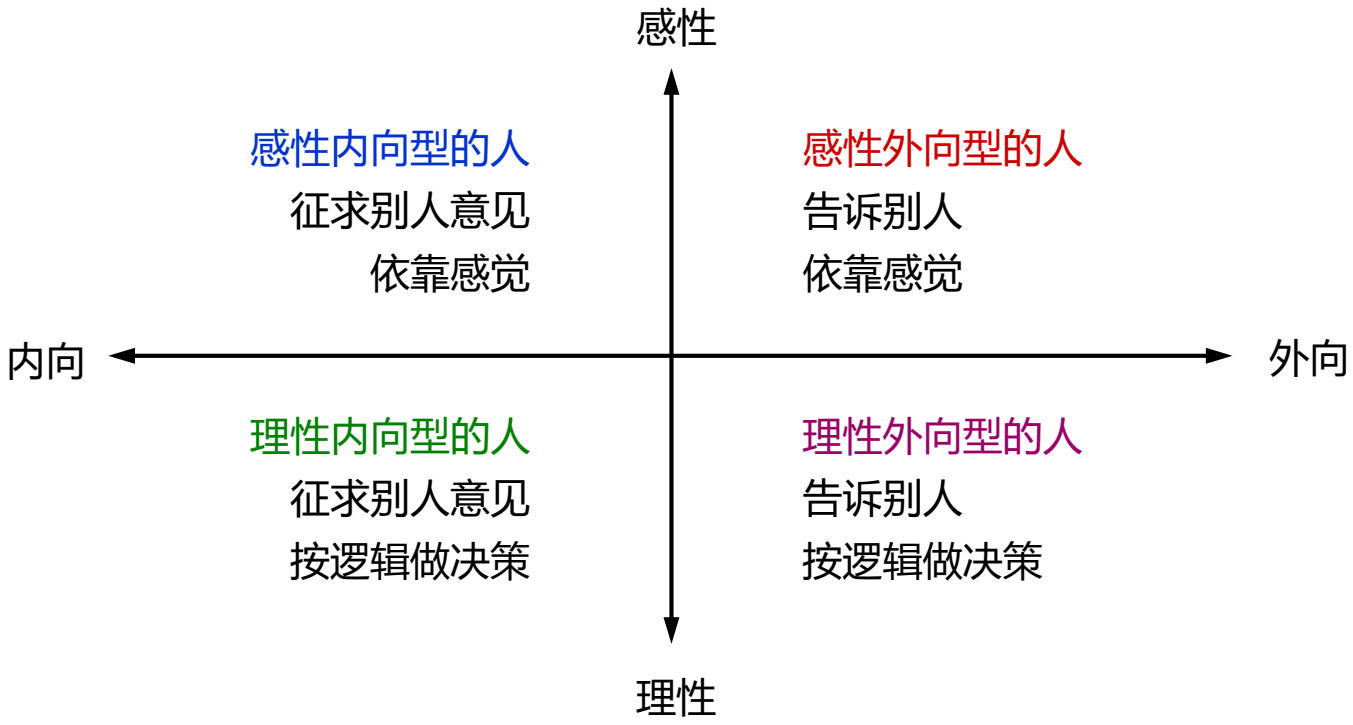
你如何选择团队成员？这样选择的理由是什么？

人员的选择



- 应用领域经验
- 平台经验
- 编程语言经验
- 教育背景
- 解决问题能力
- 沟通能力
- 适应性
- 工作态度
- 个性

人员的选择


感性
感性内向型的人
征求别人意见
依靠感觉
感性外向型的人
告诉别人
依靠感觉
内向
外向
理性内向型的人
征求别人意见
按逻辑做决策
理性外向型的人
告诉别人
按逻辑做决策
理性

人员的选择

如果你负责的项目工作出现进度落后问题，
不同性格的人将会怎样与你交流？

- 理性外向型：告诉你什么时候必须完成工作，将会为你安排一个新的进度。
- 感性外向型：告诉你什么时候必须完成工作，可能会提供一些建议让工作步入正轨。
- 理性内向型：询问你什么时候可以完成，在分析自己意见时希望知道落后的原因。
- 感性内向型：询问你什么时候可以完成，并问你能够帮你做些什么。

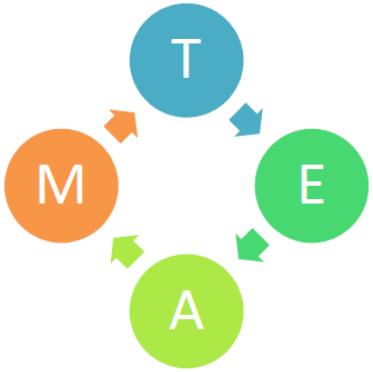
人员的选择

- 应该考虑团队中的技术、经验和个性是否整体均衡 。
- 选择性格互补的成员组成的团队可能比仅仅根据技术能力选择成员的团队更有效率。
- 团队的领导力来自于成员的尊重，而不是名义上的头衔。

团队的概念

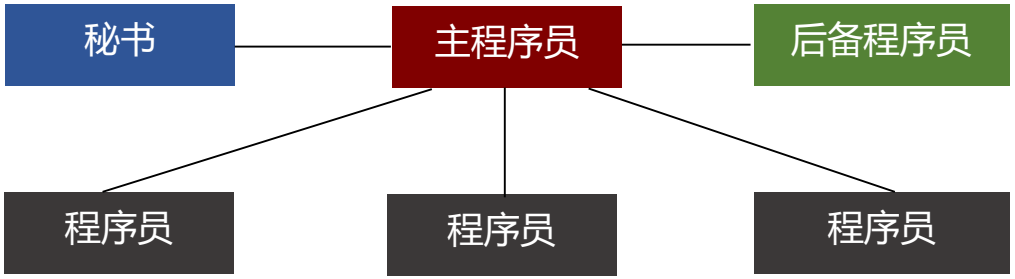
团队：由少量的人组织，具有互补的技能，对一个共同目的、绩效目标及方法做出承诺并彼此负责。

T	together	结合	众
E	everyone	大家	志
A	achieve	完成	成
M	Mission	任务	城



软件开发团队组织

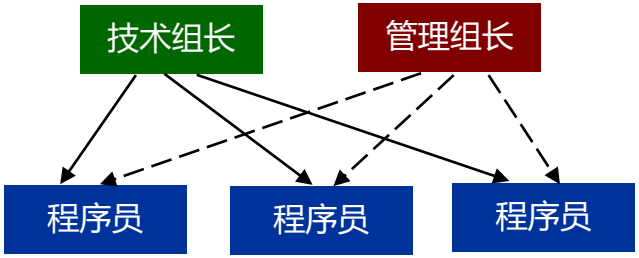
主程序员式组织结构：以主程序员为核心，主程序员既是项目管理者也是技术负责人，团队其他人员的职能进行专业化分工。



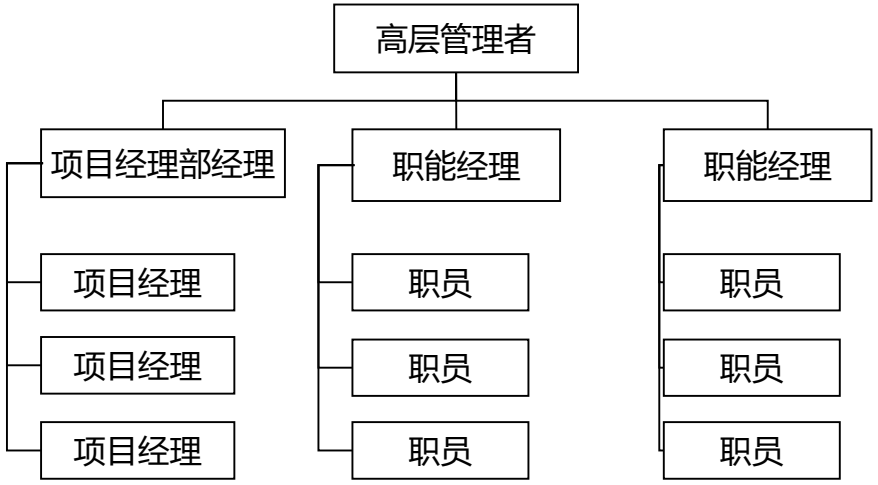
- 优点：成员之间采取简单的交流沟通模式
- 缺点：很难找到技术和管理才能兼备的主程序员

软件开发团队组织

矩阵式组织结构：将技术与管理工作进行分离，技术负责人负责技术决策，管理负责人负责非技术性事务的管理决策和绩效评价。



在这种组织中，明确划分技术负责人和管理负责人的权限是十分重要的。



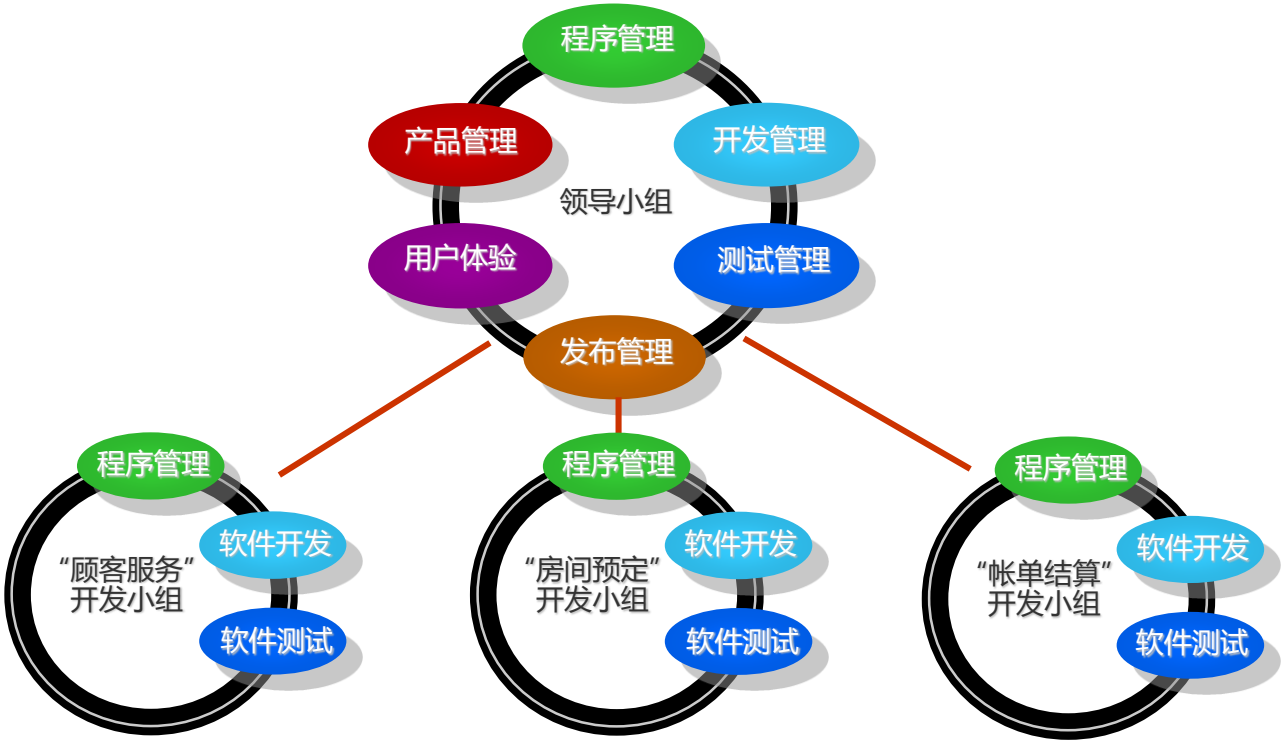
微软开发团队组织

微软开发团队的特点：

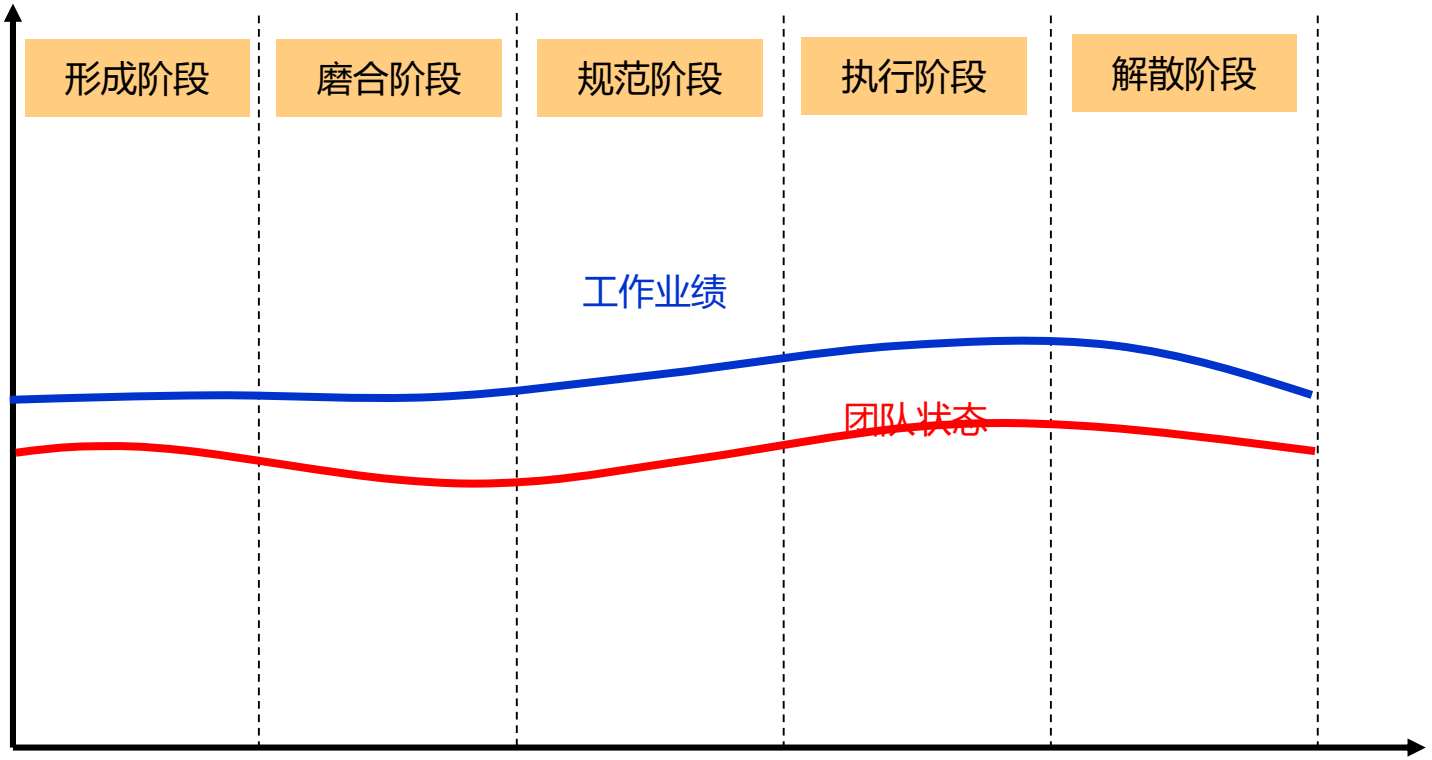
- 小型的、多元化的项目组织
- 相互依赖的角色与共同分享的职责
- 具备专深的技术水平和业务技能
- 具有强烈的产品意识，关注最终发布的软件产品
- 清晰的目标和远景
- 人人参与设计
- 项目组成员在同一地点办公
- 对于规模较大的项目，采取类似小型项目组的运作模式

Microsoft®

微软开发团队组织

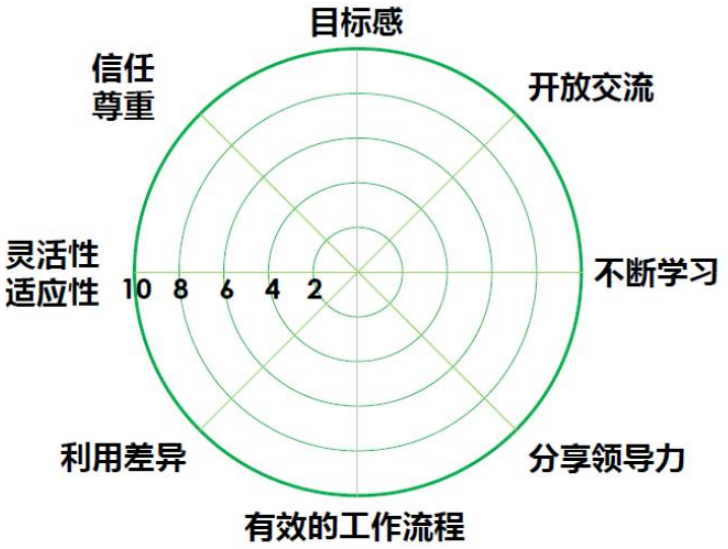

程序管理
产品管理
开发管理
领导小组
用户体验
测试管理
发布管理
程序管理
软件开发
“顾客服务”
开发小组
软件测试
程序管理
软件开发
“房间预定”
开发小组
软件测试
程序管理
软件开发
“帐单结算”
开发小组
软件测试

团队发展生命周期



高效项目团队的特征

- 营造一种支持性的人力资源环境（沟通，交流，团队学习）
- 团队成员的自豪感
- 让每一位成员的才能与角色相匹配
- 设定具有挑战性的团队目标
- 正确的绩效评估



团队建设活动

教学提纲

2 项目沟通管理

- 沟通的基本概念
- 沟通方式与技巧
- 项目沟通活动
- 演讲与表达的艺术

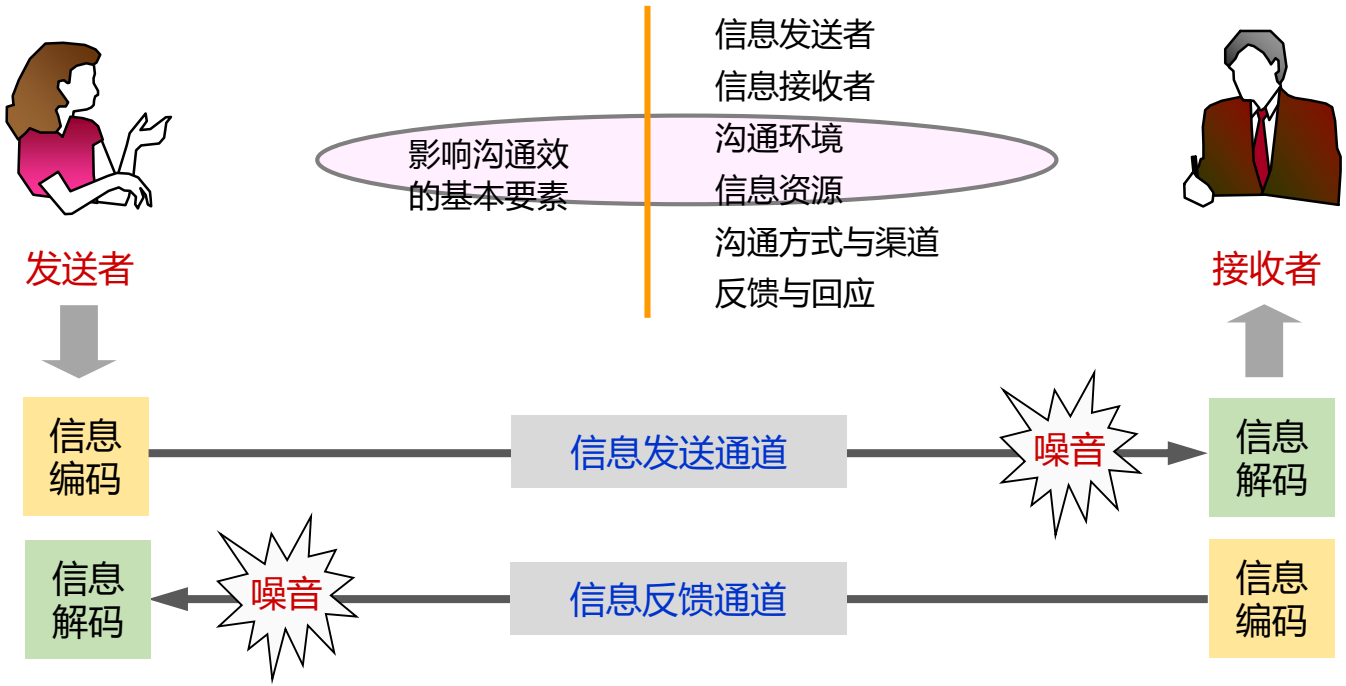
沟通的概念

沟通是为了达到一定的目的，将信息、思想、情感在个人或群体之间进行传递或交流的过程。

- 沟通就是相互理解
- 沟通就是提出和回应问题与要求
- 沟通交换的是信息和思想
- 沟通是一种有意识的行为

“沟通是你被理解了什么而不是说了什么”

沟通的模型


信息发送者
信息接收者
沟通环境
信息资源
沟通方式与渠道
反馈与回应
影响沟通效的基本要素
发送者
接收者
信息编码
信息发送通道
噪音
信息解码
信息解码
噪音
信息反馈通道
信息编码

沟通的复杂性

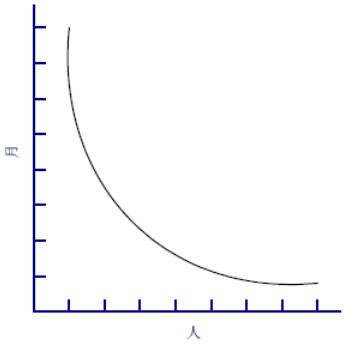
软件开发是一项错综复杂关系的工作，**沟通与交流**的工作量极大。

- 举例一：1个农夫可在10天内采摘完一块草莓地，那么同样的草莓地是否可以用10个农夫可在1天内采摘完？
- 举例二：1头大象需要孕育22个月才能生下1头小象，那么增加大象的数量是否可以加快这个过程呢?
- 举例三：假设开发某个模块需要2人月的工作量，那么是否可以认为2个程序员在1个月内就可以完成这个模块的开发？

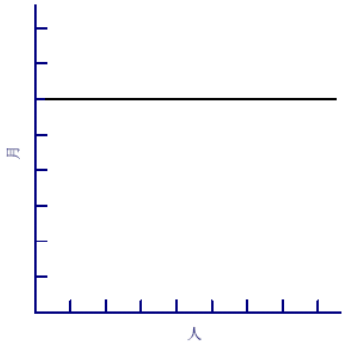
沟通的复杂性

人员数量与项目时间的关系：

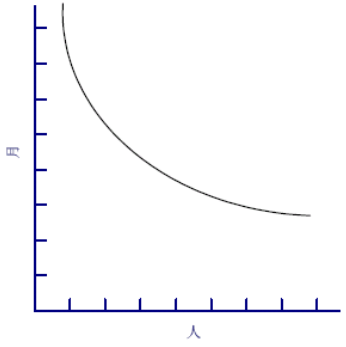
- 有些任务可以共同分担，有些任务则不行
- 沟通花费大量的时间



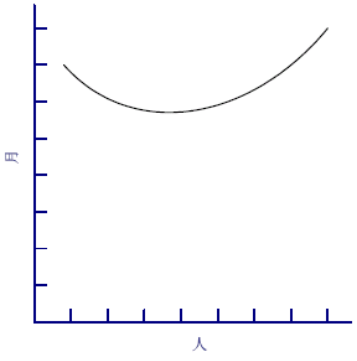
完全可分解的任务



无法分解的任务



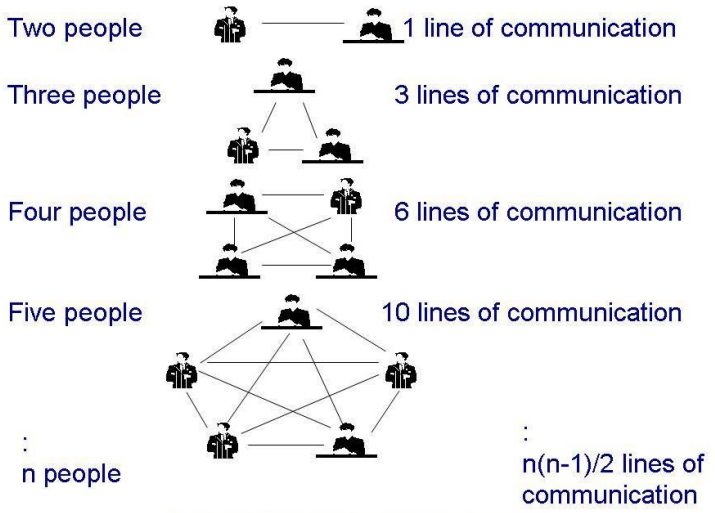
需沟通的可分解任务

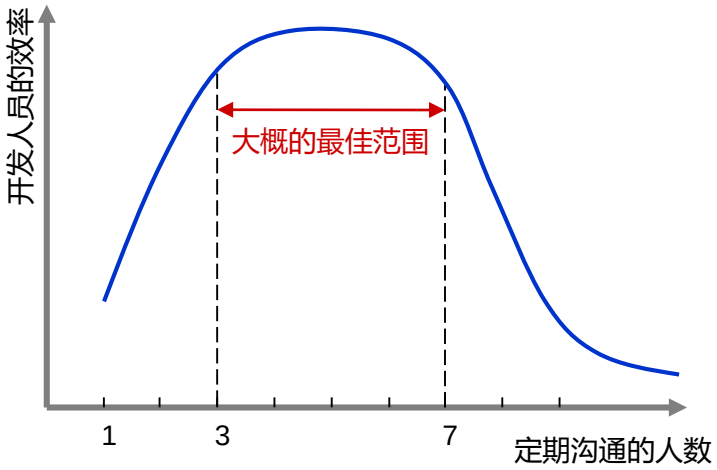


关系错综复杂的任务

沟通的复杂性

Brooks法则：向一个进度延迟的软件项目中增加人员可能会使其进度更加推迟。





常用沟通方式

口头	• 正式的：演讲、报告、汇报、谈判、会议 • 非正式：谈话、电话、打招呼
书面	• 正式的：合同、报告、会议纪要、报表、备忘录 • 非正式：笔记、便条
非语言	• 正式的：手语、信号灯、音乐 • 非正式：表情、声调、拥抱、握手
工具	• 电话、传真、手机、Email、面对面、协作工具

思考讨论

你认为以下情形应该选择什么合适的方式?

- 公司主管副总经理通知你，说他想详细了解项目进展情况，并且想在从今天开始的一周后与你见面。
- 客户在检查项目阶段成果时，指出上个月曾经提出的某个产品特性没有包含在其中，并抱怨早就以口头方式反映给了项目组成员，但作为项目经理的你却一无所知，而那位成员解释说把这件事忘记了。
- 你负责的一个项目准备在月底召开项目工作汇报会议，需要通知所有与会代表。
- 你作为硕士研究生，即将开始论文工作，你需要经常与导师进行联系和交流。

沟通技巧

自然赋予我们人类一张嘴、两只耳朵，也就是让我们多听少说。

—— 苏格拉底

- 努力做一个百分之百的听众
- 善于抓住关键点
- 努力排除干扰
- 培养分清主次的能力
- 通过提问和行为表示赞同与鼓励
- 对所听的内容进行及时反馈

项目沟通活动

项目组内的沟通：

•项目组成员的四个主要沟通需求：职责、协调、状态、授权
•任务分配清晰
•会议：项目启动会、成员进度汇报、项目进展会
•设置沟通期望
•及时、公开、恰到好处

管理层和客户之间的沟通：

•谁、为什么需要信息?
•需要什么类型的信息？何种详细程度？频率如何?
•当你与管理层或客户沟通时，你的目标是什么？采用什么样的方法来沟通?

项目会议

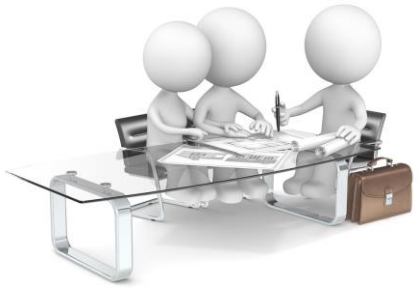
第一次项目团队会议（至关重要）

- 目标
  - ✧ 项目概况：范围与目标、总体进度、方法和程序
  - ✧ 确定项目人员的角色和任务
  - ✧ 确立团队的工作模式
- 形式
  - ✧ 重大项目：精心准备、集中1-2 天；前期介绍与建立基本规则
  - ✧ 一般项目：简单有效；回顾项目范围与成员互相自我介绍
- 建立基本规则
  - ✧ 计划决策、追踪决策、管理变动决策、关系决策

项目会议

项目阶段进展会议（每月一次）

- 向项目干系人和高层管理者汇报项目进展
- 解决需要高层管理者支持的问题



项目组周例会（每周一次）

- 明确短期目标，制定具体工作计划
- 协调资源需求
- 解决项目组工作中发生的任何问题

项目组内部会议（每天一次）

- 通报项目组成员的工作进展
- 确定项目组需要向上汇报并解决的问题

召开有效的会议

三星公司开会的三个原则：

1.周三不开会；
2.会议时长1小时，最多不超过1.5小时；
3.将会议的内容整理成一张纸。

三星公司开会的七项规定：

1.严格遵守时间；
2.在会议材料中写明投入会议的经费；
3.参加者限制为必要的合适人选和负责人，会议规模尽可能小；
4.明确会议目的；
5.事先分发会议资料；
6.要让所有参加者发言；
7.尽可能地减少会议记录，仅记录决定了的事项并进行保存。

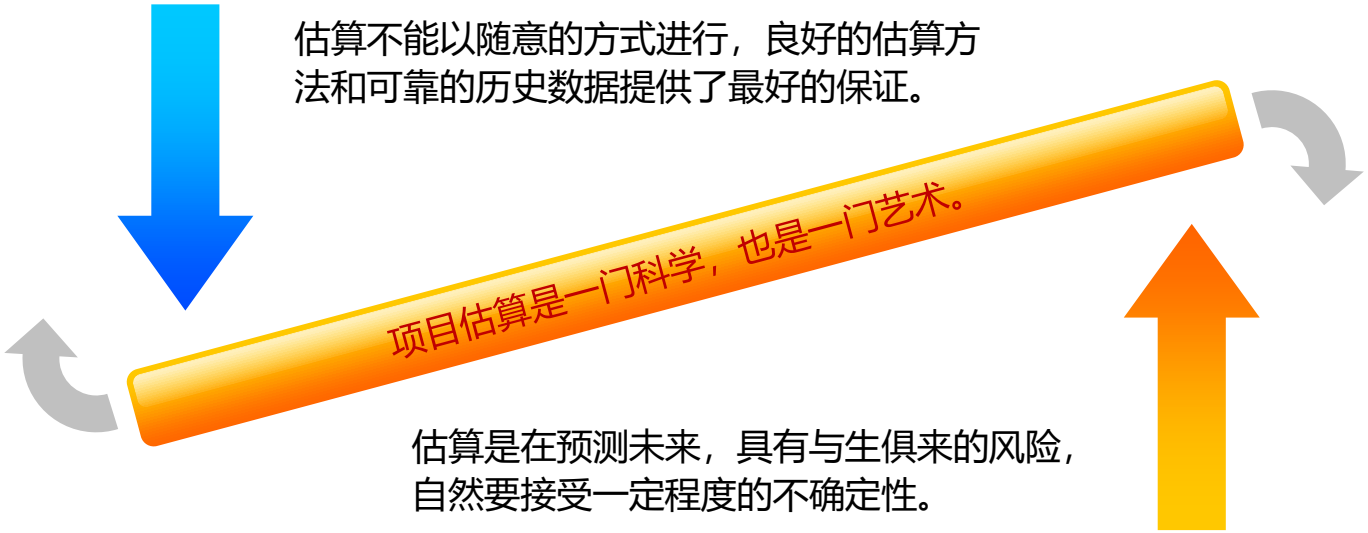
教学提纲

3 软件项目估算

- 项目估算概述
- 功能点与COCOMO模型
- 用例估算与故事点方法
- 机器学习方法

项目估算

项目估算是对项目的规模、工作量、时间和成本等进行预算和估计的过程。



项目估算的挑战

为什么软件开发的周期
总是预估的2~3倍？

People Tend To Overestimate What Can Be Done In One Year And To Underestimate What Can Be Done In Five Or Ten Years

项目估算的挑战

没有很好地制定计划是软件项目常见的一个严重错误，很多技术人员宁愿从事技术工作，也不愿意花费时间制定计划。

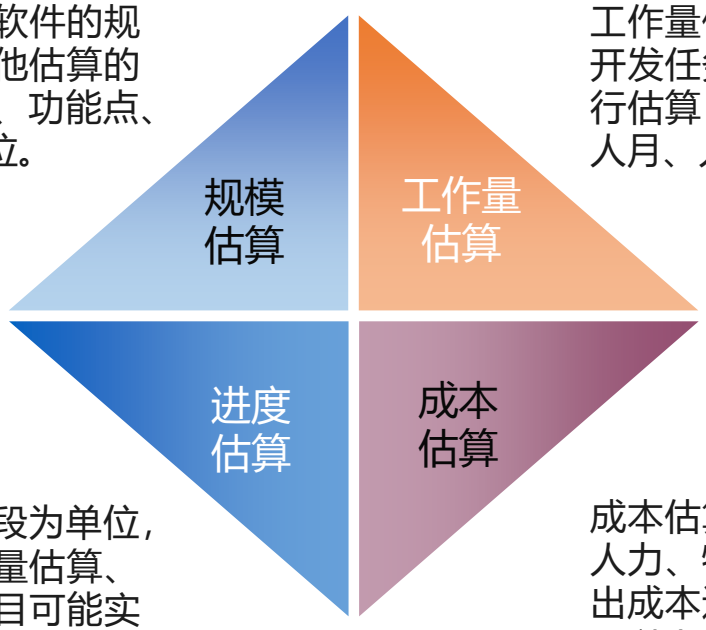
软件项目估算的挑战是项目的复杂性和不确定性

- 软件规模越大，复杂性越高，不确定性就越大
- 需求的不确定性会对项目估算产生很大影响
- 没有可靠的历史数据使项目估算缺少参照物

项目管理者不应该被估算所困扰，应该克服困难，做出一个相对的有价值的估算。

项目估算内容

规模估算是对所开发软件的规模进行估计，它是其他估算的基础，通常以代码行、功能点、故事点等作为衡量单位。



工作量估算是结合投入人力和开发任务所需要的工作时间进行估算，通常以人时、人天、人月、人年等作为衡量单位。

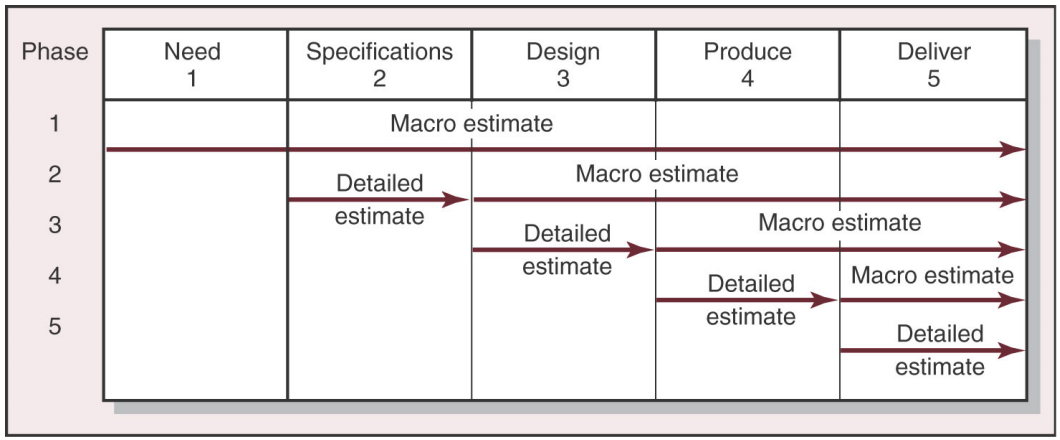
进度估算是以项目阶段为单位，通过任务分解、工作量估算、有效资源分配等对项目可能实施的进度给出正确的评估。

成本估算是对包括项目投入的人力、物质、有形和无形的支出成本进行估算，一般软件项目以人力成本为主要部分。

软件项目估算

分段估算：

- 从宏观估算开始，在项目执行中对各阶段的估算进行细化
- 适用于最终产品不可知或不确定性很大的项目



基本估算方法

专家判断：

- 通过借鉴历史信息，专家提供项目估算所需的信息，或根据以往类似项目的经验，给出相关参数的估算上限。

类比估算：

- 以过去类似项目的参数值（如持续时间、预算、规模、工作量和复杂性等）为基础，估算未来项目的同类参数或指标。
- 类比估算综合利用项目历史信息和专家判断，它是一种粗略的估算方法，有时需要根据项目复杂性方面的已知差异进行调整。
- 类比估算通常成本低、耗时少，但准确性较差，可以与其他估算方法联合使用。

基本估算方法

参数估算：通过对大量的项目历史数据进行统计分析，使用项目特性参数建立经验估算模型，估算诸如成本、预算和持续时间等活动参数。

功能点方法

功能点方法是依据软件信息域的基本特征和对软件复杂性的估计，估算出软件规模。这种方法适合于在软件开发初期进行估算，并以功能点为单位度量软件规模。

功能点估算使用5种信息域：

- 外部输入：通过界面等的输入，插入和更新等操作都是典型外部输入
- 外部输出：仅仅是输出，例如导出、报表、打印等
- 外部查询：先输入数据，再根据输入数据计算得到输出，例如查询
- 内部逻辑文件：可以理解为业务对象，可能对应多个数据表
- 外部接口文件：其它应用提供的接口数据

功能点方法

信息域加权因子：

信息域参数	加权因子			合计
	简单	中等	复杂	
外部输入	3	4	6	∑
外部输出	4	5	7	∑
外部查询	3	4	6	∑
内部逻辑文件	7	10	15	∑
外部逻辑文件	5	7	10	∑
未调整功能点 UFC				∑

系统复杂度调整值 Fi：取值 0..5

F_1	可靠的备份和恢复	F_8	在线升级
F_2	数据通信	F_9	复杂的界面
F_3	分布式处理	F_{10}	复杂的数据处理
F_4	性能	F_{11}	代码复用性
F_5	大量使用的配置	F_{12}	安装简易性
F_6	联机数据输入	F_{13}	多重站点
F_7	操作简单性	F_{14}	易于修改

功能点计算： $FP = \underline{UFC} \times [0.65 + 0.01 \times \sum F_i]$

COCOMO模型

结构性成本模型 COCOMO（COnstructive COst MOdel）是一种利用经验模型进行成本估算的方法。

$$PM_{nominal} = A * (Size)^{B}$$

- PMnominal：人月工作量
- A：工作量调整因子
- B：规模调整因子
- Size：规模，单位是千行代码或功能点数

类型	A	B	说明
组织型	2.4	1.05	相对小的团队在一个高度熟悉的内部环境中开发规模较小，接口需求较灵活的系统。
嵌入型	3.6	1.2	开发的产品在高度约束的条件下进行，对系统改变的成本很高。
半独立型	3.0	1.12	介于上述两者中间

用例估算

	基本流	扩展流	业务规则	折合标准用例
功能A	12	3	4	2.3
功能B	8	4	3	1.8
功能C	6	2	3	1.4
			合计	5.5

标准用例 = （基本流 + 扩展流 + 2×业务规则） / 10

生产率 = 6工作日 / 单位用例

工作量 = 6×5.5 = 33工作日

故事点方法

故事点：

- 故事点是用于表达用户故事、功能或其他工作的总体规模的度量单位，它是一个相对度量单位。
- 使用时可以给每个故事分配一个点值，点值本身并不重要，重要的是点值的相对大小。

理想日：

- 理想日是用于表达用户故事、功能或其他工作的总体规模的另外一种度量单位，它是一个绝对度量单位。
- 理想时间是某件事在剔除所有外围活动以后所需的时间；一般为一天有效工作时间的 60-80% 比较合理，但绝不会是全部。

故事点方法

故事点的基本做法：把一些常见“标准任务”给出一个“标准点数”，形成比较基线；估算时只要是同一类型任务，直接写故事点数而非天数。

举例：自动售货机

用户故事	说明
购买饮料	用户投钱并购买指定饮料
取消购买	用户投钱之后取消购买
输入管理密码	授权人输入管理密码，以便进行补货、定价、取钱等操作
补充饮料	在输入管理密码后，授权人补充饮料
设定价格	在输入管理密码后，授权人可以重新设定饮料价格
取出钱款	在输入管理密码后，授权人可以取出钱箱中的钱
打印月报	在输入管理密码后，授权人打印月销售报表
发出报警	在异常情况发生时，系统自动打开安全警报

故事点方法

故事点的基本做法：把一些常见“标准任务”给出一个“标准点数”，形成比较基线；估算时只要是同一类型任务，直接写故事点数而非天数。

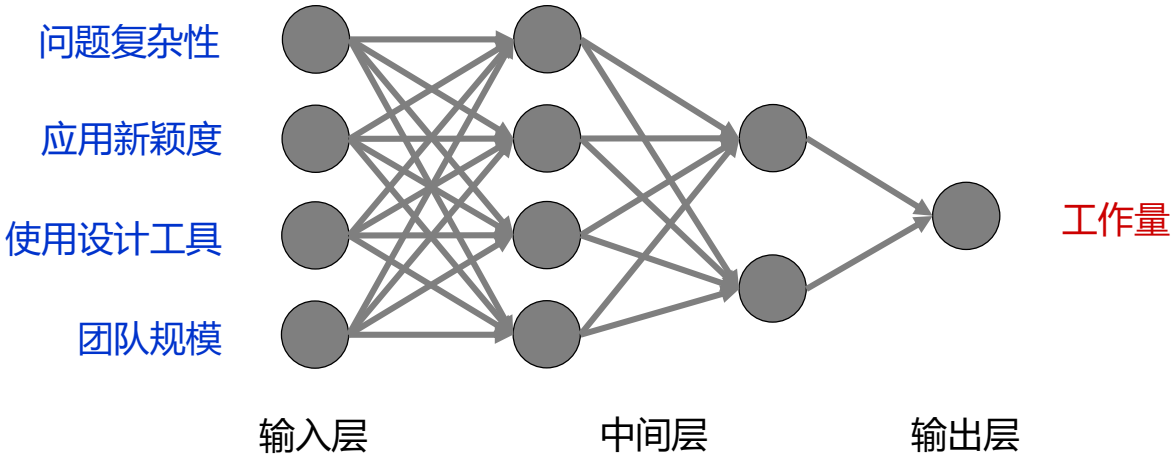
举例：自动售货机

用户故事	故事点
购买饮料	4
取消购买	2
输入管理密码	1
补充饮料	3
设定价格	2
取出钱款	1
打印月报	4
发出报警	2

19

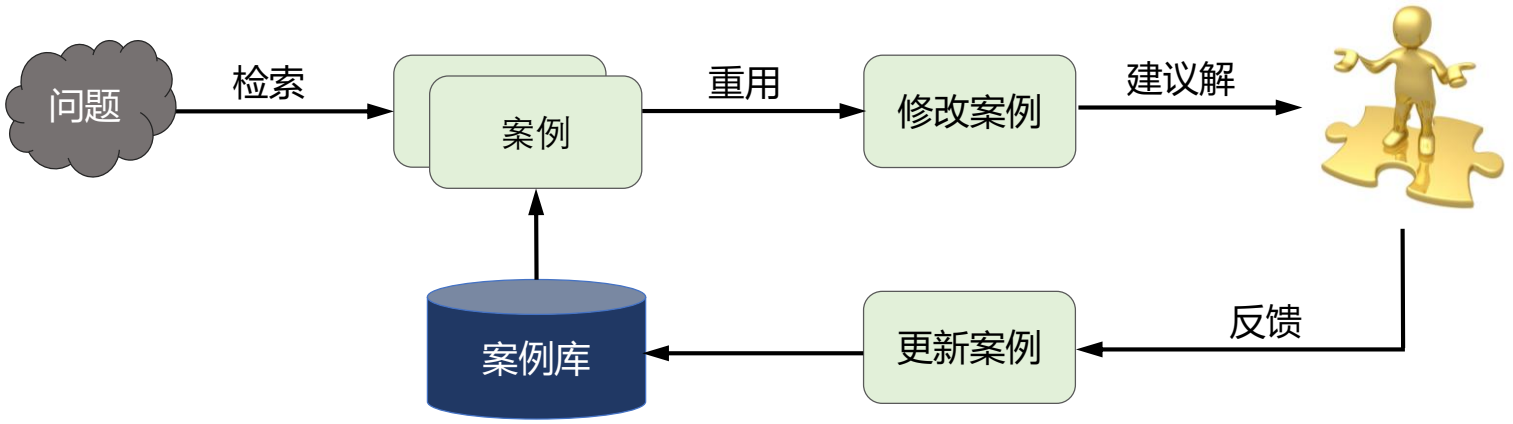
机器学习方法

人工神经网络是采用一种学习方法导出一种预测模型，首先建立神经网络，再使用一组历史项目数据（样本数据）训练网络，训练后的网络可以用于估算新项目的工作量。



机器学习方法

基于案例的推理方法可以用于基于类推的估算，即识别出与新项目类似的案例，再调整这些案例，使其适合新项目的参数。



教学提纲

4 项目计划与进展跟踪

- 项目管理活动
- 软件项目计划
- 项目风险管理
- 项目进展跟踪

软件项目管理计划

软件项目管理计划是一个用来协调所有其他计划、以指导项目实施和控制的文件，它应该随着项目的进展和信息的补充进行定期完善。

引言	项目的目标、影响项目管理的各种约束条件
项目组织	开发团队的组织方式、人员构成与分工
风险分析	可能的风险以及发生的可能性、降低风险的策略
资源需求	项目所需的硬件与软件资源
工作分解	将项目分解成一系列的活动，指定项目里程碑和可交付的文档
项目进度	项目中各活动之间的依赖关系、完成每个里程碑预期需要的时间、在活动中的人员分配
监控和报告机制	需要提交的管理报告、提交时间以及项目监控机制

制定软件项目计划

软件项目计划流程

开发问题描述 → 定义顶层设计 → 定义工作分解结构 → 建立初始时间表

开发问题描述

问题描述

问题描述是描述系统应该说明的问题、目标环境、客户交付和验收标准的简短文档。

问题描述是对系统所表述问题的共同认识，通常是由项目团队和客户共同开发形成的，它定义了问题提出的背景、需要支持的功能和性能以及系统运行的目标环境等。

在线编程协作平台V2.0

微信 | 公众平台

自定账号名

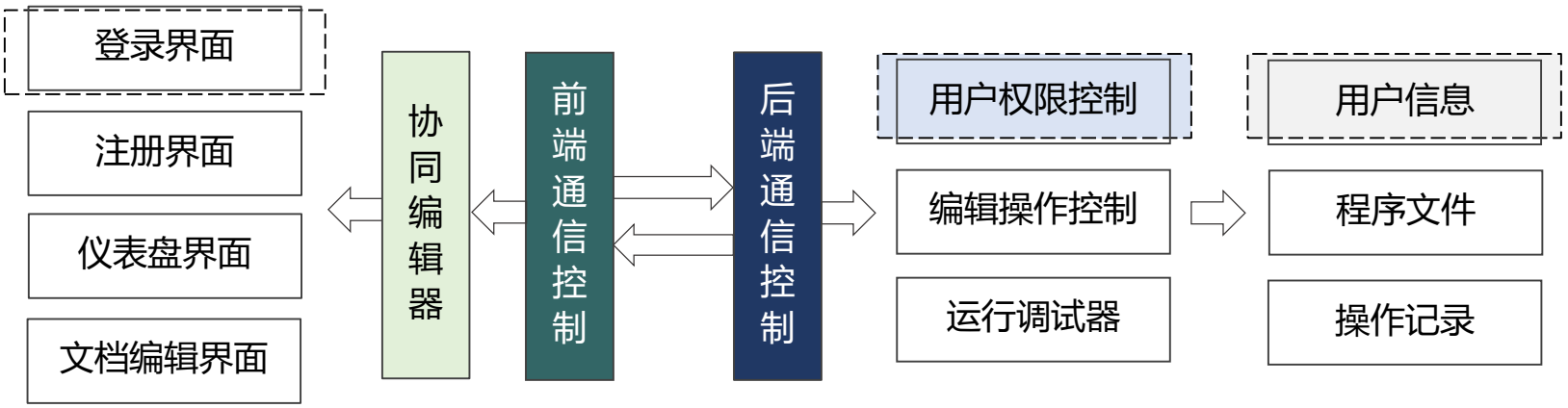
定义顶层设计

顶层设计描述了最初从系统到子系统的分解，它描述了系统的软件体系结构。软件架构师在需求分析的基础上，根据项目的设计目标，将系统分解为更小的若干子系统。

明确设计目标	应考虑系统设计质量，例如性能、可扩展性等。
初始子系统分解	比较不同的设计方案，可采用标准的体系结构风格，需要考虑系统构造策略，诸如软硬件环境、数据存储、系统控制、访问控制等。
不断分解和求精	初始分解也许不能满足所有设计目标，需要进一步分解和求精。
任务组织和分配	不同的子系统分配给不同的团队或开发人员完成，由他们协商定义子系统的服务及其接口。

定义顶层设计

举例：在线编程协作平台的顶层设计



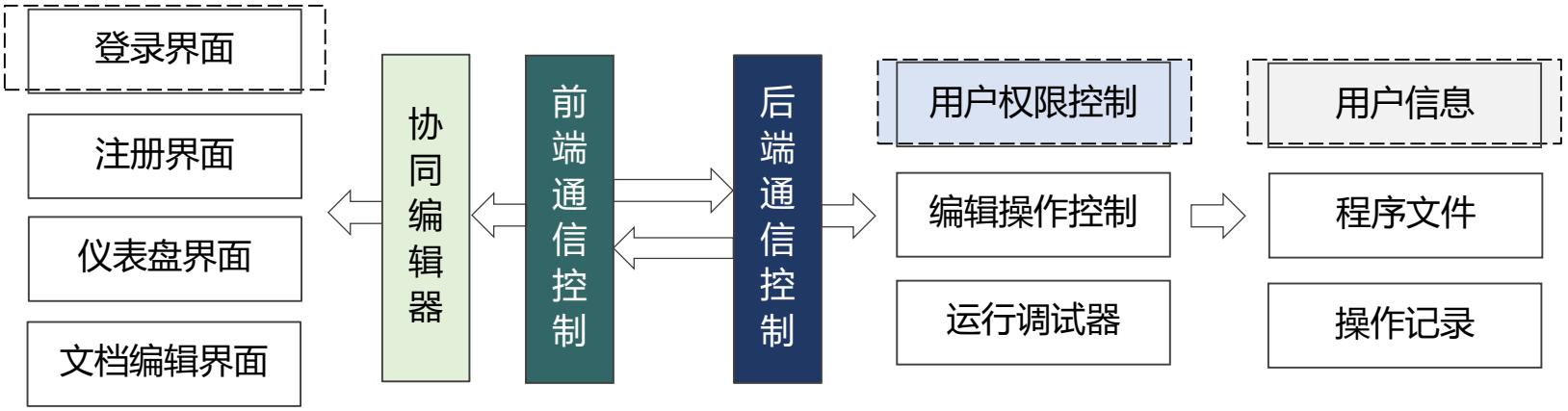
定义项目工作分解

项目工作分解是将项目整体分解成较小的、易于管理和控制的若干子项目或工作单元，直到可交付成果定义的足够详细，足以支持项目将来的活动。

假设一个开发团队由5人组成，某团队基于系统功能将项目工作进行分解，将注册、登录、编辑、运行、聊天等不同功能的开发分配给不同的成员，你如何评价这种分解方案？

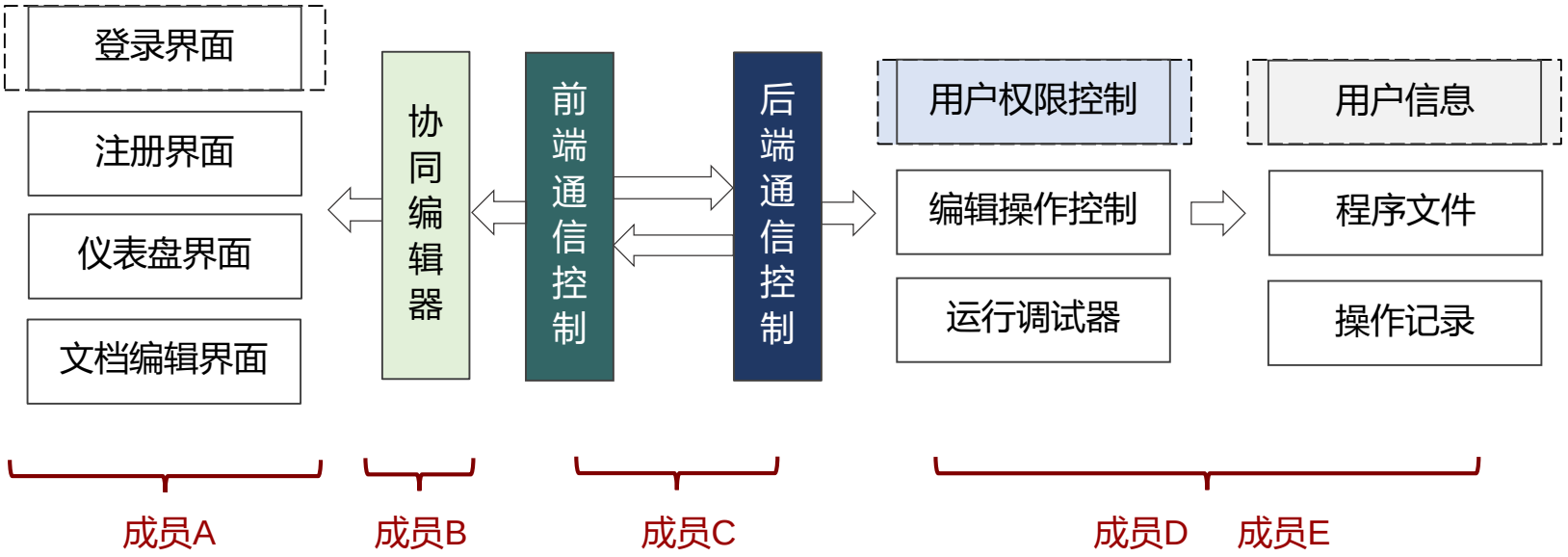
定义项目工作分解

思考：如果根据子系统的分解来定义项目工作，你认为应该怎样分解任务？



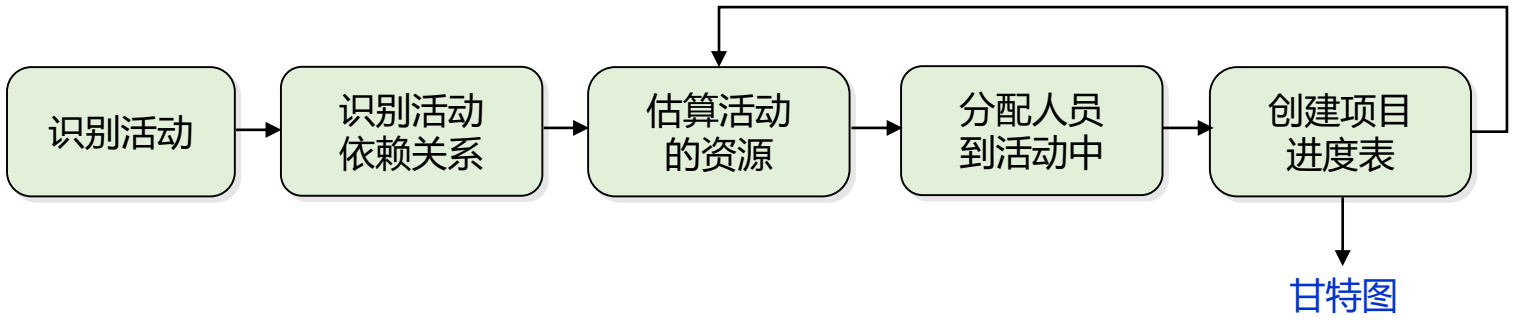
定义项目工作分解

思考：如果根据子系统的分解来定义项目工作，你认为应该怎样分解任务？



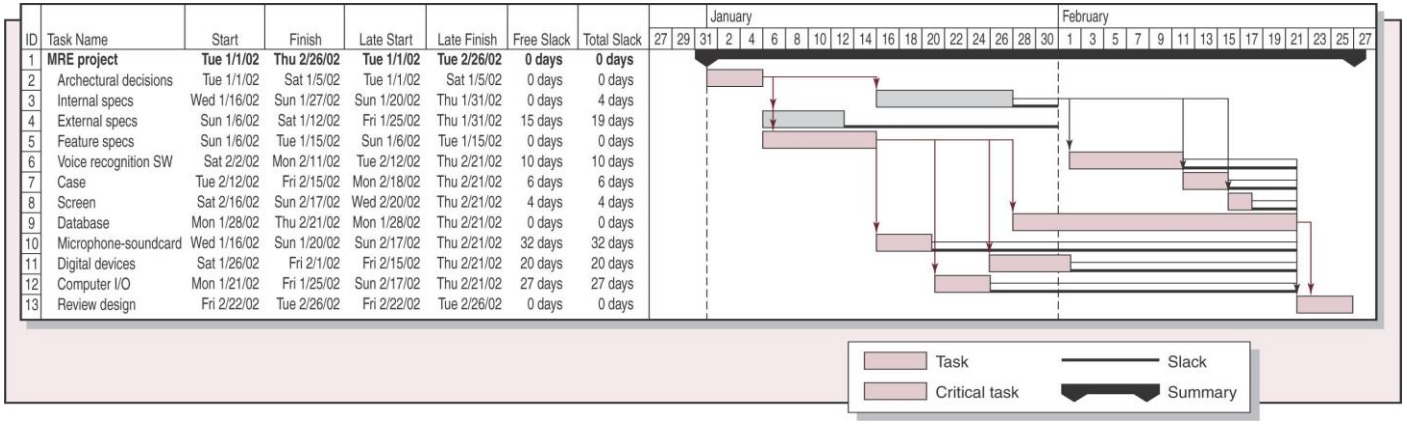
建立初始时间表

在项目工作分解的基础上，进一步估算活动所需的时间和资源，并按照一定的顺序将这些活动进行组织和调度，从而创建项目的进度计划表。



建立初始时间表

甘特图（Gantt Chart）是一种通用的显示进度方法，其横轴表示时间，纵轴表示活动，线条表示在整个期间上计划和实际的活动完成情况。



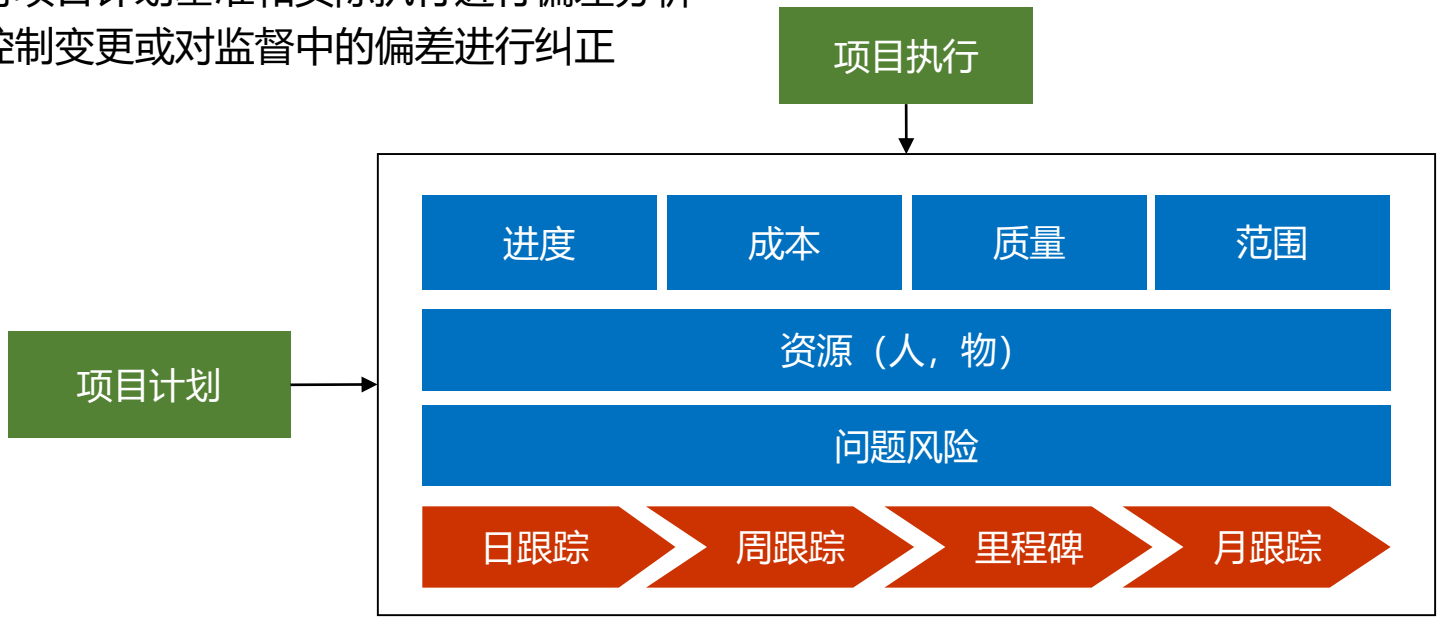
ID	Task Name	Start	Finish	Late Start	Late Finish	Free Slack	Total Slack
1	**MRE project**	**Tue 1/1/02**	**Thu 2/26/02**	**Tue 1/1/02**	**Tue 2/26/02**	**0 days**	**0 days**
2	Archectural decisions	Tue 1/1/02	Sat 1/5/02	Tue 1/1/02	Sat 1/5/02	0 days	0 days
3	Internal specs	Wed 1/16/02	Sun 1/27/02	Sun 1/20/02	Thu 1/31/02	0 days	4 days
4	External specs	Sun 1/6/02	Sat 1/12/02	Fri 1/25/02	Thu 1/31/02	15 days	19 days
5	Feature specs	Sun 1/6/02	Tue 1/15/02	Sun 1/6/02	Tue 1/15/02	0 days	0 days
6	Voice recognition SW	Sat 2/2/02	Mon 2/11/02	Tue 2/12/02	Thu 2/21/02	10 days	10 days
7	Case	Tue 2/12/02	Fri 2/15/02	Mon 2/18/02	Thu 2/21/02	6 days	6 days
8	Screen	Sat 2/16/02	Sun 2/17/02	Wed 2/20/02	Thu 2/21/02	4 days	4 days
9	Database	Mon 1/28/02	Thu 2/21/02	Mon 1/28/02	Thu 2/21/02	0 days	0 days
10	Microphone-soundcard	Wed 1/16/02	Sun 1/20/02	Sun 2/17/02	Thu 2/21/02	32 days	32 days
11	Digital devices	Sat 1/26/02	Fri 2/1/02	Fri 2/15/02	Thu 2/21/02	20 days	20 days
12	Computer I/O	Mon 1/21/02	Fri 1/25/02	Sun 2/17/02	Thu 2/21/02	27 days	27 days
13	Review design	Fri 2/22/02	Tue 2/26/02	Fri 2/22/02	Tue 2/26/02	0 days	0 days

项目风险管理

风险识别	确定项目有哪些风险，包括运用专家判断法、头脑风暴法等分析项目风险产生的各种原因或者影响因素，以确定风险事件及其来源。
风险评估	比较风险的大小，确定风险的性质。通过对各种风险进行定性、定量的分析，包括发生概率、影响严重性等，确定出每种风险的大小和性质。
应对计划	制定风险响应的措施和实施步骤，按照风险的大小和性质，制定相应的措施去应对和响应风险，包括风险接受、风险转移等。
风险控制	监督、检查风险事件的发生情况以及风险措施的落实情况，通过对风险事件及其来源的控制和对风险计划落实情况的监督，确保风险措施有效。

项目进展跟踪

监督：将项目计划基准和实际执行进行偏差分析
控制：控制变更或对监督中的偏差进行纠正



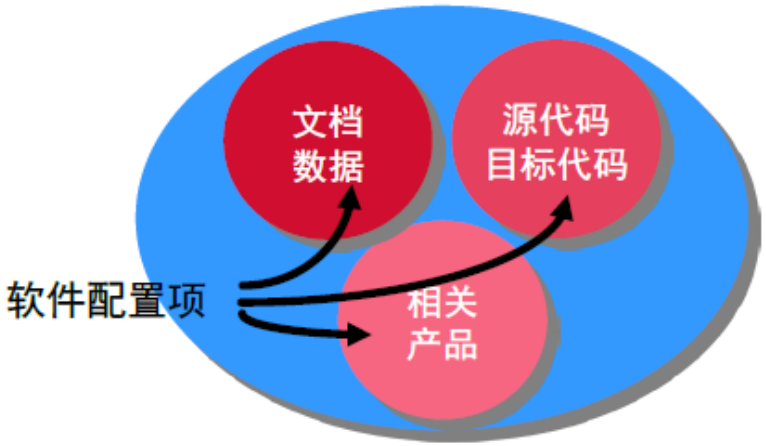
软件配置管理

软件配置管理

- **软件配置管理**是一种标识、组织和控制修改的技术，它作用于整个软件生命周期，其目的是使错误达到最小并最有效地提高生产率。
- 软件配置管理的作用
  - 管理在软件生命周期中建立和修改的各种不同元素
  - 协调和整理所开发的产品
  - 管理软件的构建和测试环境
  - 管理发布和安装工具
  - 管理软件的改错和功能增加

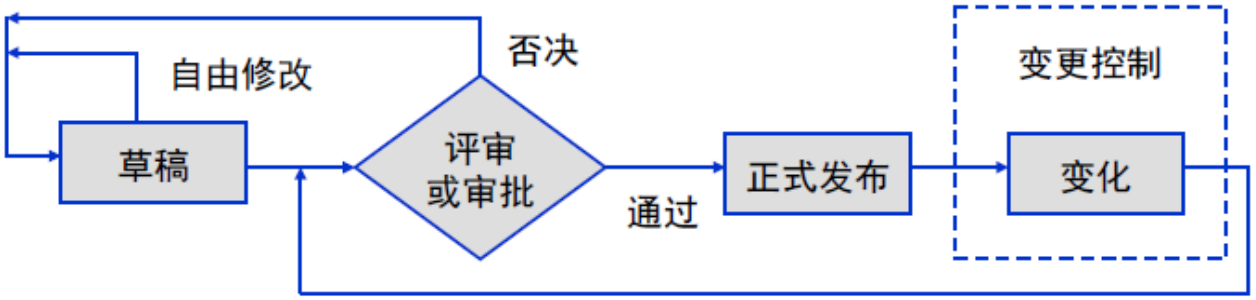
软件配置管理

- **软件配置项**（Software Configuration Item，简称SCI）是为了配置管理而作为单独实体处理的一个工作产品或软件



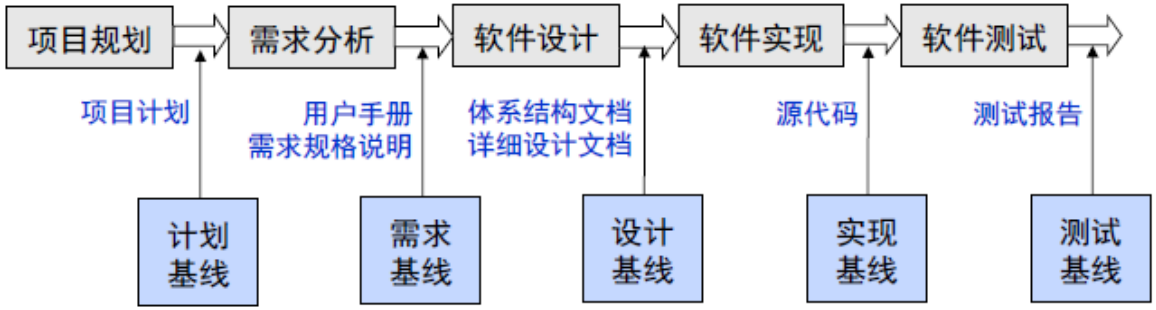
软件配置管理

- SCI属性
  - 名称、标识符、状态、版本、作者、日期等
- SCI状态
  - 草稿(Draft)，正式发布(Released)，变化(Changing)



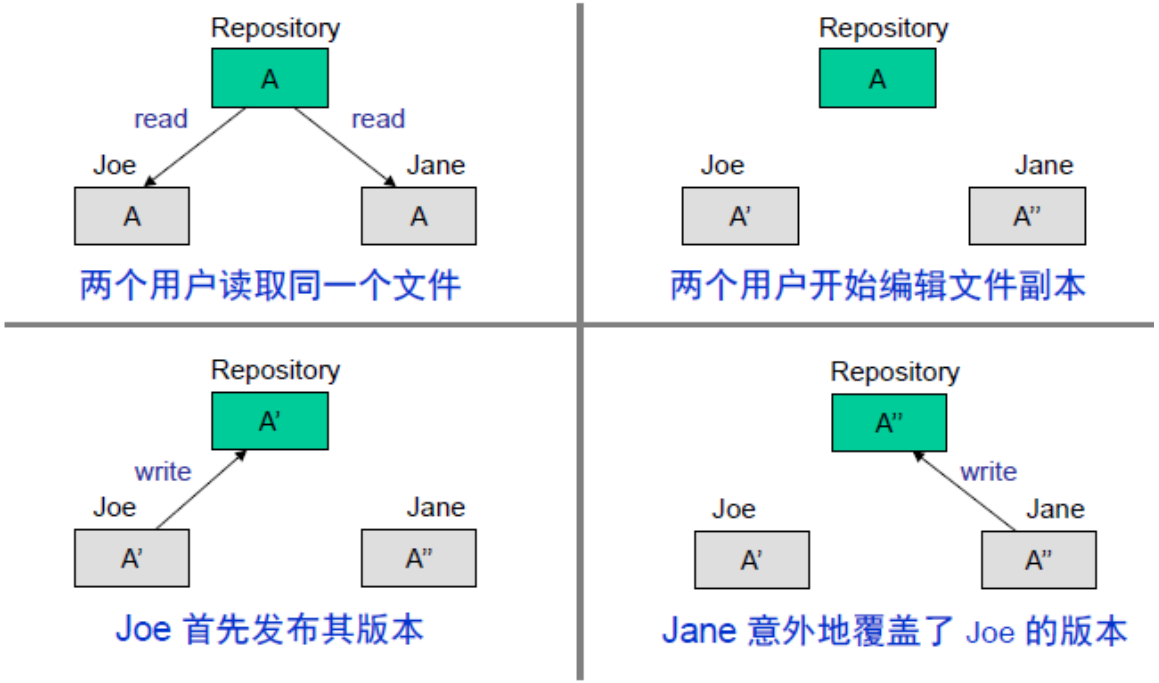
软件配置管理

- **基线（BaseLine）**是已经通过了正式复审的规格说明或中间产品，它可以作为进一步开发的基础，并且只有通过正式的变化控制过程才能改变。
- 基线标志着软件开发过程的各个里程碑（Milestone）。



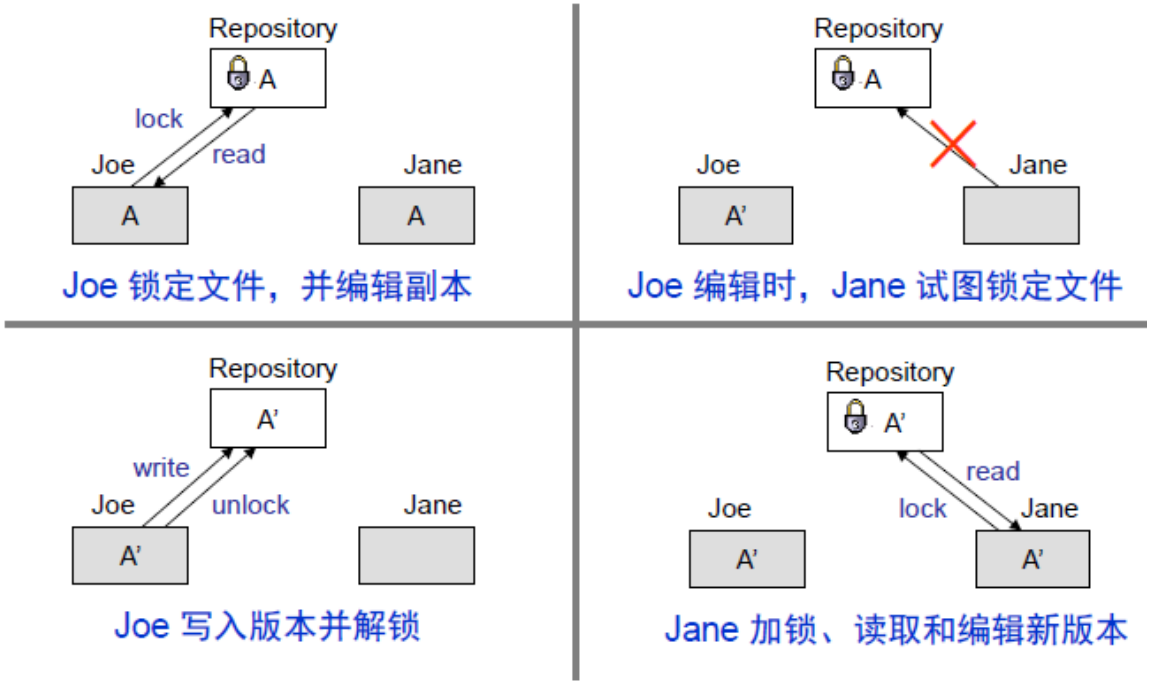
软件配置管理

- 文件共享的问题



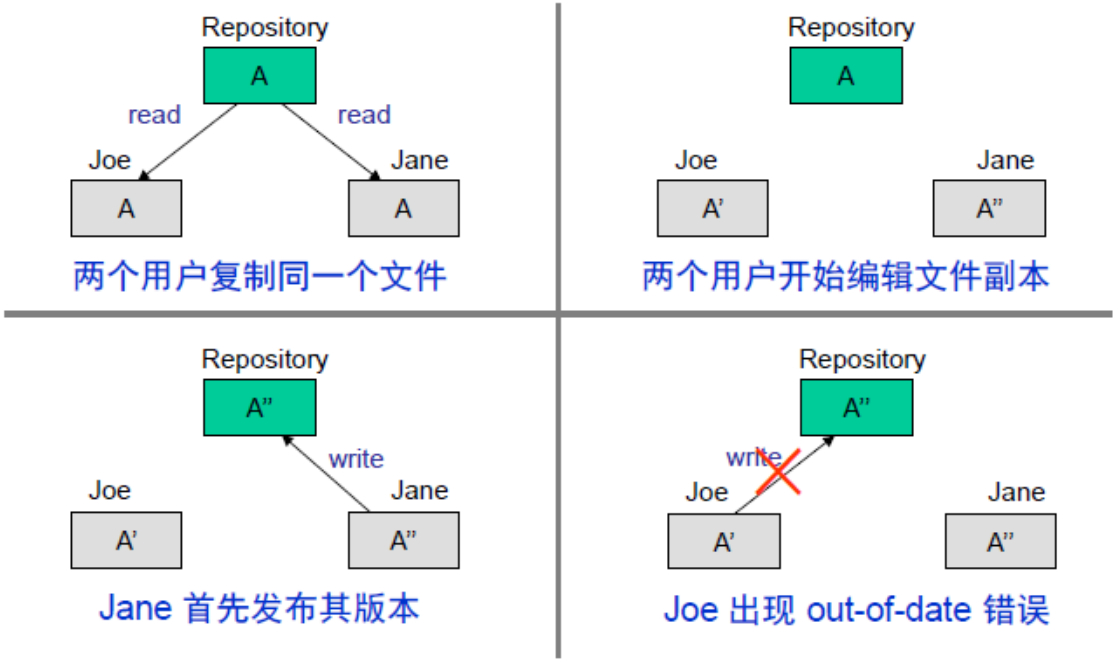
软件配置管理

- 版本控制模型（独占工作模式）



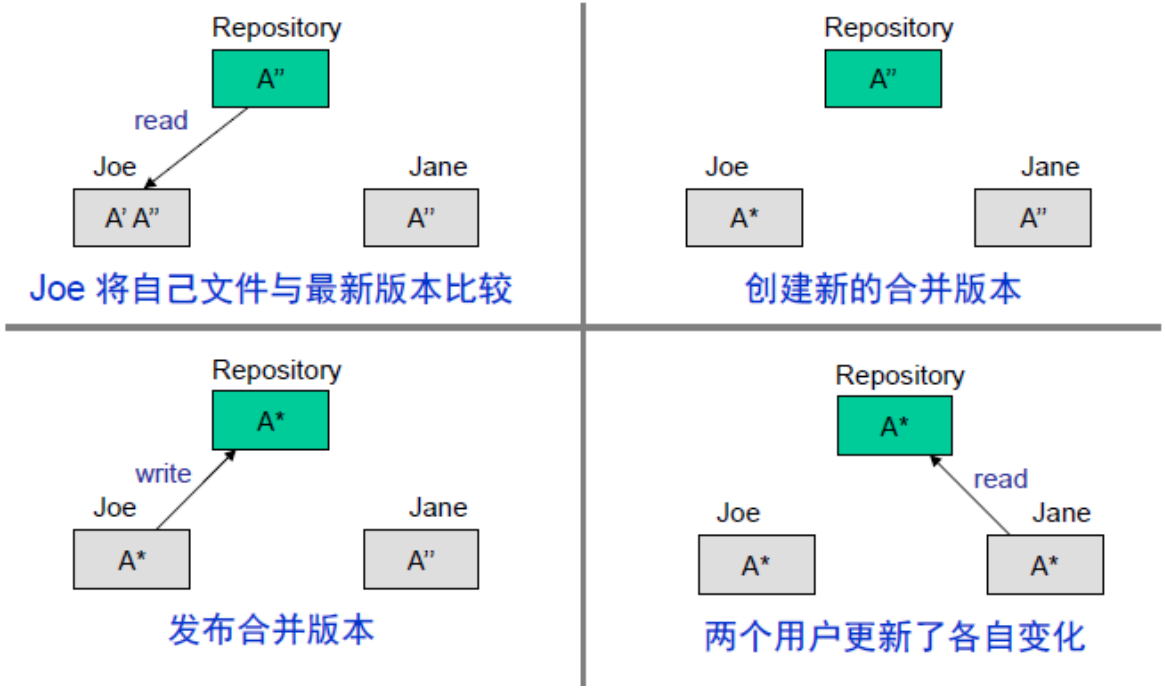
软件配置管理

- 版本控制模型（并行工作模式）



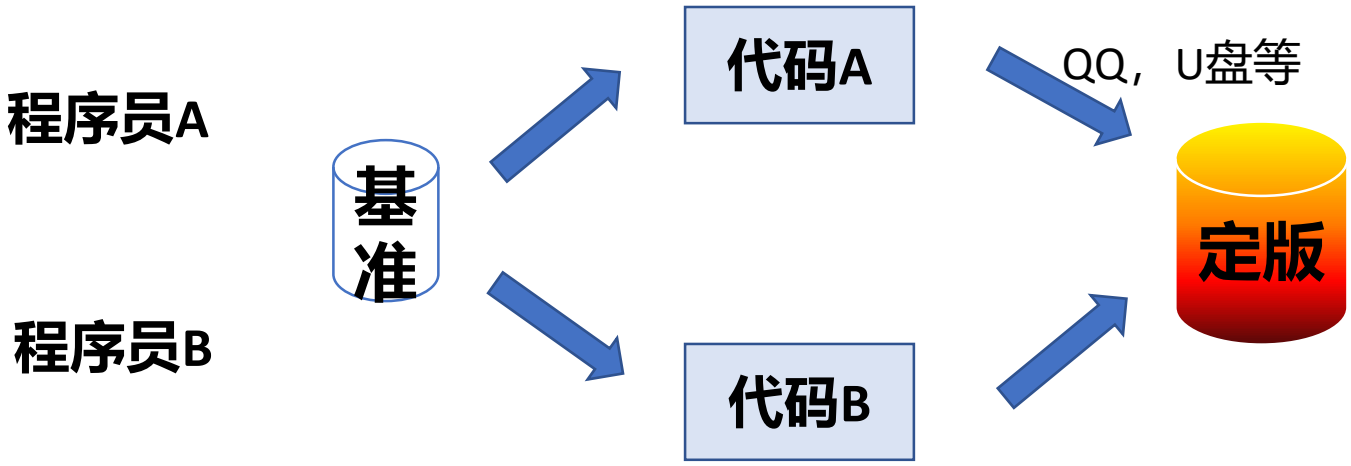
软件配置管理

- 版本控制模型（并行工作模式）



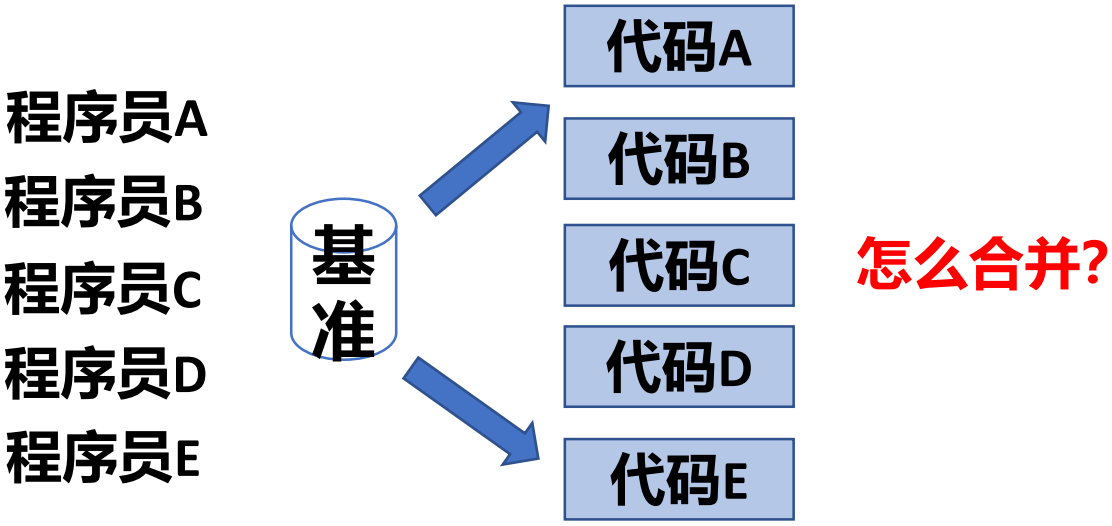
团队开发的版本问题

- 小程序开发
  - 涉及人数1-2人
  - 没有版本的概念



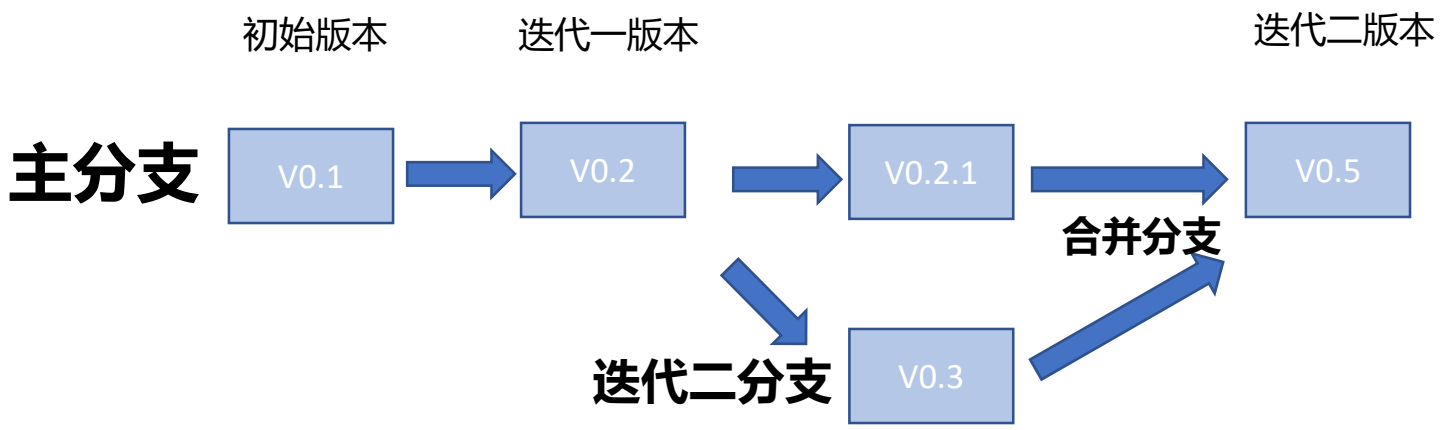
团队开发的版本问题

- 大项目开发
  - 涉及人数4-5人以上
  - 有明确的版本概念（迭代一版本，稳定版等等）



团队开发的版本问题

- 大项目开发
  - 涉及人数4-5人以上
  - 有明确的版本概念（迭代一版本，稳定版等等）



Reference

- 清华大学国家级精品课程 《软件工程》主讲人 刘强 副教授 刘璘 副教授
- https://www.icourses.cn/sCourse/course_3016.html
- https://www.xuetangx.com/course/THU08091000367/5883555?channel=learn_title

谢谢大家！

THANKS