



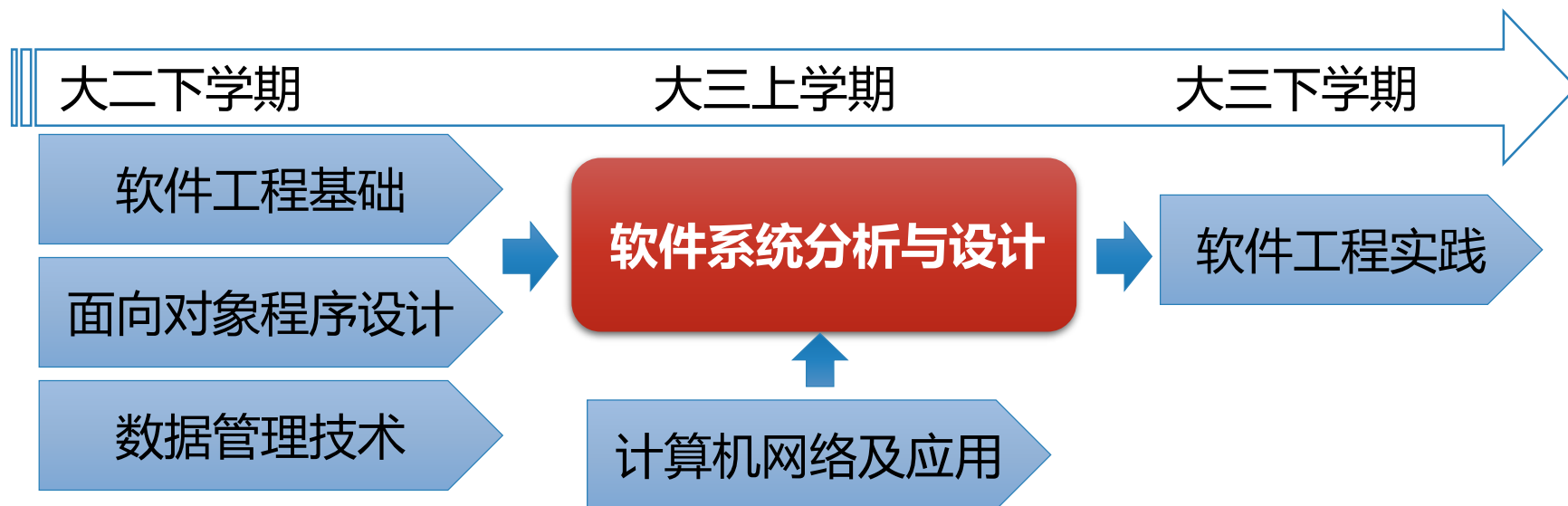
# 软件系统分析与设计

谭火彬，潘海侠

# 关于本课程

## ❖ 目标

- ◆ 通过具体的**案例分析**和**实践训练**，培养学生**利用软件工程思想进行系统分析和设计的能力**，为从事软件开发实践提供必备的技能：需求分析和系统设计以及相关工具的使用等



# 课程教师、助教、微信群和网站

## ❖ 授课老师：谭火彬，潘海侠

◆ Email: thbin@buaa.edu.cn  
haixiapan@buaa.edu.cn

◆ 办公地点：世宁大厦1003

## ❖ 使用多方课程资源

◆ 课程网站、雨课堂

## ❖ 课程网站

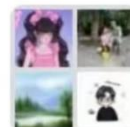
◆ <http://10.134.136.71:9995>

◆ 校内登录，账号和密码为学号

## ❖ 云资源

◆ 后续开发统一申请华为云服务器

◆ 前期可使用华为代码托管平台



课程2022-软件系统分析与设计



该二维码7天内(9月2日前)有效，重新进入将更新

# 软件系统分析与设计

软件系统、软件密集型系统

a software system, a **software-intensive** system

理解并详细说明信息系统应该做什么的过程

the process of understanding and specifying in detail **what** the information system should do

详细说明信息系统的许多组件在物理上是怎样实施的过程

the process of specifying in detail **how** the many component parts of the information system should be implemented

# 什么是分析与设计?

认识问题 → 分析问题 → 解决问题



问题分析—理解需求



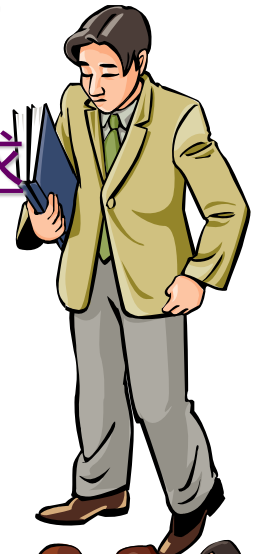
问题定义—获取需求



问题解决—实现需求



最终用户(提出问题)



开发团队(解决问题)

# 教学模式

## ❖ 互动式教学方法

### ◆ 课堂讲解

- 老师主讲
- 以ppt为主，讲解基本概念、方法和案例
- 利用雨课堂，让学生积极参与课程

### ◆ 随课实践

- 学生团队开展实践
- 多次随课实践，要求课后按时完成；理论课程完成后的集中实践

### ◆ 课程讨论

- 学生团队和老师共同主导
- 课堂实践、探讨实践问题、汇报实践内容

# 教学日历 (1 – 8周)

序号	日期	节数	教 学 内 容	备 注
1	8 月 30 日	1-2	讲解：系统分析与设计基础 课后实践：团队组建	
2	9 月 1 日	3-4	讲解：需求调研 课后实践：开展多种形式的需求调研，理解需求	作业期限：9 月 6 日讲解，9 月 9 日 22 点提交截止
3	9 月 6 日	5-6	讨论：需求调研问题研讨 课后实践：完成需求调研	
4	9 月 8 日	7-8	讲解：面向对象与 UML 课后实践：开展需求调研（续）	
5	9 月 13 日	9-10	讲解：需求建模 课后实践：开展需求建模，完成基本的用例模型	作业期限：9 月 20 日讲解，9 月 23 日 22 点提交截止
6	9 月 15 日	11-12	讲解：需求建模（续） 课后实践：继续完成需求模型	
7	9 月 20 日	13-14	讨论：需求建模问题研讨 课后实践：完成规格说明文档	
8	9 月 22 日	15-16	课堂实践：需求建模	
9	9 月 27 日	17-18	讲解：需求建模作业讲解	
10	9 月 29 日	19-20	讲解：对象（领域）分析 课后实践：开展需求分析，完成分析模型	作业期限：10 月 13 日讲解，10 月 19 日 18 点提交截止
11	10 月 11 日	21-22	讲解：对象（领域）分析 课后实践：开展需求分析，完成分析模型	
12	10 月 13 日	23-24	讨论：对象分析问题研讨 课后实践：完成对象分析文档	
13	10 月 18 日	25-26	讲解：对象分析作业讲解 课堂实践：领域对象分析	

# 教学日历 (9 – 13周)

14	10月20日	27-28	讲解：架构设计 课后实践：架构调研和系统架构设计，调研目前市面上成功软件的架构，组织开展研讨	作业期限：10月27日讲解架构调研，11月1日讲解架构设计，11月4日22点提交截止
15	10月25日	29-30	讲解：架构设计（续） 课后实践：系统架构设计和关键技术方案	
16	10月27日	31-32	讨论：架构调研 课后实践：继续系统架构设计和关键技术方案	
17	11月1日	33-34	讨论：架构设计 课后实践：改进架构设计方案	
18	11月3日	35-36	讲解：架构设计作业讲解 课后实践：系统架构设计和关键技术方案	
19	11月8日	37-38	讲解：构件设计-数据设计、界面设计 课后实践：数据架构设计	作业期限：11月16日提交截止
20	11月10日	39-40	讲解：构件设计-对象设计	
21	11月15日	41-42	讲解：构件设计-对象设计（续）	
22	11月17日	43-44	讲解：构件设计-设计原则与模式	
23	11月22日	45-46	讲解：构件设计-设计原则与模式（续）	
24	11月24日	47-48	讲解：系统上线和维护（DevOps）	
25	10月26日-12月7日	49-76	实践：软件开发实践	架构设计完成后，就可以启动开发实践（9-15周）
26	12月14日	77-80	系统演示、答辩、期末考试	第16周（具体时间待定）



# 考核方式

## ❖ 实践和成果讨论(40%)

- ◆ 需求调研 (5%) (1~2周)
- ◆ 需求模型 (10%) (3~6周)
- ◆ 分析模型 (5%) (7~8周)
- ◆ 架构调研与设计 (5%+10%) (9~11周)
- ◆ 数据设计 (5%) (12~13周)

## ❖ 项目展示(20%)

- ◆ 初步计划: 9~16周完成项目开发, 演示

## ❖ 期末考试 (40%)

- ◆ 开卷, 预计17周考试, 具体安排以教务为准

# 作业计分规则

- ❖ 本学期开始作业计分试行互评制，各小组要每次作业都要互评若干个其他组的作业
- ❖ 作业成绩分三部分组成
  - ◆ 教师给分（70%）
  - ◆ 互评得分（20%）
    - 每个组的作业会由若干个其他小组进行评价，取各组给本小组分数的平均分计分
  - ◆ 互评差异分（10%）
    - 评价各小组是否认真评价其他小组的作业；以某份作业的教师给分作为均值，根据各个互评分计算方差，来计算差异分

# 教材与参考资料

- ❖ 1. 谭火彬, UML2面向对象分析与设计 (第2版), 清华大学出版社, 2019.1 (提供配套视频)
- ❖ 2. Jeffrey L. Whitten, Lonnie D. Bentley, Systems Analysis and Design Methods (7<sup>th</sup> ed.), McGraw-Hill, 2005.11 (肖钢, 孙慧译, 系统分析与设计方法, 第7版, 机械工业出版社, 2007.8)
- ❖ 3. Grady Booch, Robert A. Maksimchuk, Object-Oriented Analysis and Design with applications (3<sup>rd</sup> ed.), Addison-Wesley, 2007.5 (王海鹏, 潘加宇译, 面向对象分析与设计, 人民邮电出版社, 2016.5)
- ❖ 4. 徐锋, 有效需求分析, 电子工业出版社, 2017.1
- ❖ 5. 李智慧, 大型网站技术架构: 核心原理与案例分析, 电子工业出版社, 2013.9
- ❖ 6. 王新栋, 架构修炼之道: 亿级网关、平台开放、分布式、微服务、容错等核心技术修炼实践, 电子工业出版社, 2019.3
- ❖ .....



# 系统分析与设计基础

## Chapter 1

你是否参与开发实际的软件系统，有真实的用户使用，并获得一定的资金？

- ☐ A 是，有实际用户，且有收入
- ☐ B 是，有实际用户，但没收到钱
- ☐ C 是，但最后没有上线，也没有实际用户
- ☐ D 否



提交

# 案例研究：图片浏览软件

某个老师 (T) 想要考察一个同学 (S) 的学习情况和技术水平，于是交给该学生一个任务

T: 我有一个朋友想要一个图片浏览软件，能够查看多种格式的图形，包括BMP、TIFF、JPG、PNG，并且能够支持一般的放大、缩小、漫游。你能做这样一个软件吗？

S: 就是类似ACDSee这样的软件吗？

T: 差不多，不过不需要那么强大的功能，我这个朋友计算机是外行，最好能做的比较方便，傻瓜型的，例如象ACDSEE自动翻页这种功能还是要的

S: 我以前学过BMP和JPG的图形格式解析，我想没有问题

T: 好的，给你30天时间，下周你再来一趟，跟我讲一下你的工作进度



# 如何完成该项目?

# 任务列表—分析

这位同学非常明白老师的意图，回去后想了一下，并列出了一个工作清单：

## 一、功能：

1. 读取、显示、另存四种格式图片（BMP、TIFF、JPG、PNG）
2. 放大、缩小、漫游
3. 列出当前目录下所有四种格式图片文件名
4. PAGEUP(PAGEDOWN)自动调出当前目录上一张（下一张）图片

## 二、其它说明（非功能）：

1. 界面尽量简介，容易操作
2. 不要图片预览和打印

(接下页)



# 任务列表—设计

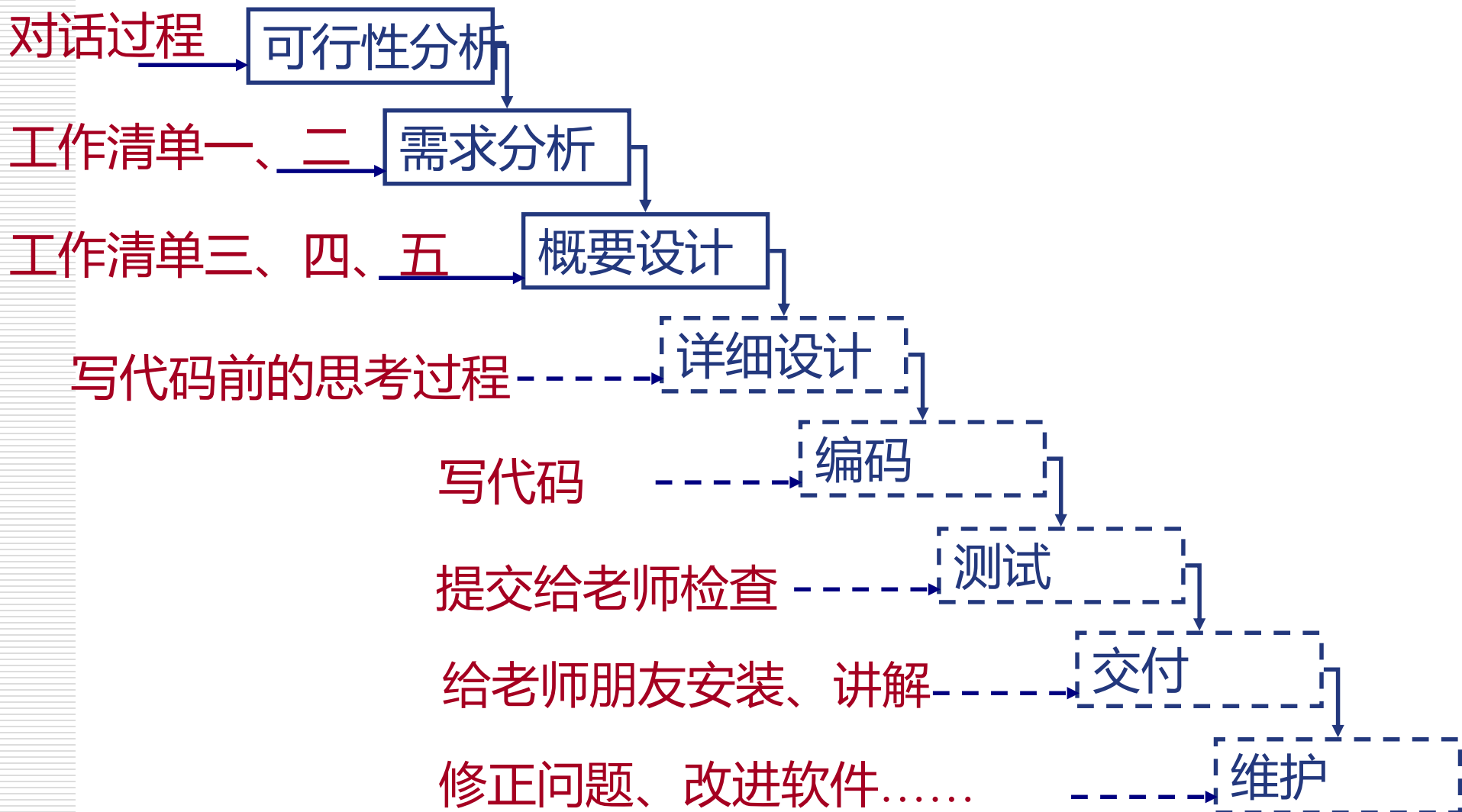
三、开发工具：C#

四、开发环境：普通PC机；Win 7

五、工作量：

1. 研究一下四种图片的格式
2. 设计一个解析器类，解析这四种格式
3. 设计一个文档类，实现读取、另存和目录浏览功能
4. 设计一个视图类，实现显示、缩放、漫游功能

# 软件开发过程



# 结局 (1)

一切顺利，学生S按期交付了软件，经过一两周的试用、修改、完善后，三方都比较满意，该软件在老师的朋友那里成为一个得心应手的工具



理想中的软件工程

## 结局 (2)

一周后，学生去见老师，并提交了工作清单，他发现老师的这位朋友 (C) 和老师在一起

S: 这是工作清单，我已研究了四种文件格式，可以写代码了

T: 很好，不过我这位朋友有一些新想法，你不妨听听

C: 你好。我新买了扫描仪，你的程序能否直接扫描图片进来。

S: 你可以自己扫描呀，买扫描仪的时候一般都会送正版软件的。

C: 是的，可是我一直不太会用，你知道我计算机水平不高，学一些新东西很累，也没有时间，如果你能直接链接扫描仪，我只要学会你的软件就行了，我愿意多支付一些费用.....，还有，我想建一个图片库，你知道，我工作时需要上百个图片，经常找不到，最好还带模糊查询...

## 结局 (2) (续)

S: .....!!!!

C: 还有一些，现在一时想不起来，我想起来的话会再跟你联系，时间上可以长一些。

S: .....!!!! !!!!! !!!!!

T: 要不这样吧，你先做个样子出来给C看看，一边做一边改。

C: 这样最好，看见一个基本样子我就知道我想要什么了。

事情就这样定下来了，S愤怒的撕掉了自己的工作清单.....，回去后S花1天时间用C#做了界面，只能读BMP和JPG文件，做了些菜单和工具栏，用Access建了一个图片库。就这个“假”的程序，S和C讨论了一天，S又修改了几次，又讨论了几次，一周后，这个“假”的程序表面看起来和真的一模一样。

## 结局 (2) (续)

于是S打算重新设计这个程序，但他很快发现在原型上继续补充新的功能更方便，因为至少界面不用重做了，于是……，两个月后，这个事情终于结束了。

S顺利的完成了他的毕业设计《图片浏览软件的设计与实现》，C一直使用这个软件管理他的图片，并庆幸花了这么少的钱得到了这么有用的东西，而T，则正在考虑如何为他下一批学生分派任务。



现实中的软件工程

## 结局 (3)

正像上一种情况一样，用户提出了很多新要求，但是麻烦还不止这些……。一天，老师T匆匆忙忙的找到S。

T: 我的研究生正在做的“海量多媒体数据库管理技术”的基金项目，需要一个对图像管理的模块，主要是数据库对象和图像文件之间的转换、显示和一些编辑操作，时间很紧，你目前在做的代码可否直接利用一下？

S: 恐怕有难度，我不清楚……

T: 最好能够模块化强一些，你做的东西两边都能用，我这边比较急，一周后就要，我可以给你增加一个人一起做。

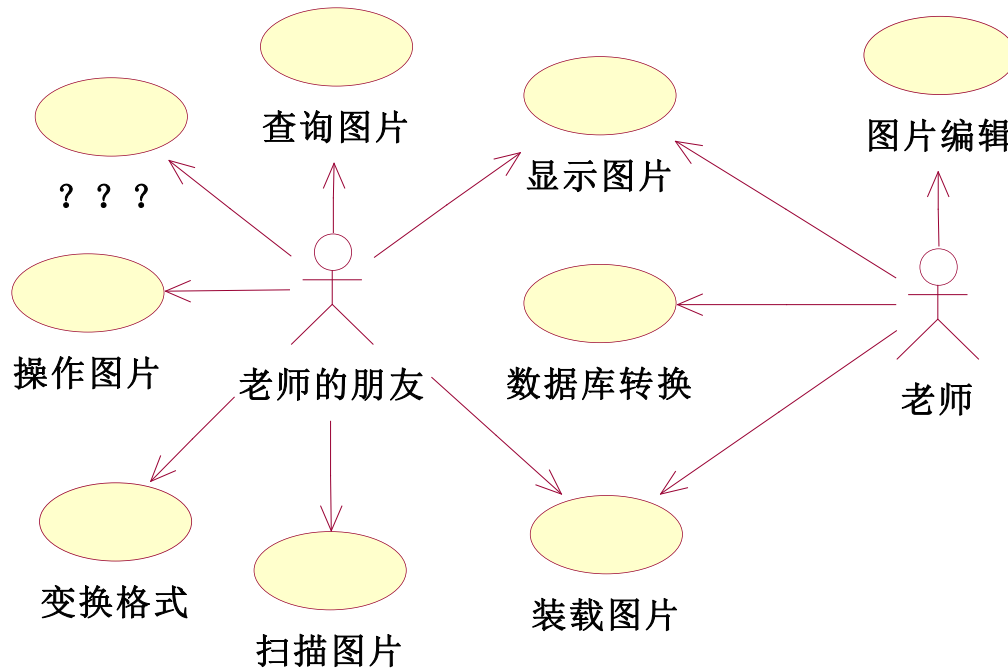
S: 可是……

T: 没有关系，就这样决定了，这是一次锻炼机会。我再帮你找一个这方面的专家，你可以请教他。下周这个时间我会再来。

## 结局 (3) (续)

S感觉头脑里面“海量”、“图像”、“编辑”、“复用”、“一周时间”等等交织在一起，剪不清，理还乱。于是他准备去请教一下专家(E)。

E听了S说的情况，帮他画了两个图：



## UML: Use Case Diagram



# 结局 (3) (续)

## 图片文件

📁 文件名 : string  
📁 文件大小 : int  
📁 文件类型 : int  
📁 文件属性 : int

💎 读取()  
💎 存储()  
💎 另存()

## 图片格式解析器

📁 解析器类型 : int

💎 解析() : 图片文件

## 图片扫描管理器

📁 扫描仪参数 : int

💎 打开扫描仪()  
💎 扫描()

## 文件图片库

📁 位置 : string  
📁 文件列表 : vector  
📁 文件数量 : int  
📁 图片类型 : int  
📁 图片库名称 : string

💎 增加()  
💎 删除()  
💎 遍历()  
💎 查询()  
💎 模糊查询()

## 图片编辑器

💎 剪切()  
💎 锐化()  
💎 模糊()  
💎 平滑()  
💎 合并()

## 图片显示器

📁 当前图片 : int

💎 显示图片()  
💎 放大()  
💎 缩小()  
💎 刷新()

# UML: 类图

## 结局 (3) (续)

E要求S自己再画这样几张图：结合用例所描述的和业务和系统类图，采用其它的UML模型设计个功能模块。

S试着这样做，采用类图、顺序图等完成下列模块的设计：

1. 图片文件类模块和图片库类模块
2. 图片格式解析器父类模块；5个图片解析子类模块（4个文件格式和一个数据库格式）
3. 图片扫描管理器模块
4. 图片编辑器模块
5. 图片显示器模块

S发现在网上有很多现成的图片扫描管理控件和图片编辑控件，完全满足要求，他自己花了一天一夜的时间编写了图片文件类模块和图片格式解析器父类，以及数据库解析子类，剩下的几天，他和老师新来的同学一起完成了剩余的模块。

## 结局 (3) (续)

一周过去了，他将图片文件类模块、图片格式解析器父类模块、数据库解析子类，以及自己封装的图片编辑器交给了自己的老师，而由于每一个模块都是相对独立的，即使开始的用户要求修改图片显示、图片库、扫描，也不会影响他现在的工作代码。



# 规范化的软件工程实践

# 软件开发难在何处?

## 1. 业务问题

- ◆ 用户无法明确定义自己所要的
- ◆ 开发人员无法理解用户的要求
- ◆ 用户的想法经常发生变化



**需求**

## 2. 软件（架构）问题

- ◆ 软件功能无法满足用户需求
- ◆ 软件功能无法应对需求变更
- ◆ 软件性能等非功能无法达到用户要求



**分析  
设计**

## 3. 实现技术问题

- ◆ 编程语言、开发效率、代码质量等



**编码**

## 4. 可持续发展问题

- ◆ 软件维护成本日益增加
- ◆ 无法满足新需求、新技术



**维护**



# 讨论



## 教务系统集成单点登录服务

某教务系统原来独立运行，具有独立的帐号体系；现按照学校的统一要求，需集成学校单点登录服务实现统一登录认证，如果让你来实施这个项目，需要怎么做？

# 复习: 软件?

- ❖ **软件**是计算机系统中与硬件相互依存的另一部分，它是包括程序，数据及其相关文档的完整集合
  - ◆ **程序**是按事先设计的功能和性能要求执行的指令序列
  - ◆ **数据**是使程序能正常操纵信息的数据结构
  - ◆ **文档**是与程序开发，维护和使用有关的图文材料

# 复习：软件工程？

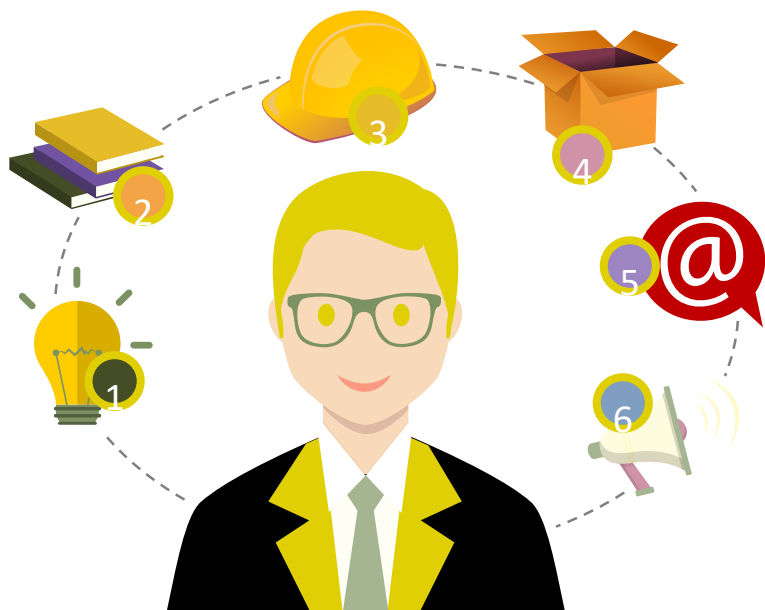
## ❖ Software Engineering (IEEE1993):

- ◆ (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software
- ◆ (2) The study of approaches as in (1)

## ❖ 软件工程 = 软件 + 工程

- ◆ 运用工程学的原理和方法来组织和管理软件的生产和管理，以保证软件产品开发、运行和维护的高质量和高生产率

# 思考?



## 软件工程 V.S. 文档

- ◆ 软件工程是一系列.....原理、方法.....
- ◆ 文档只是一种确保这些原理、方法得到有效应用的辅助手段

不写文档是否可行?  
是否也体现了软件工程原理、方法等?



# 软件工程（续）

## ❖ 维基百科（Webopedia）

◆ The computer science discipline concerned with developing **large applications**

- Software engineering covers not only the technical aspects of building software systems
- but also management issues, such as directing programming teams, scheduling, and budgeting

# 复习：信息系统

## ❖ 系统

- ◆ 相互作用的多元素的复合体
- ◆ 系统的整体特性不等于局部特征之和

## ❖ 信息系统(information system )

- ◆ 是人、数据、过程和信息技术的组合，通过相互作用，收集、处理、存储和提供支持机构运作的信息

## ❖ 信息技术(Information technology )

- ◆ 是关于信息的产生、发送、传输、接收、变换、识别、控制等应用技术的总称，是在信息科学的基本原理和方法的指导下扩展人类信息处理功能的技术
- ◆ 其主要支柱是通讯(Communication)技术、计算机(Computer)技术和控制(Control)技术

# 信息系统的组成部分



# 复习: 项目干系人

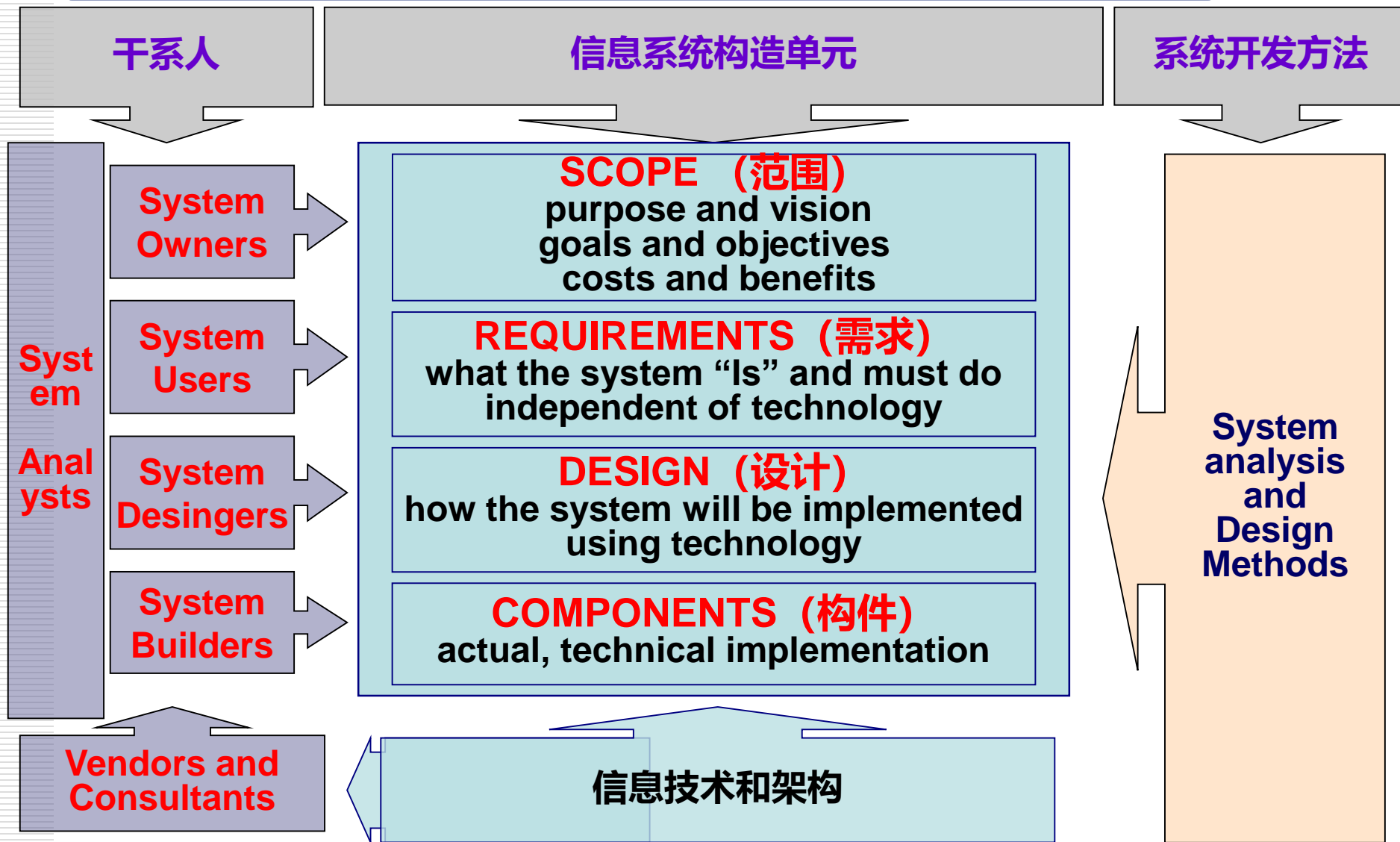
## ❖项目干系人 (Stakeholders, 系统关联人员):

- ◆与系统建设和使用有关系的任何人
- ◆技术人员或非技术人员

## ❖主要分类

- ◆系统所有者(System owners)
- ◆系统用户(System users)
- ◆系统分析师(System analysts)
- ◆系统设计员(System designers)
- ◆系统构造人员 (System builders)
- ◆IT厂商与顾问(IT vendors and consultants)

# 系统开发视角的项目干系人视图



# 系统所有者与用户

- ❖ 系统所有者(System owners)
  - ◆ 系统的发起人和倡导者, 信息系统开发、运行、维护的投资者
- ❖ 系统用户(System users)
  - ◆ 信息系统的使用者或其工作受信息系统的存在而影响的工作人员
  - ◆ 内部用户( Internal users)
  - ◆ 外部用户(External users)
    - 远程和移动用户
    - 顾客、供应商、合作伙伴

# IT厂商与顾问

## ❖IT厂商与顾问(IT vendors and consultants)

- ◆出售软件、硬件厂商，集成商等
- ◆出售特殊的专业知识和经验，为项目的完成提供服务。一般类属于咨询机构或服务机构
- ◆如：技术工程师、销售工程师、系统顾问、系统集成、签约程序员、法律顾问等等

## ❖关于IT服务

- ◆利用行业专家和技术专家，帮助行业用户建立、管理和优化信息获取及业务流程的服务

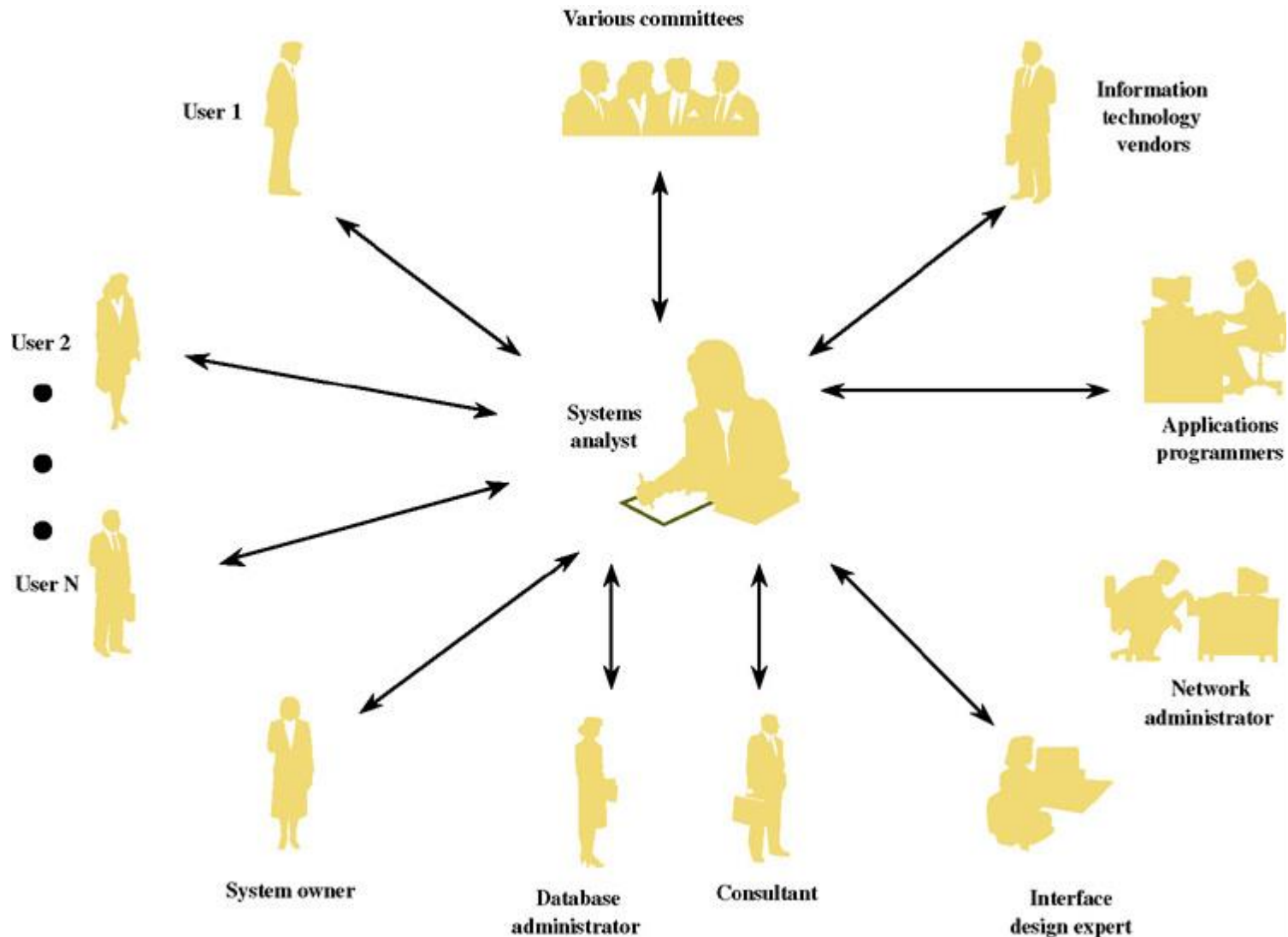
# 系统分析师

## ❖ 系统分析师(Systems analyst)

- ◆ 研究组织存在的问题和需求确定人员、数据、过程、信息技术的使用和有机结合达到最大化对企业作贡献的目的
- ◆ 业务分析员、系统架构师、系统工程师、信息工程师、信息分析员、系统集成师



# 系统分析是系统核心角色



# 计算机职业道德的10条戒律

1. 不能使用计算机**伤害**别人
2. 不能**干预**其他人的计算机工作
3. 不能**窥探**其他人的计算机文件
4. 不能使用计算机**盗窃**
5. 不能使用计算机作**伪证**
6. 不能**非法复制和使用**那些没有付费的专用软件
7. **未经授权**或认可，不得使用其他人员的计算机资源
8. 不可**盗用**他人的智力成果
9. 应该考虑你正在编写的程序或正在设计系统的**社会影响**
10. 在使用计算机时，必须确保对他人的体贴和**尊重**

# 附录1：团队建设

## ❖ 团队：8~10人

- ◆ 角色划分：团队必须确定1~2名具有绝对权威的负责人，负责管理团队并和老师、助教沟通，并有权分配团队内部各成员的成绩；团队中建议安排专人负责提交作业、参与作业互评
- ◆ 其他人员分工根据每次作业的情况再另行确定
- ◆ 在网站上完成团队组建工作
  - 请务必在网站上设置正确的个人邮箱，确保可以收到邮件
  - 负责人创建团队、其他人员加入团队
- ◆ 其他说明
  - 允许跨班分组，但后续答辩分组由听从任课老师安排