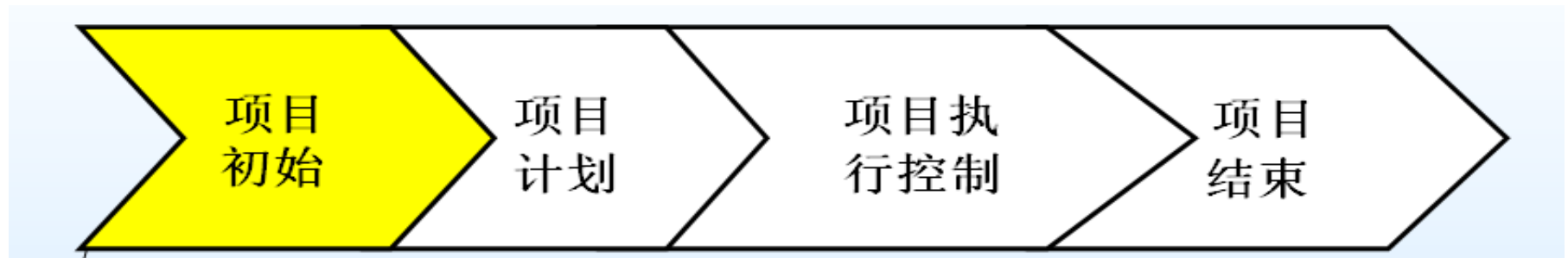
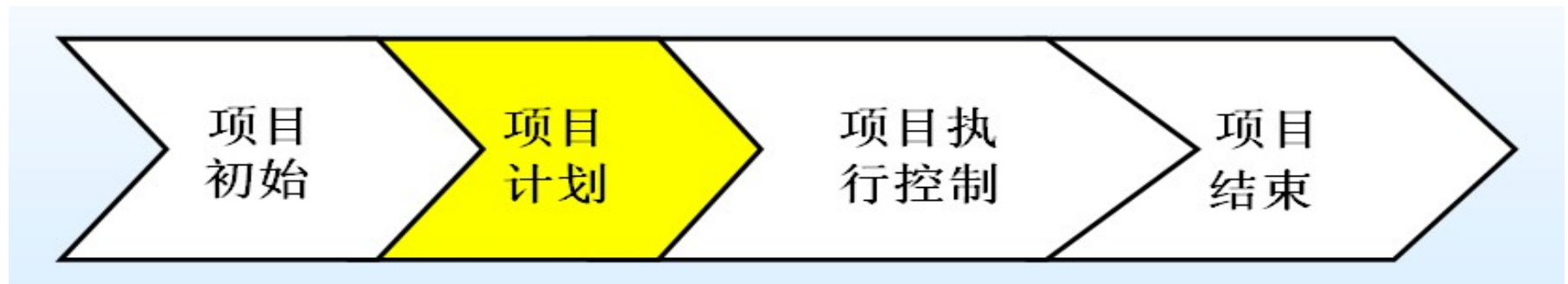


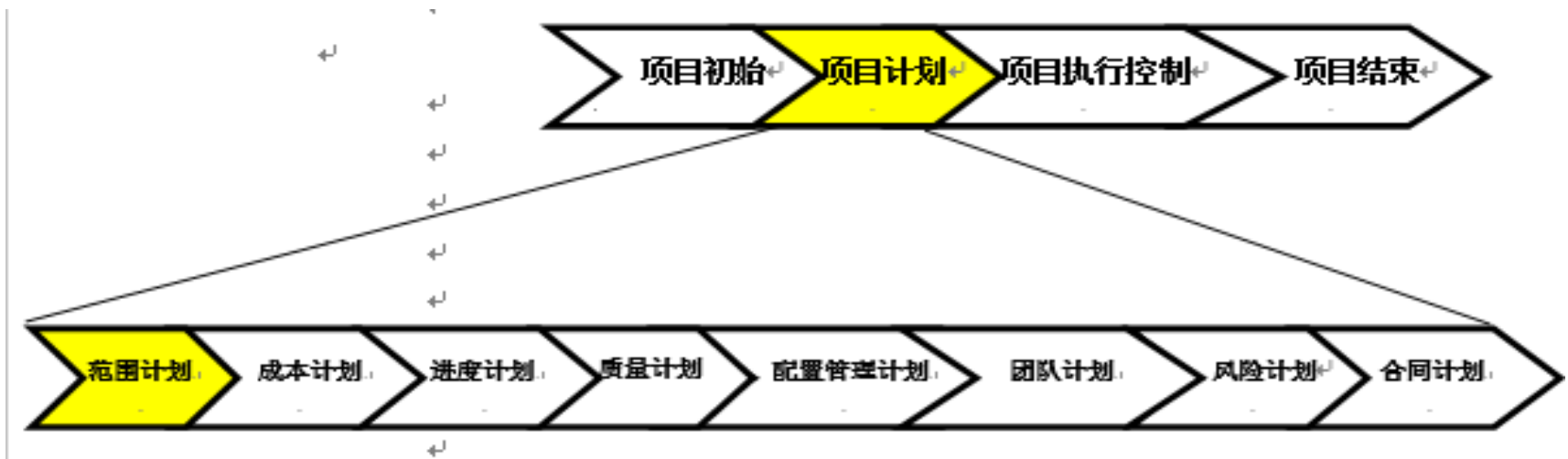
# 承上启下



# 项目计划



# 范围计划





# 软件项目管理 第二篇

## 第 4 章 软件项目需求管理



## 需求的隐含错误



用户不断增加需求、变更需求



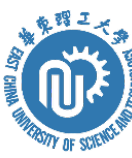
# 项目失败的原因分析



No.	Top 10 Factors	平均值
1	<b>Inadequate requirements specification</b> 不充分的需求规范	<b>4.5</b>
2	<b>Changes in requirements</b> 需求的改变	<b>4.3</b>
3	<b>Shortage of systems engineers</b> 缺乏系统工程师	<b>4.2</b>
4	<b>Shortage of software managers</b> 缺乏了解软件特性的经理人	<b>4.1</b>
5	<b>Shortage of qualified project managers</b> 缺乏合格的项目经理	<b>4.1</b>
6	<b>Shortage of software engineers</b> 缺乏软件工程师	<b>3.9</b>
7	<b>Fixed - price contract</b> 固定价合同	<b>3.8</b>
8	<b>Inadequate communications for system integration</b> 系统集成阶段, 交流与沟通不充分	<b>3.8</b>
9	<b>Insufficient experience as team</b> 团队缺乏经验	<b>3.6</b>
10	<b>Shortage of application domain experts</b> 缺乏应用领域专家	<b>3.6</b>
Scale: 5 = Very Serious    3 = Serious    1 = No Serious		

Source: Carnegie-Mellon University, Software Engineering Institute

# 软件需求定义



需求是指用户对软件的功能和性能的要求。

精品课程建设项目“软件项目管理课程” - Windows Internet Explorer

http://www.buptsse.cn/SPM/JSP/index.jsp

查看(V) 收藏夹(A) 工具(T) 帮助(H)

建议网站 免费 Hotmail 3万 网页快讯库

精品课程建设项目“软件项目管理课程”

北京邮电大学软件学院 School Of software Engineering Of BUPT 软件项目管理

首页 成绩查询 行业信息 下载区 留言板

### 课程介绍

- 课程简介
- 教学大纲
- 课时安排
- 课程特色
- 考评方式
- 参考书目

### 课程内容

- 授课教案
- 教学录像
- 练习题
- 知识点索引
- 考试大纲
- 模拟试卷

### 实验与实践

### 课程介绍

随着信息技术的飞速发展,软件产品的规模也越来越庞大,各软件企业都在积极将软件项目管理引入到开发活动中,对软件项目实施有效的管理。软件项目管理是为了使软件项目能够按照预定的成本、进度、质量顺利完成,而进行分析和管理的活动。另外,进行软件项目管理有利于将个人开发能力转化成企业的开发能力,企业的软件开发能力越高,表明这个企业的软件生产越趋向于成熟,企业越能够稳定发展。特别是...



### 课程特色

本课程具有以下特点:

- 1、实践性强。
- 2、合理安排实验。
- 3、经典理论与新型技术相

### 最新动态

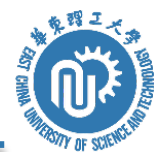
- 关于...的问题!
- 新的淘金点!Google测试交互式Widget广告...
- Google将推出PowerPoint和Wiki!
- IBM免费办公软件Lotus发布 冲击微软市场!
- 欧盟称Vista影响电脑安全 可能禁止其销售...

### 教师信箱

casey\_han@263.net



# 本章要点



一

软件需求管理过程

二

传统需求建模方法

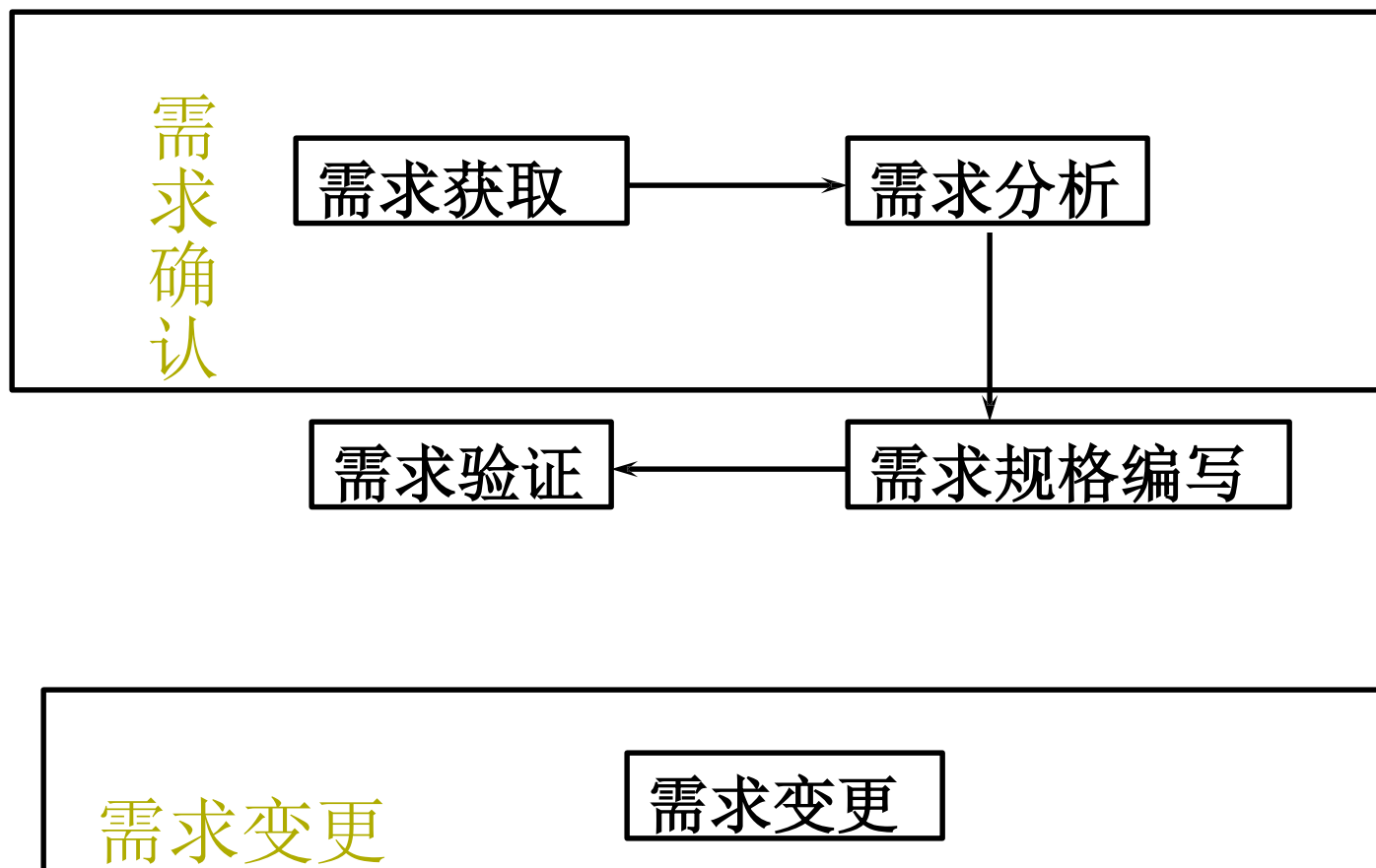
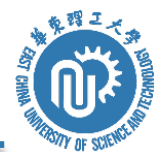
三

敏捷需求建模方法

四

案例分析

# 软件需求管理的过程



# 1、需求获取

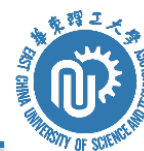
Requirements: What's the Point?



# 需求获取的方法

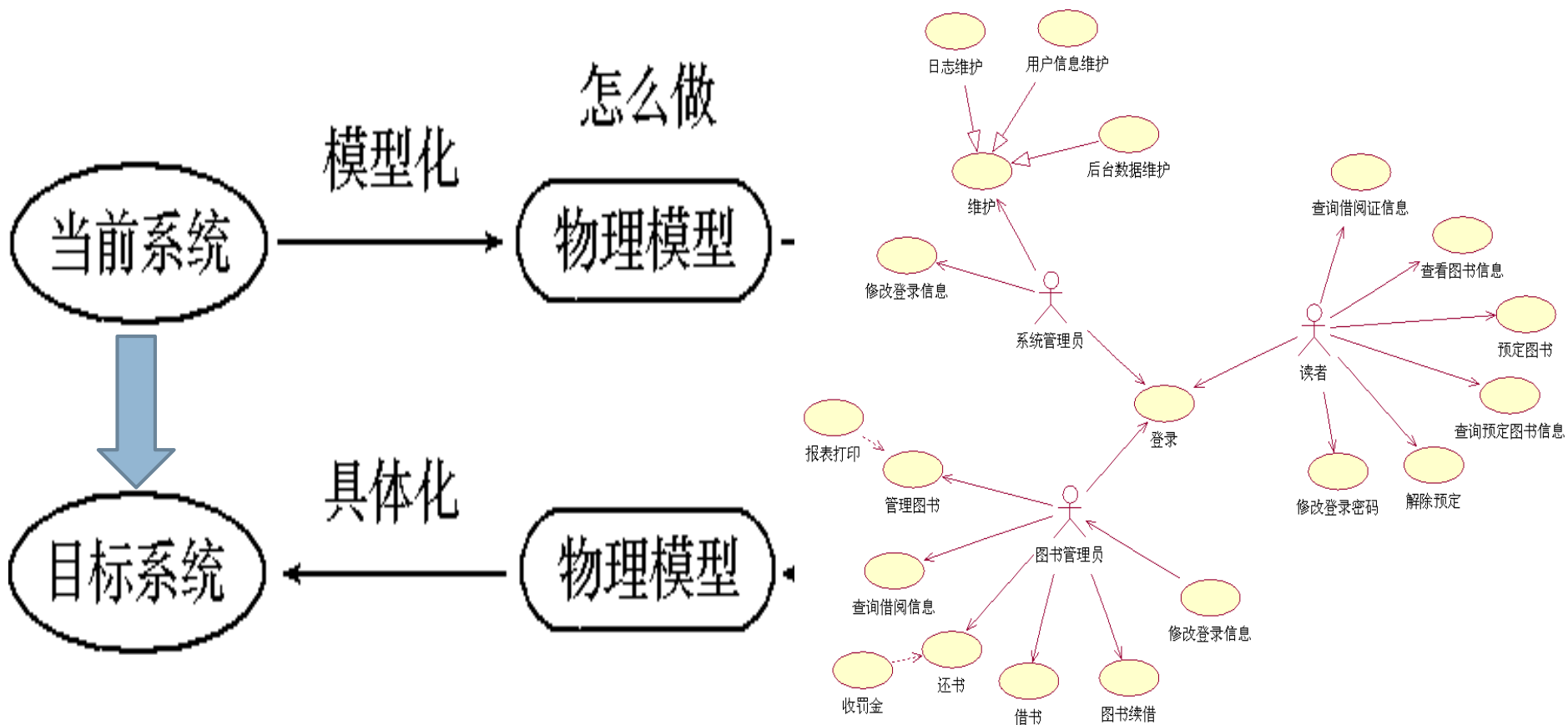


## 2、需求分析



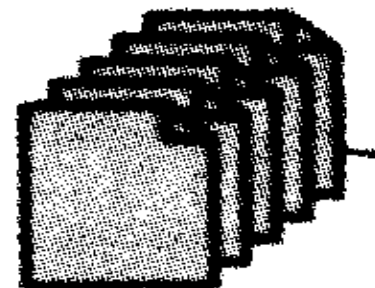
需求分析是为最终用户所看到的系统建立一个概念模型，是对需求的抽象描述。

# 需求分析模型

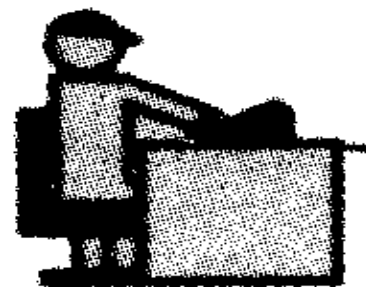


### 3、需求规格编写

需求分析工作完成的一个基本标志是形成了一份完整的、规范的需求规格说明书



要求管理



软件开发

# 需求规格文档参考

文档类别：  
使用对象：  
1. 引言：  
1.1 目的：  
1.2 范围：  
1.3 缩写说明：  
1.4 术语定义：  
1.5 引用标准：  
1.6 参考文献：  
1.7 版本更新条件：  
1.8 版本更新信息：  
2. 系统定义：  
3. 应用环境：  
4. 功能规格：  
5. 性能需求：  
6. 产品提交：  
7. 其他约束：  
对于电信网络管理项目，需求规格书还应包括“信息模型”章节。

## 4. 内容定义

这一章分节描述了需求规格中各个部分的内容要求。

### 4.1 引言

本节为文档标准的部分，详细内容参见引用标准[2]。

### 4.2 系统定义

本节描述内容包括：

- 项目来源及背景；
- 项目要达到的目标，如市场目标、技术目标等；
- 系统整体结构，如系统框架、系统提供的系统功能、涉及的接口等；
- 系统级部分结构。

### 4.3 应用环境

本节应概括用户的要求对系统的运行环境进行定义，描述内容包括：

- 设备环境；
- 系统运行硬件环境；
- 系统运行软件环境；
- 系统运行网络环境。

- 用户操作模式；
- 当前应用环境。

设备环境：

本系统运行时所管理的设备，在非管理项目中，可不包括此部分。

系统硬件环境：

本系统运行时所依赖的硬件平台，因为这些因素会影响到系统的设计和实现，在系统的开发中有时还会涉及对这些硬件的编程。

系统软件环境：

软件系统包括操作系统、语言环境、应用软件和相应的 API，仅仅当这些软件系统是用户要求的双边产品策略确定的，才需列在此处，否则软件支持环境将是一个设计决策问题。

系统网络环境：

网络环境是系统运行时所依赖的网络通信协议支持环境，基于上述两端的理由，只有当这些协议是必须时，才需列在此处。

用户操作模式：

通过用户对模式的描述，能够帮助我们进一步明确了用户需求，使系统更好地为用户服务，这部分应明确：

- 用户需要完成哪些工作；
- 用户在日常工作中如何完成工作；
- 是否新系统；
- 用户期望的系统模式。

当前应用环境：

描述当前用户所拥有的与系统有关的技术环境，包括：

- 网络环境；
- 硬件环境；
- 软件环境；
- 外部系统接口。

### 4.4 功能规格

本节应概括合同中定义的系统级部分分别描述其功能，描述应包括：

- 功能编号；
- 所属产品编号；
- 优先级；
- 功能定义；
- 功能描述。

功能编号：

按照功能实现的要求对功能进行拆分和编号，便于管理和实现验证。

所属产品编号：

按照合同规定的产品编号确定该功能所属的产品模块，无合同或合同中无产品编号时可无此项。

功能实现优先级：

按照用户提出的需求的紧急性和关联程度对功能实现的优先级，特进行排序，此项根据具体情况可选。

功能定义：

对于所提供的功能进行定义，要求定义必须明确，不能产生歧义性。

功能描述：

描述功能实现的过程，可采用自然语言、脚本语言及面向对象的方法进行。

### 4.5 性能需求

本节描述用户对系统的性能需求，可能的系统性能需求有：

- 系统响应时间需求；
- 系统吞吐量需求；
- 系统可靠性需求；
- 系统可扩展性和可维护性需求；
- 系统安全性需求；
- 现有资源利用率需求。

### 4.6 产品提交

本节描述产品提交的方式、时间需求、产品安装的需求以及产品维护的需求，具体描述如下：

产品提交方式：

包括产品交付介质、复制要求、产品的包装方式及产品提交的流程等。

产品提交时间需求：

有时在一个项目中可能会出现多次产品提交，因此在此需将产品提交的时间要求定义清楚。

产品安装的需求：

定义是否需要产品安装，安装的场所及范围，安装计划的确认方式等。

产品维护的需求：

定义产品维护的启动方式、维护期限、版本升级要求以及维护是否收费等要求。

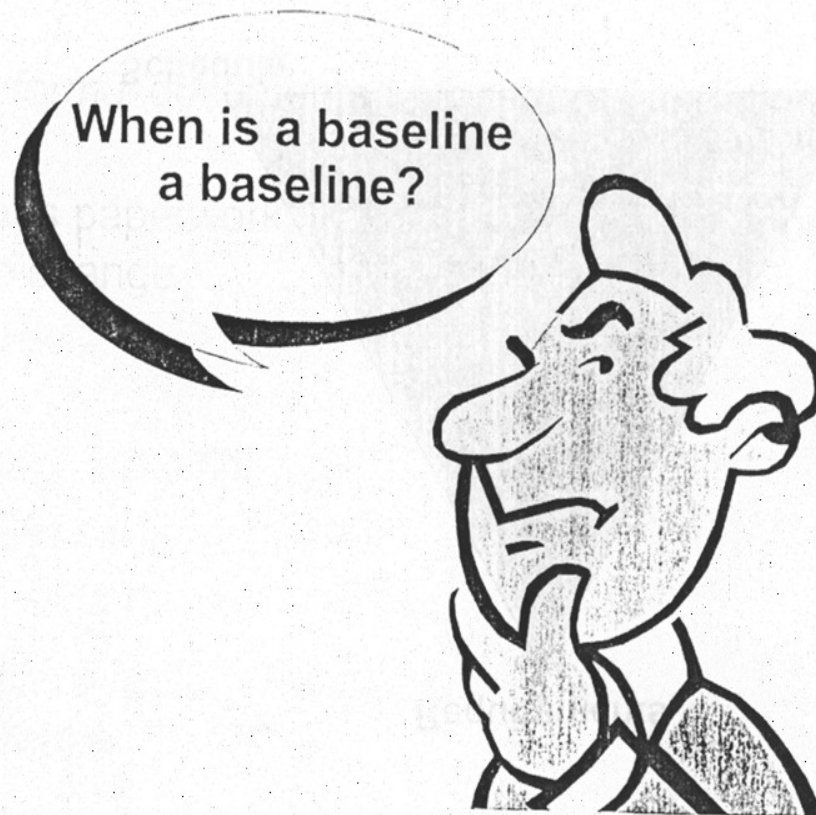


## 4、需求验证



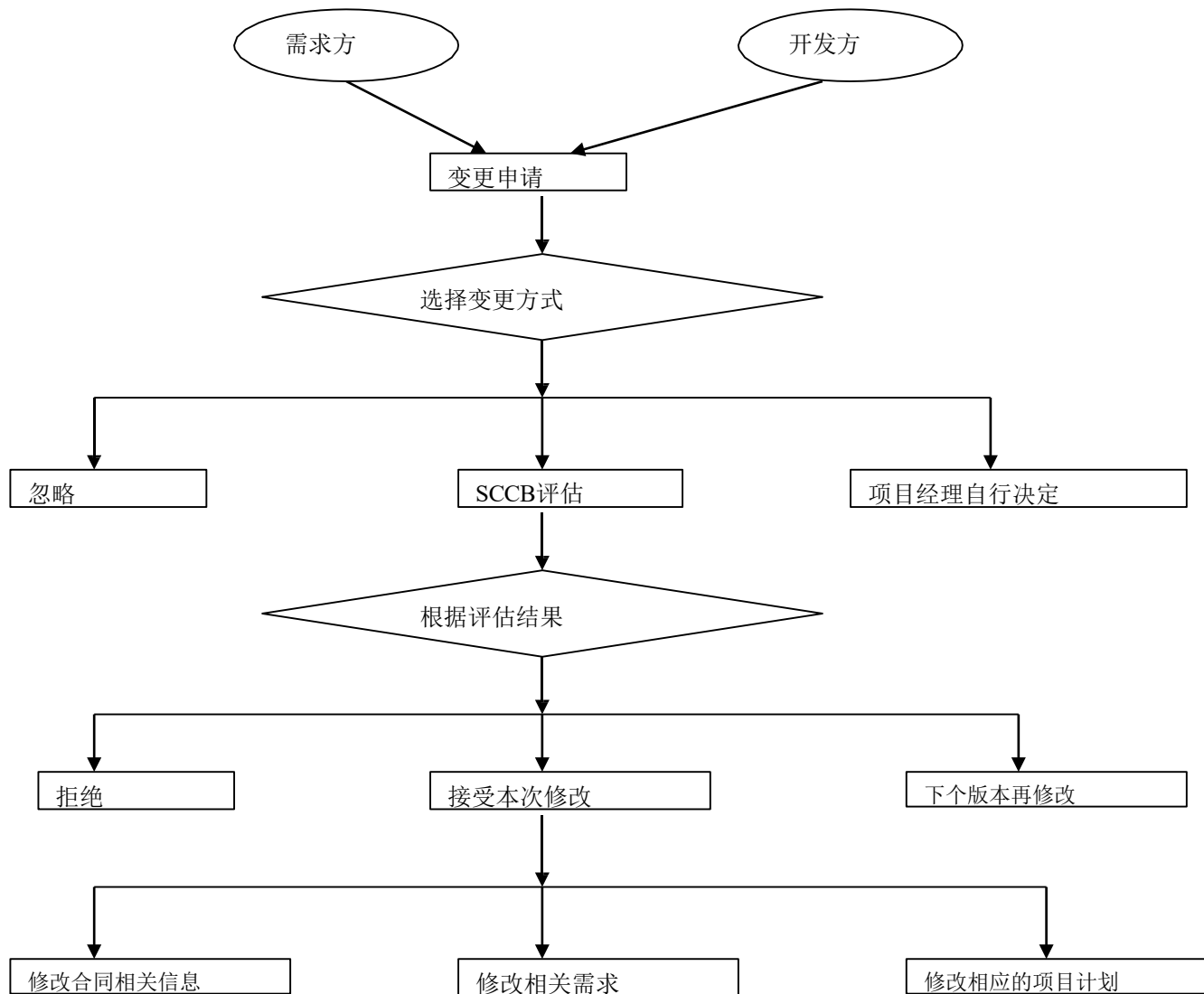
- 需求是正确的吗？
- 需求是一致的吗？
- 需求是完全的吗？
- 需求是实际可行的吗？
- 需求是必要的吗？
- 需求是可检验的吗？
- 需求是可跟踪的吗？
- 最后的签字

## 5、需求总在变化

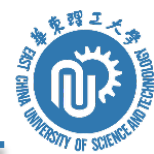


- ① 确定需求变更控制过程
- ② 建立变更控制委员会(SCCB)
- ③ 进行需求变更影响分析
- ④ 跟踪所有受需求变更影响的工作产品
- ⑤ 建立需求基准版本和需求控制版本文档
- ⑥ 维护需求变更的历史记录
- ⑦ 跟踪每项需求的状态
- ⑧ 衡量需求稳定性

# 需求变更控制流程



# 本章要点



一

软件需求管理过程

二

传统需求建模方法

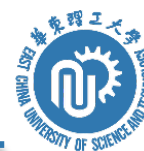
三

敏捷需求建模方法

四

案例分析

# 需求建模的基本方法介绍



传统方法:

1. 原型方法
2. 基于数据流建模
3. 基于UML建模

1. 敏捷方法

第一章 引言	
1.1 软件缺陷跟踪管理系统介绍	
1.1.1 项目信息管理系统介绍	
1.1.2 缺陷管理系统现状	
1.2 系统课题任务	
1.3 论文结构	
第二章 系统关键技术与架构简介	
2.1 SSH 框架	
2.1.1 表示层框架 Struts	
2.1.2 业务逻辑层框架 Spring	
2.1.3 数据持久化层框架 Hibernate	
2.2 Tomcat	
2.3 MySQL	
第三章 系统的需求分析	
3.1 系统需求概述	
3.2 系统功能性需求	
3.2.1 系统角色分配	
3.2.2 登录退出模块	
3.2.3 缺陷管理模块	
3.2.4 个人信息管理模块	
3.2.5 系统管理模块	
3.3 系统非功能性需求	
3.3.1 产品易用性	
3.3.2 产品可靠性	
3.3.3 产品性能	
第四章 系统的设计与实现	
4.1 系统总体设计	
4.2 系统开发环境	
4.3 系统详细设计	

#### c) 查看所有缺陷

- 功能简述：以列表方式列出当前系统记录的所有缺陷，系统管理员可以选择某条记录进行修改或是删除操作。
- 前置条件：登录系统。
- 用例图示：所有缺陷列出功能用例如图 3-5 所示。

#### d) 缺陷过滤统计

- 功能简述：用户可以按照某种规则过滤筛选缺陷，并对满足该规则的缺陷进行分类统计分析。统计生成改规则下各分类缺陷的数量和百分比，并以饼状图的形式表现出来。
- 前置条件：登录系统。
- 用例图示：缺陷统计功能用例如图 3-5 所示。

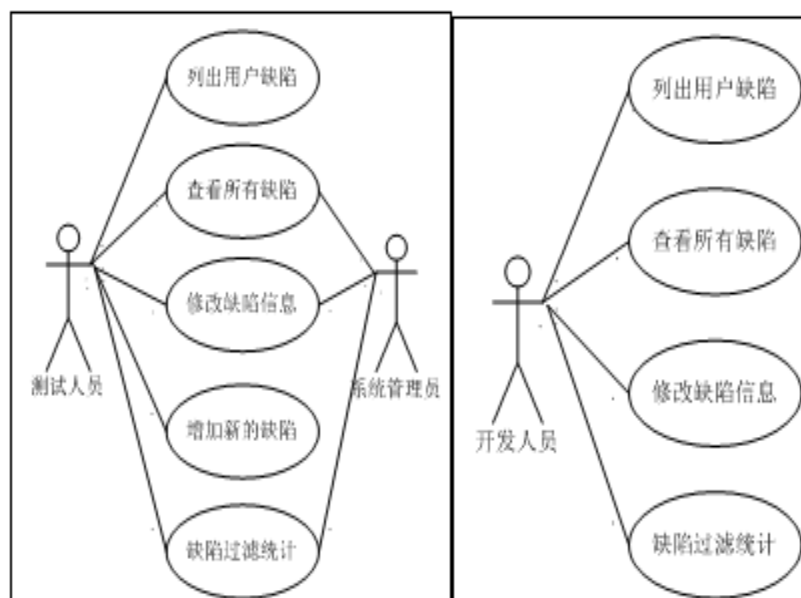


图 3-5 缺陷管理模块用例图

#### 3.2.4 个人信息管理模块

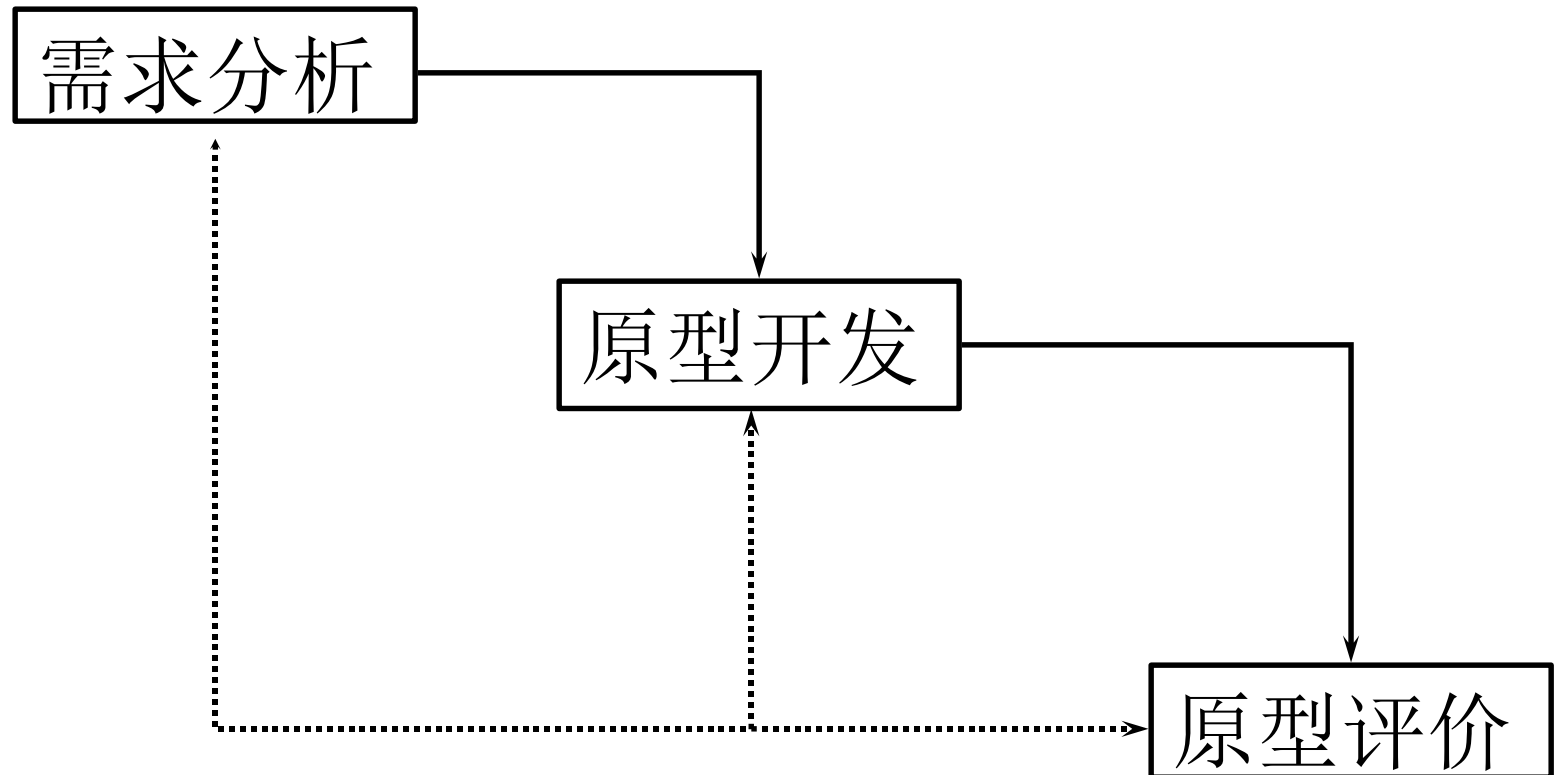
# 传统需求建模的基本方法介绍



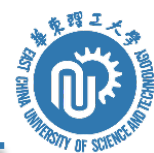
1. 原型方法
2. 基于数据流建模
3. 基于UML建模



# 1、原型方法



# 原型实例



XX公司内部办公门户网站

办公平台 | 邮件 | 会员管理 | 诚信管理 | 联系通道 | 独立董事 | 固定资产 | 人事

我的工作平台

通知公告

- 会内公告

- 部委信息

- 办公室

- 党办

- 会员部

- 资格部

- 联络部

- 国际部

- 发展战略委

- 执业标准委

- 教育培训委

- 金融创新委

- 部门留言板

- 个人信息

我的办公桌

- 通讯录

- 公司黄页

- 部门通讯录

- 个人通讯录

- 沪深行情指数

- 法律法规查询

- 每周工作预览

- 部门工作预览

- 工作监督平台

新建 删除 属性 移动

办公室信息发布

部门公告

内部信息

活动园地

部门信息共享

撰写信息 审批信息 删除信息 搜索信息 拷贝信息 移动信息

首页 前页 后页 末页 跳到  页 信息共 75 条 1 / 5 页

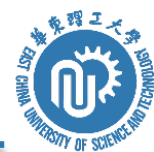
发布时间 升序 重排

	标题	作者	发表日期	附件数量
<input type="checkbox"/>	放假通知		2003-07-12	0
<input type="checkbox"/>	通知2		2003-07-12	0
<input type="checkbox"/>	新闻1		2003-05-26	2[ <a href="#">下载</a> ]

猜你喜欢

100%

# 传统需求建模的基本方法介绍



1. 原型方法
2. 基于数据流建模
3. 基于UML建模

## 2、基于数据流-结构化分析方法



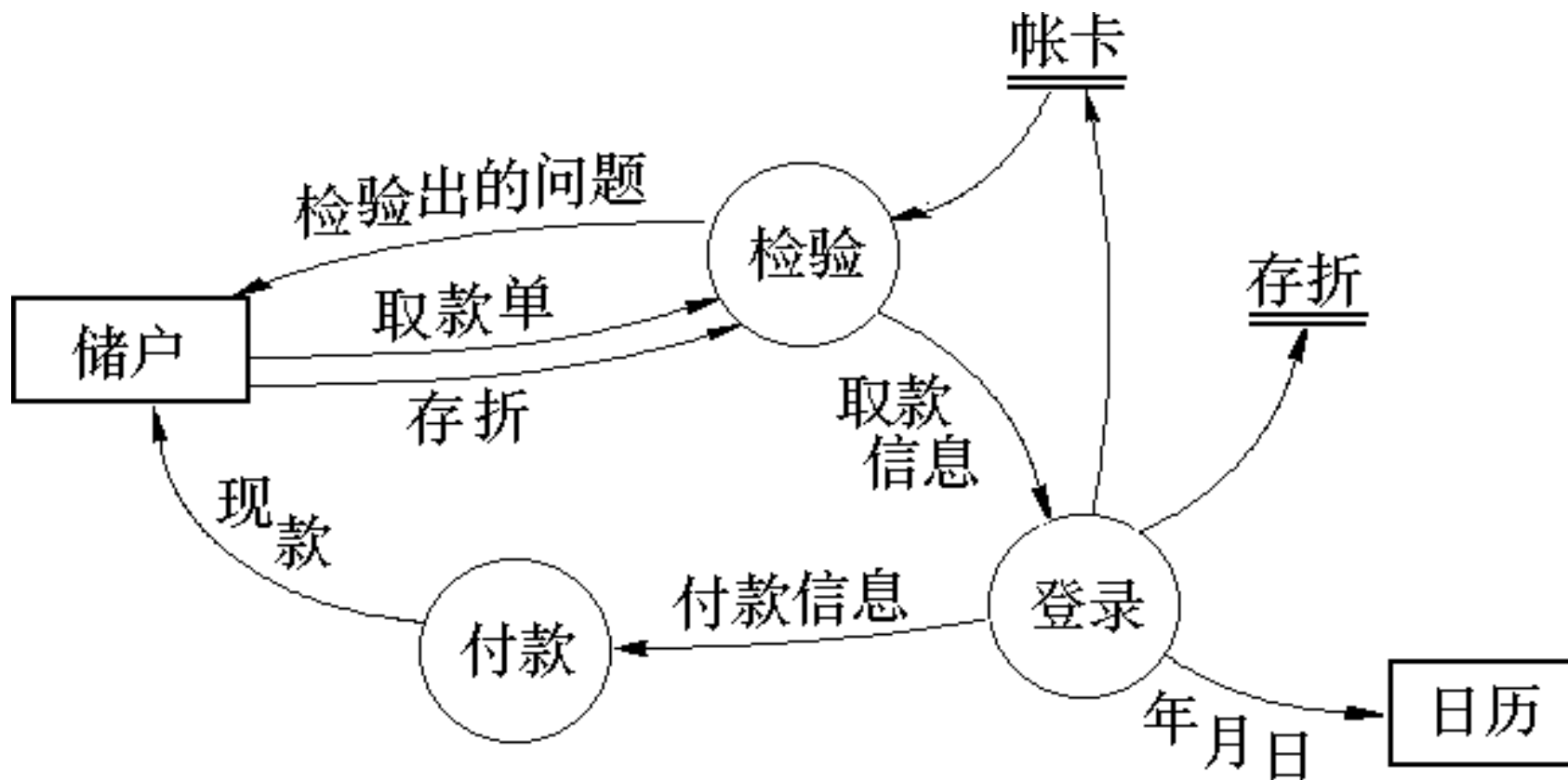
- 20世纪70年发展起来的面向数据流的方法
- 是一种自顶向下逐步求精的分析方法
- 根据软件内部数据传递、变换的关系进行分析的

# 基于数据流的技术

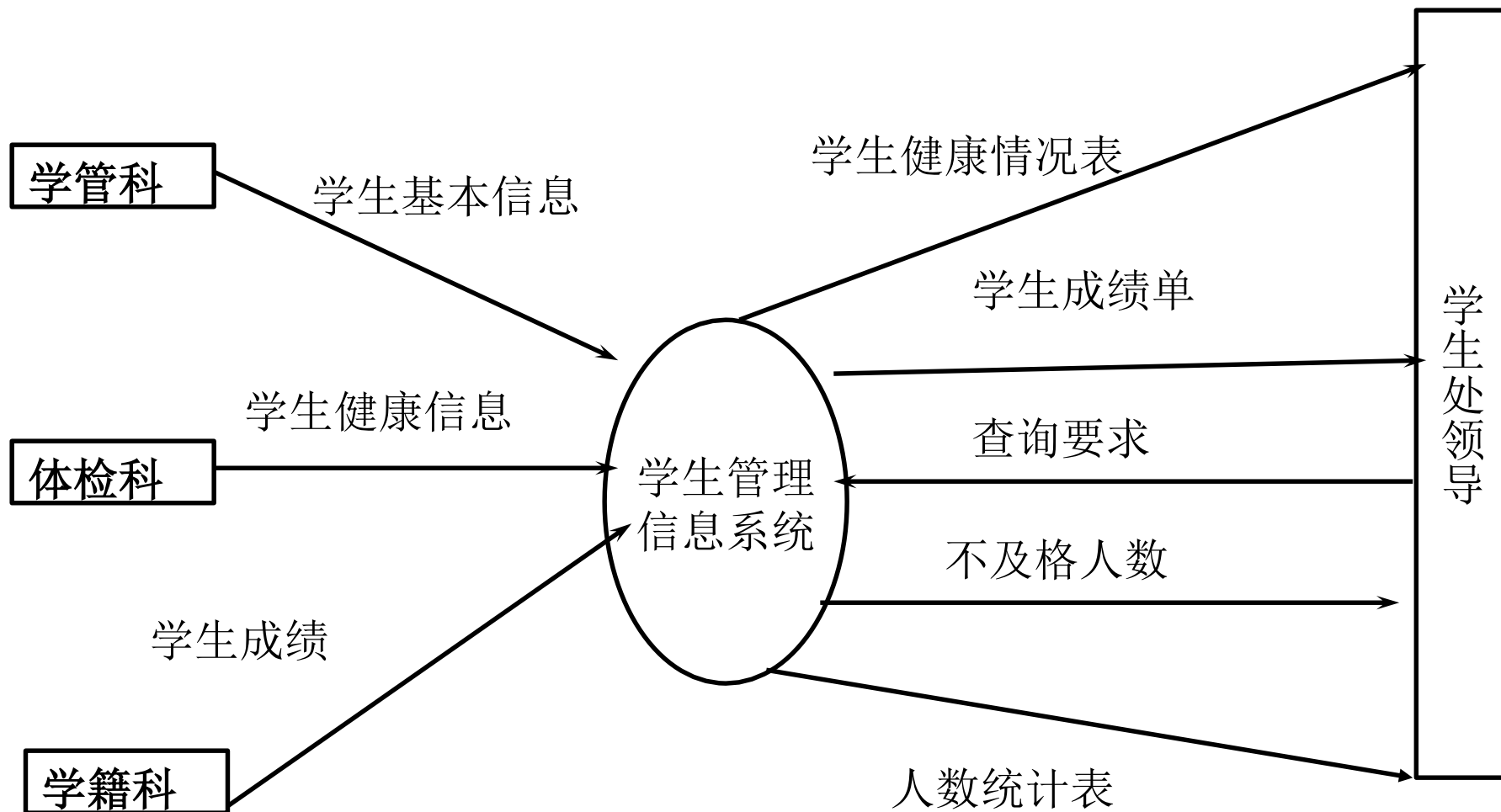
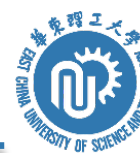


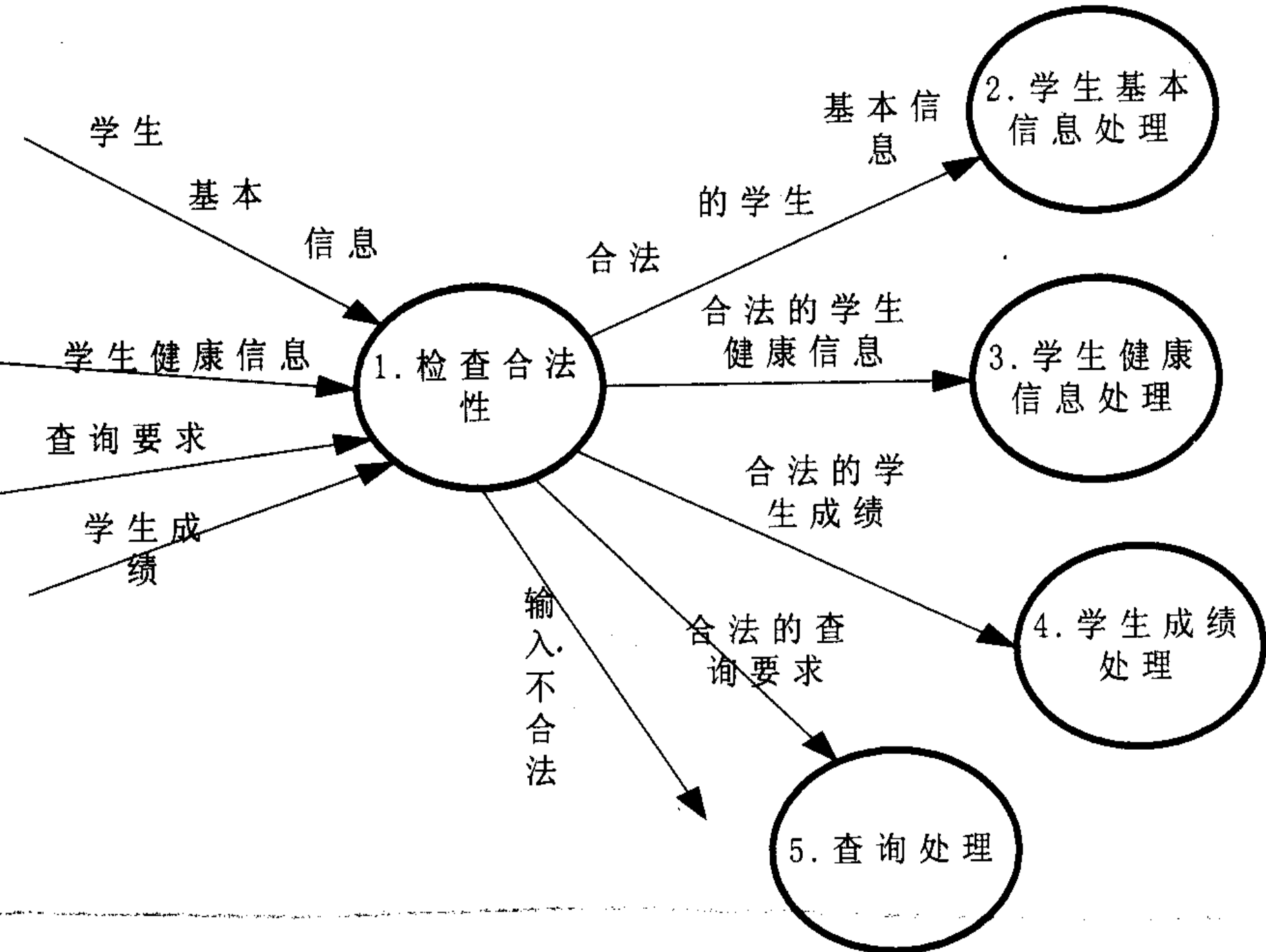
- 数据流图 (DFD)
- 数据字典 (DD)
- 系统流程图

# 描述银行取款过程的数据流图

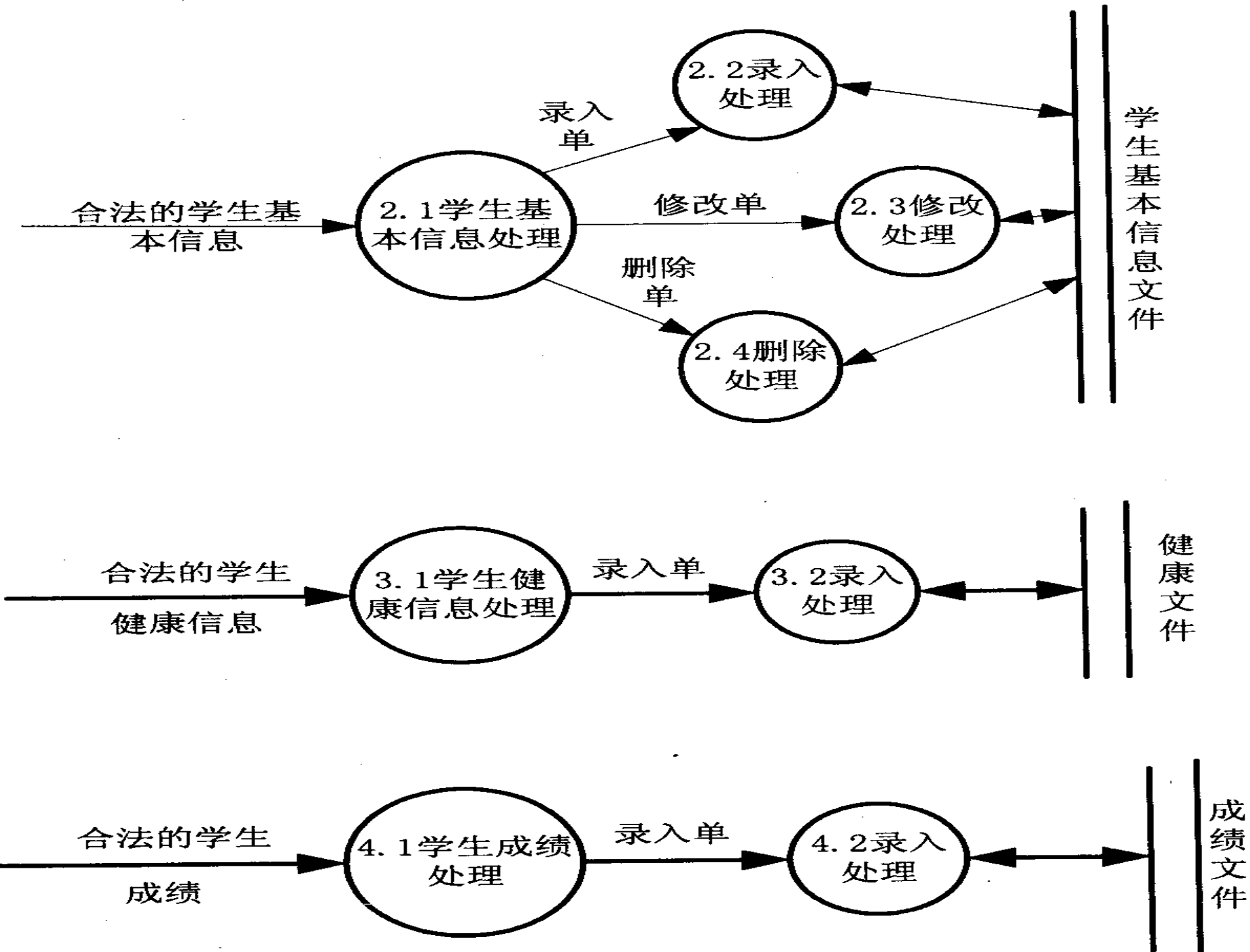


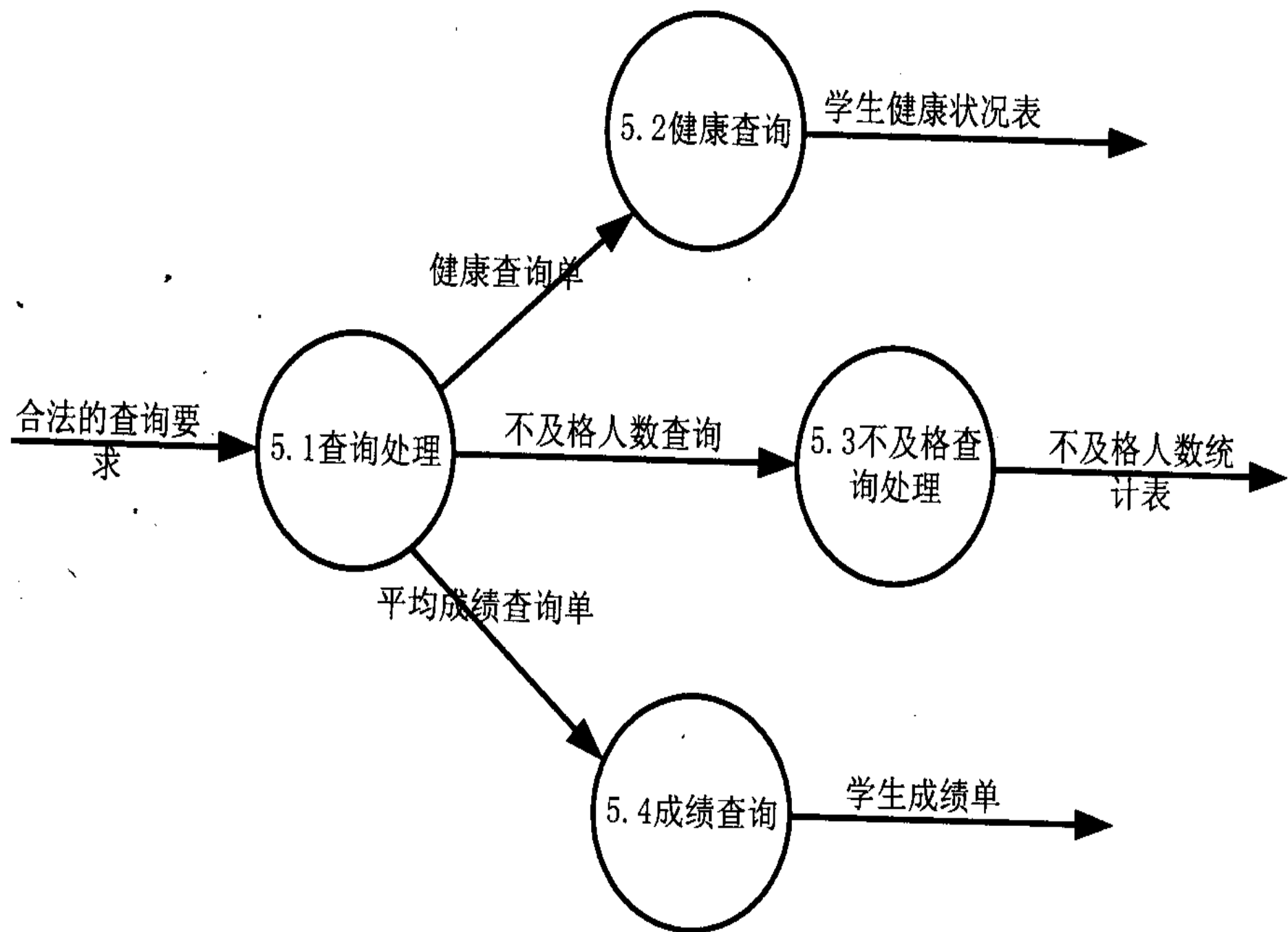
# 学生管理系统-数据流图-顶层











# 学生管理系统-数据字典-数据流



学生基本信息：学号十姓名

学生健康信息：学号十健康情况

学生成绩：学号十{课程名+成绩}

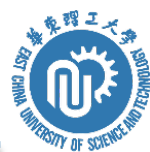
查询要求：[健康查询单 | 平均成绩查询单 | 不及格人数查询]

学生健康情况表：优%十良%十一般%十差%

学生成绩单：学号十姓名十{课程名+成绩}+总成绩

不及格人数统计表：学号十成绩十不及格总人数

# 传统需求建模的基本方法介绍



1. 原型方法
2. 基于数据流建模
3. 基于UML建模

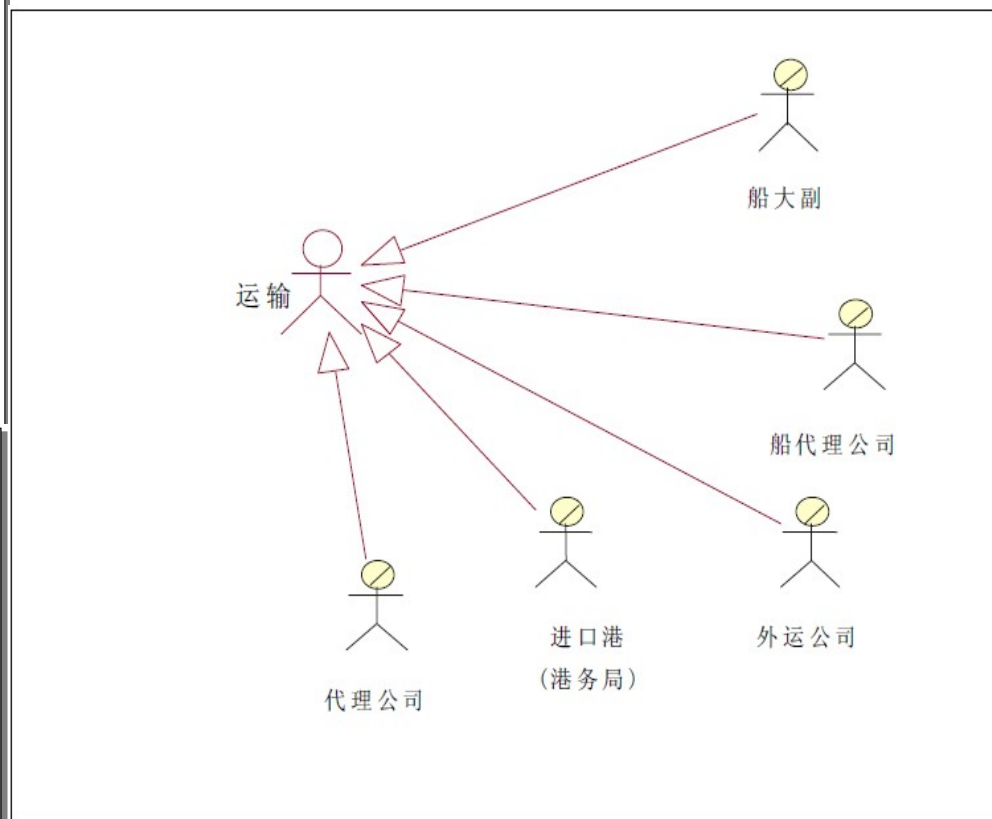
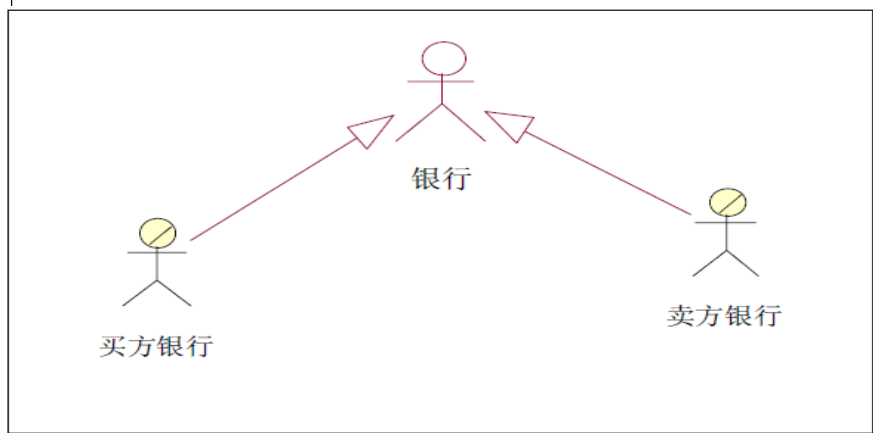
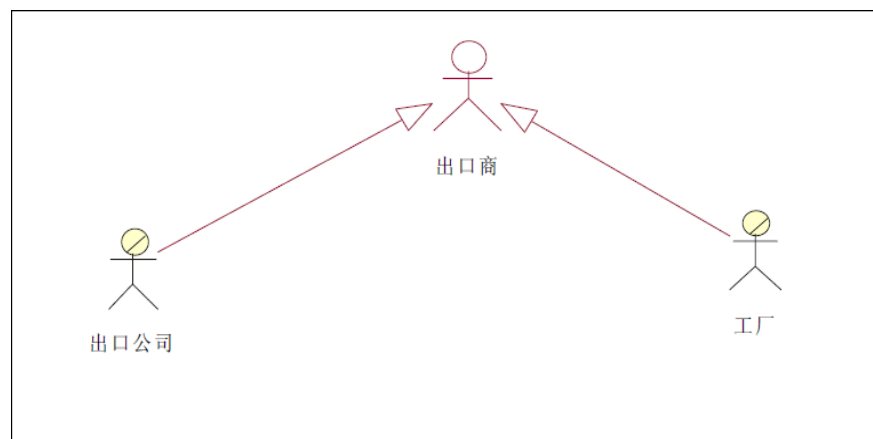
- 基于面向对象的情景分析方法
- 从用户角度出发考虑的功能需求
- 用例是系统向用户提供一个有价值的结果的某项功能

- 用例视图 (Use case Diagram)
- 顺序图 (Sequence Diagram)
- 状态图 (State Diagram)
- 活动图 (Activity Diagram)

- 识别出系统的Actor
- 描述需要的Use case
- 实现用例视图
- 实现顺序视图, 活动视图, 状态视图等

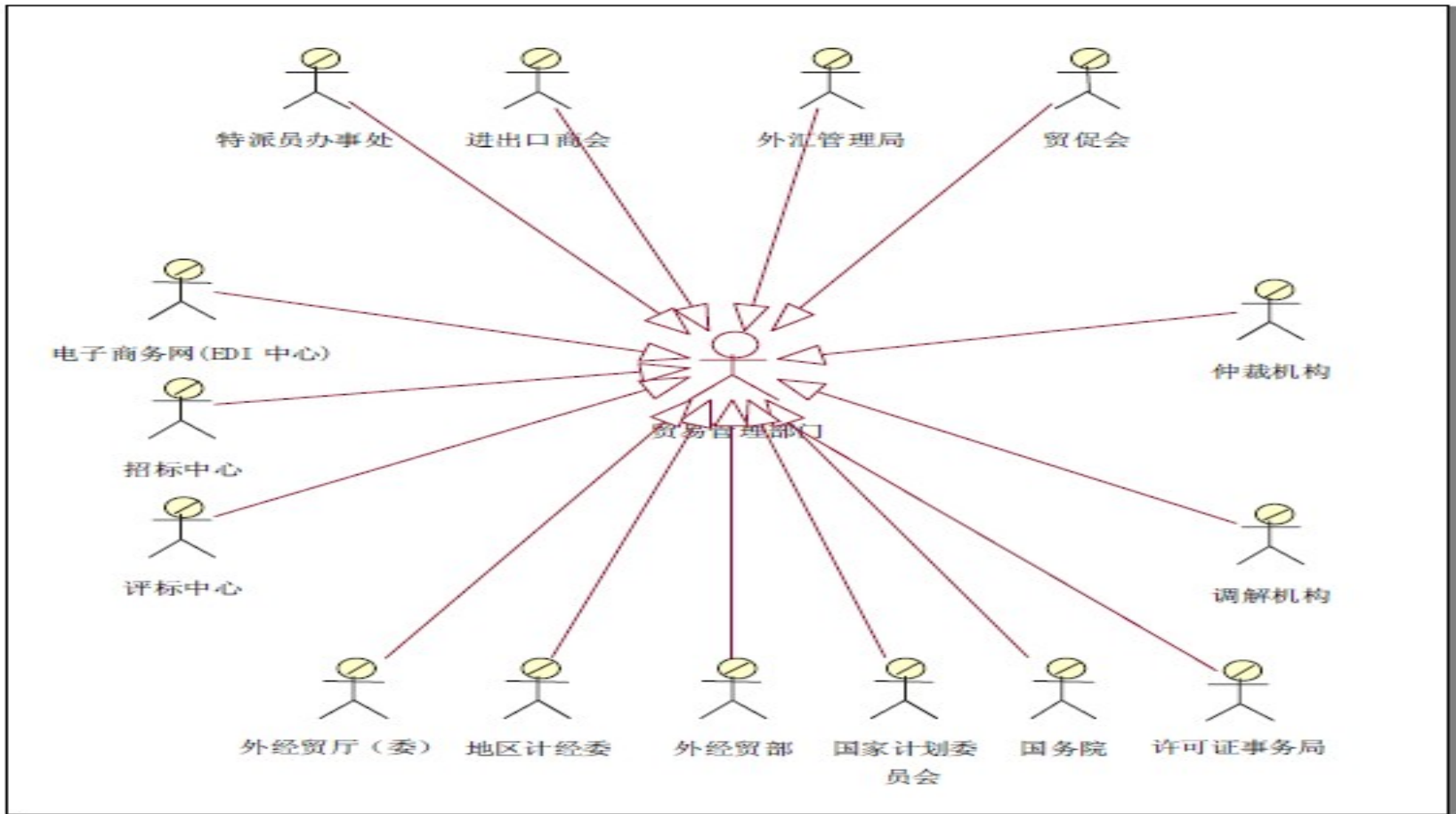
# 贸易链项目需求实例:角色

保险、出口商、存储、海关、检查部门、进口商、贸易管理部门、税务、银行、运输





# 贸易链项目需求实例:角色



## 5. 进出口贸易链主Use Case

进出口贸易的业务环节很多，涉及到配额与许可申请、询价、报价、合同洽谈、备货（出口）、信用证、商检、报关、运输、投保、付汇/结汇、核销退税（出口）等多个环节，并分别隶属于外经贸、内贸、生产部门、海关、商检、银行、税务、保险、运输等职能机构和业务主管部门。这种跨部门、跨单位的“物流”过程，同样伴随着十分复杂的“信息流”。在实际业务中，不同的交易，不同的交易条件，其业务内容和处理过程也不尽相同。在具体运作方面，其贸易处理程序及产生的单证交换，又常常是“并发”与“顺序”交叉进行。贸易程序的简化一直是贸易效率化和低成本交易的“瓶颈”。

进出口贸易按阶段可分为如下两个阶段：

- 合同签订阶段
- 合同执行阶段

详细的描述见本章的其它小节。

## 5.1 出口贸易链主Use Case

### 5.1.1 合同签订阶段

此阶段主要涉及国家对出口商品监管的要求，出口企业需要申请出口商品配额、办理许可证、选定客户并建立业务关系和洽谈成交等，本文主要分析出口商品的配额申请和洽谈成交过程。

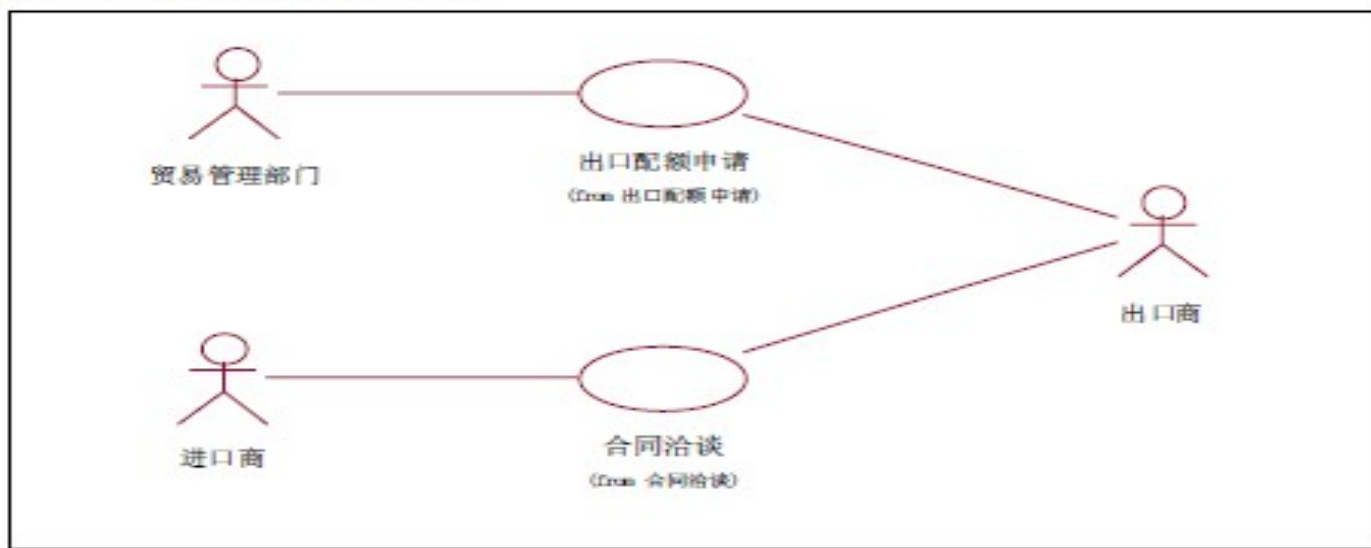


图 5-1 出口贸易链主 Use Case（合同签订阶段）

# 贸易链需求实例：合同执行阶44 段



图 5-2 出口贸易链主 Use Case 图(合同执行阶段)

## 6. 出口贸易链

### 6.1 出口配额

出口配额申请主要描述以下两种 UseCase:

- 计划分配配额（一般配额）
- 招标配额

其 Use Case 图如下所示:

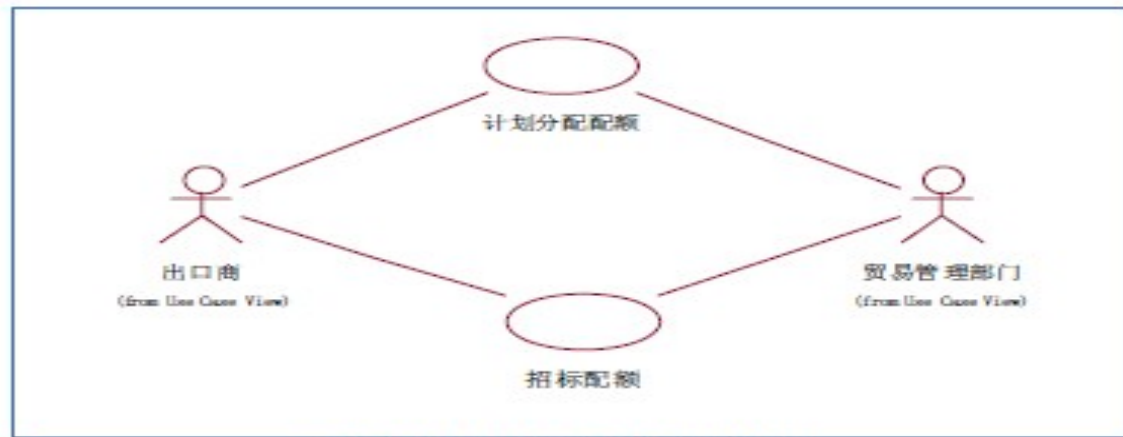


图 6-1 出口配额申请 Use Case

# 贸易链需求实例

Sequence 图

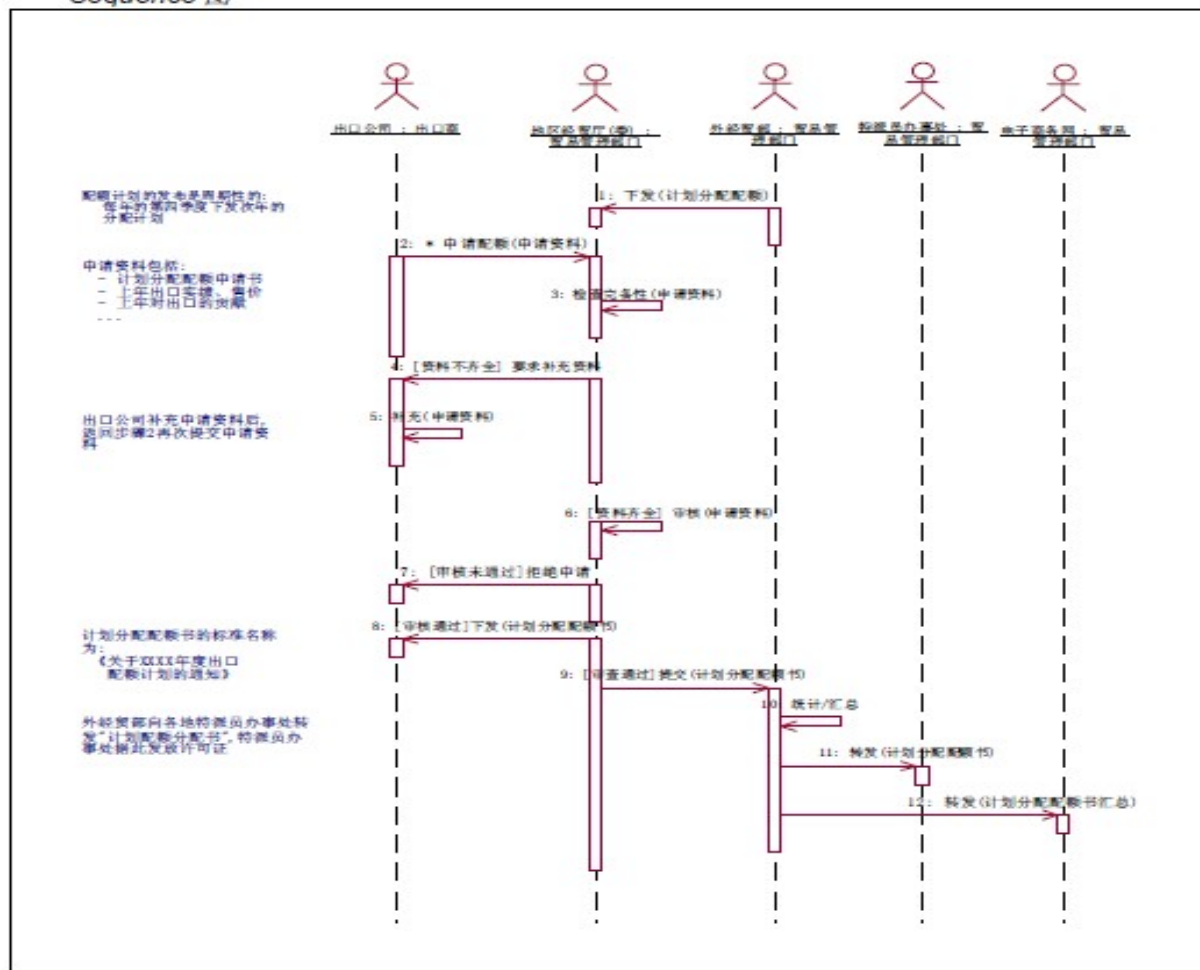


图 6-2 计划分配配额申请 Sequence 图



# 贸易链需求实例

Activity 图

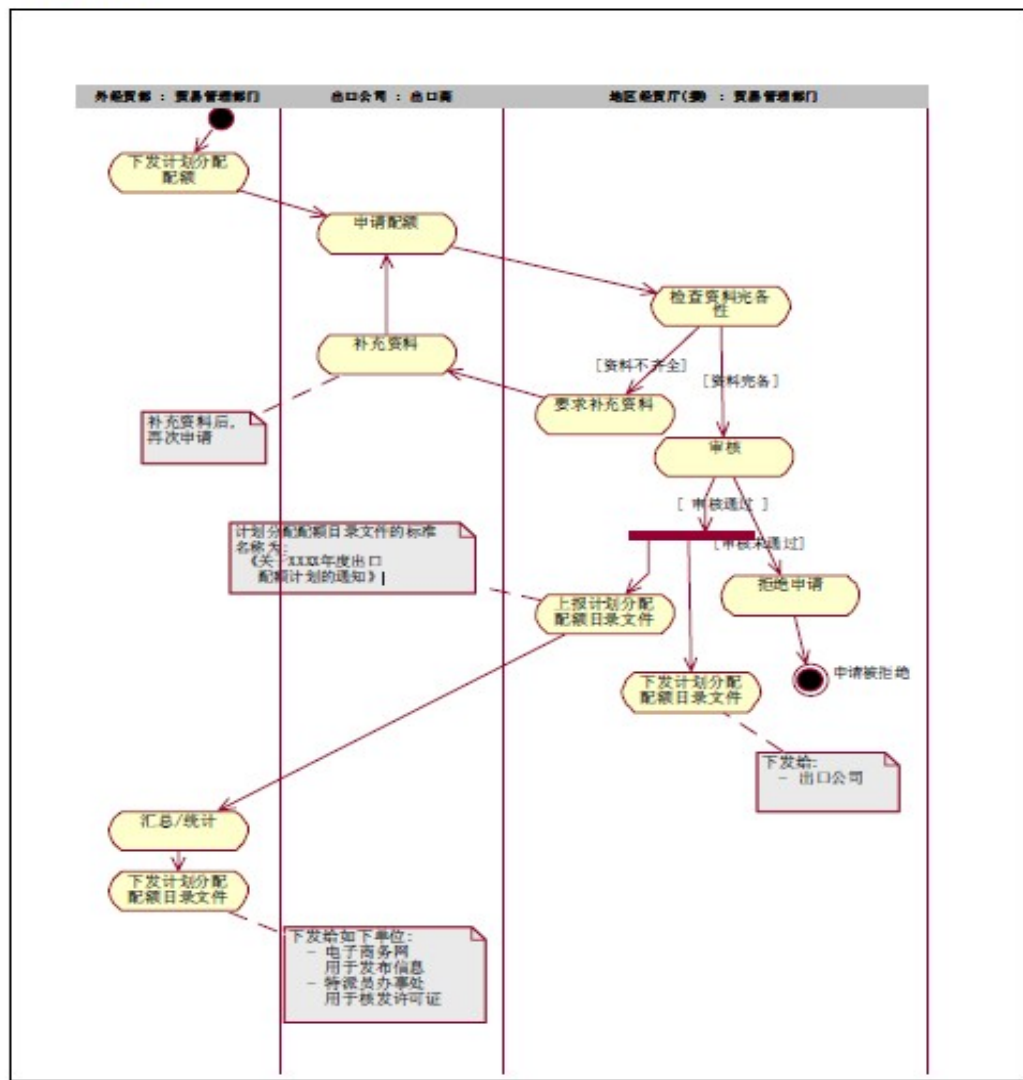
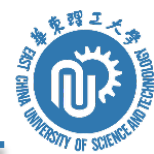


图 6-3 计划分配配额申请 Activity 图

# 本章要点



一

软件需求管理过程

二

传统需求建模方法

三

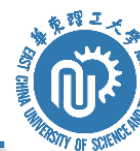
敏捷需求建模方法

四

案例分析



# 需求建模的基本方法介绍



传统方法:

1. 原型方法
2. 基于数据流建模
3. 基于UML建模

1. 敏捷方法

# Requirements in Agile process

User Story应该

1



2

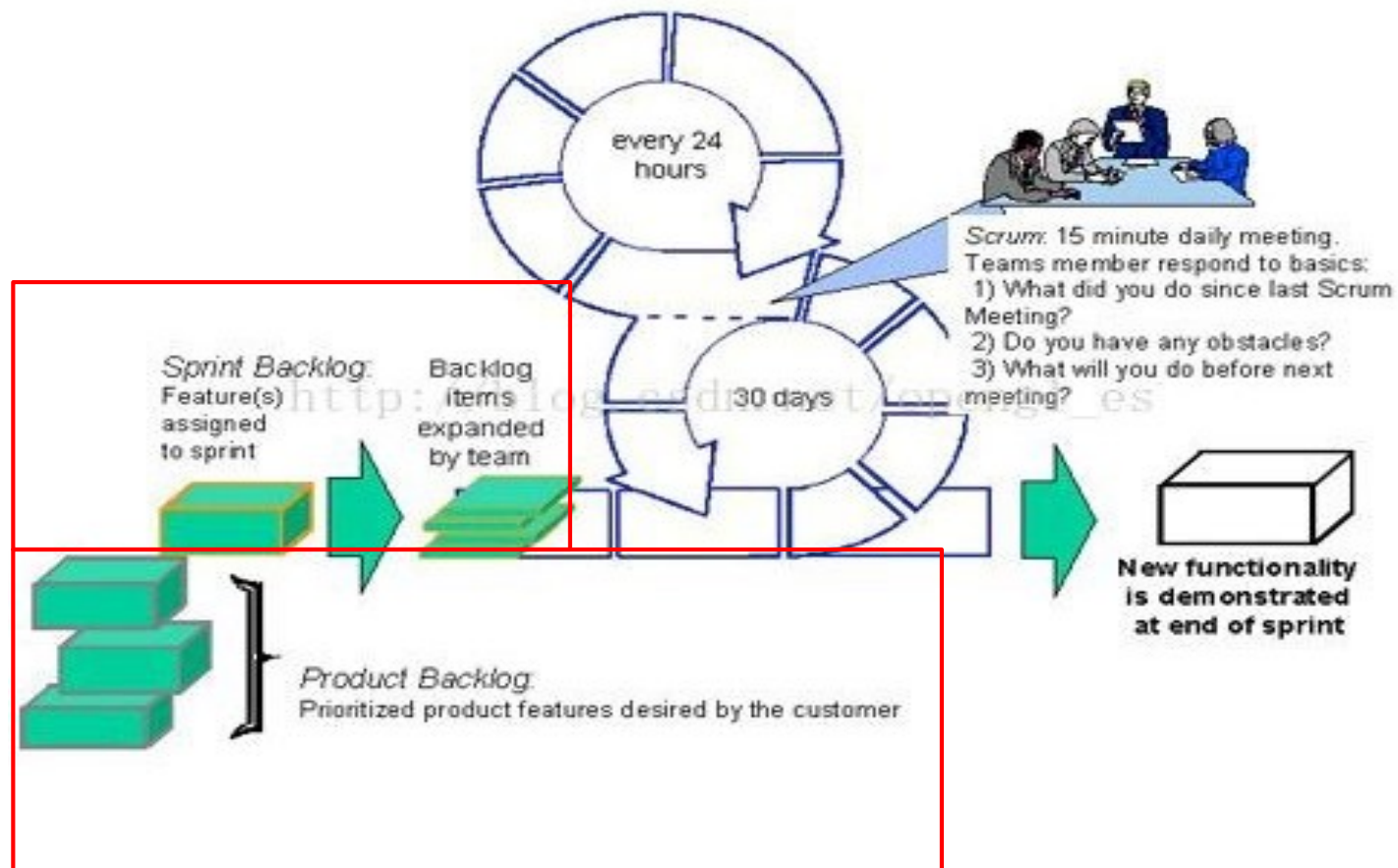


3



渐近明晰

# Backlog – 敏捷需求来源



- 需求的来源
- 包含产品想法的一个有序列表
- 一个长短不定列表
- 可以是模糊的或是不具体的
- 逐渐完善，越来越明确

- 按照迭代计划,逐步细化需求,形成Story(故事),
- 鼓励开发人员、测试人员、业务分析人员和产品负责人合作编写故事,
- 确保所有的故事都足够小,可以持续交付工作。
- 最好每天完成至少一个故事。

# User Stories

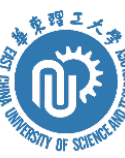


- In Agile process, user requirements are expressed as user stories.

**User stories** are short, simple description of a feature  
They typically follow a simple template:

- As a *<type of user>*,
- I want *<some goal>*
- so that *<some reason>*.

# User Stories



- **User stories** examples:

As a user,  
I want to backup my entire hard drive  
so that I won't lose any work.

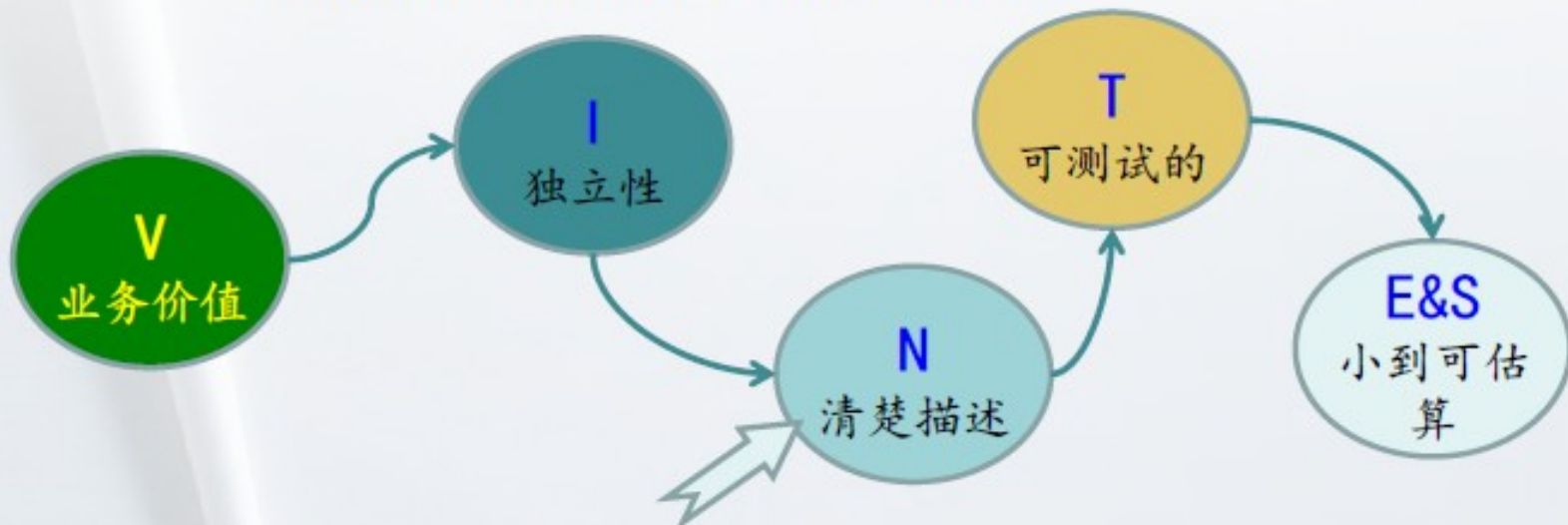


# Good user story?

## INVEST

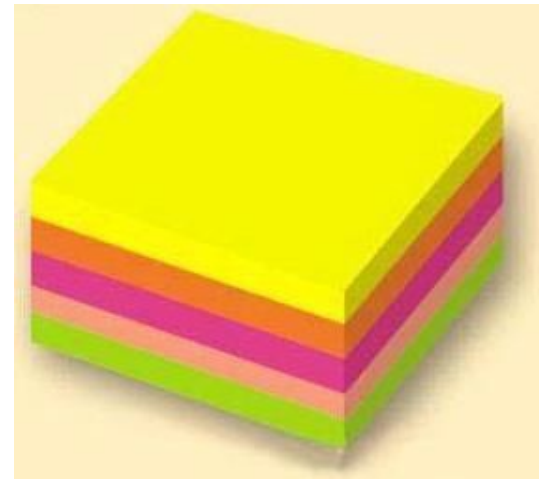
*Given <condition>, When <Time/Event>, Then <Results>*

- 六个特性: INVEST - Independent, Negotiable, Valuable, Estimable, Small, Testable



# Story cards and story wall

- User stories are often written on index cards or sticky notes, stored in a shoe box, and arranged on walls or tables to facilitate planning and discussion.
- As such, they strongly shift the focus from writing about features to discussing them. In fact, these discussions are more important than whatever text is written.



# Story cards and story wall

## Story cards and story wall



Daily team stand-up meeting

Story sorting



# Story wall

## Story Wall



# Story card template (front)

<b>Story name</b>		<b>Story ID</b>	
<b>As a</b>  <b>I want</b>  <b>So that</b>			
<b>Priority</b>	<b>high 1 2 3 4 5 low</b>	<b>Iteration number</b>	
<b>Date Started</b>		<b>Date Finished</b>	

# Story card template (back)

**Acceptance Criteria**

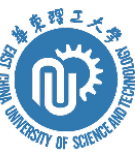
**Notes**



# Non-functional Requirements as User Stories

- Non-functional requirements (constraints) can also be handled as user stories. Examples:
  - As a customer, I want to be able to run your product on all versions of Windows from Windows XP on, so that we can make use of older PCs.
  - As the CTO, I want the system to use our existing orders database rather than create a new one, so that we don't have one more database to maintain. 兼容性---运行环境
  - As a user, I want the site to be available 99% of the time I try to access it, so that I don't get frustrated and find another site to use. 鲁棒性-健壮性
  - As someone who speaks Chinese, I want to have a Chinese version of the software. 兼容性—语言
- If you can't find a way to word the constraint, just write the constraint in whatever way feels natural.

# Prioritisation of stories



User stories must be prioritised.

- –Based on business value
- –Value must also be supported by a positive return on investment (ROI)
- –Consideration of the risks

The **customer** chooses the stories for inclusion in the next release based on their priorities and the schedule estimates.



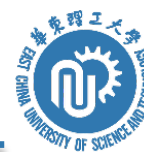
# MoSCoW

- A method of prioritisation favoured by the DSDM (dynamic systems development method). Elements are:
  - **Must have**: features must be implemented, and if not, the system would not work.
  - **Should have**: features are important but can be omitted if time or resources constraints appear.
  - **Could have**: features enhance the system with greater functionality, but the timelines of their delivery is not critical.
  - **Want to have**: features serve only a limited group of users and do not drive the same amount of business value as the preceding items.

Payment processing story:

- **–Must have:** ability to accept Visa and MasterCard.
- **–Should have:** add American Express.
- **–Could have:** add PayPal.
- **–Want to have:** add gift cards.

# 本章要点



一

软件需求管理过程

二

传统需求建模方法

三

敏捷需求建模方法

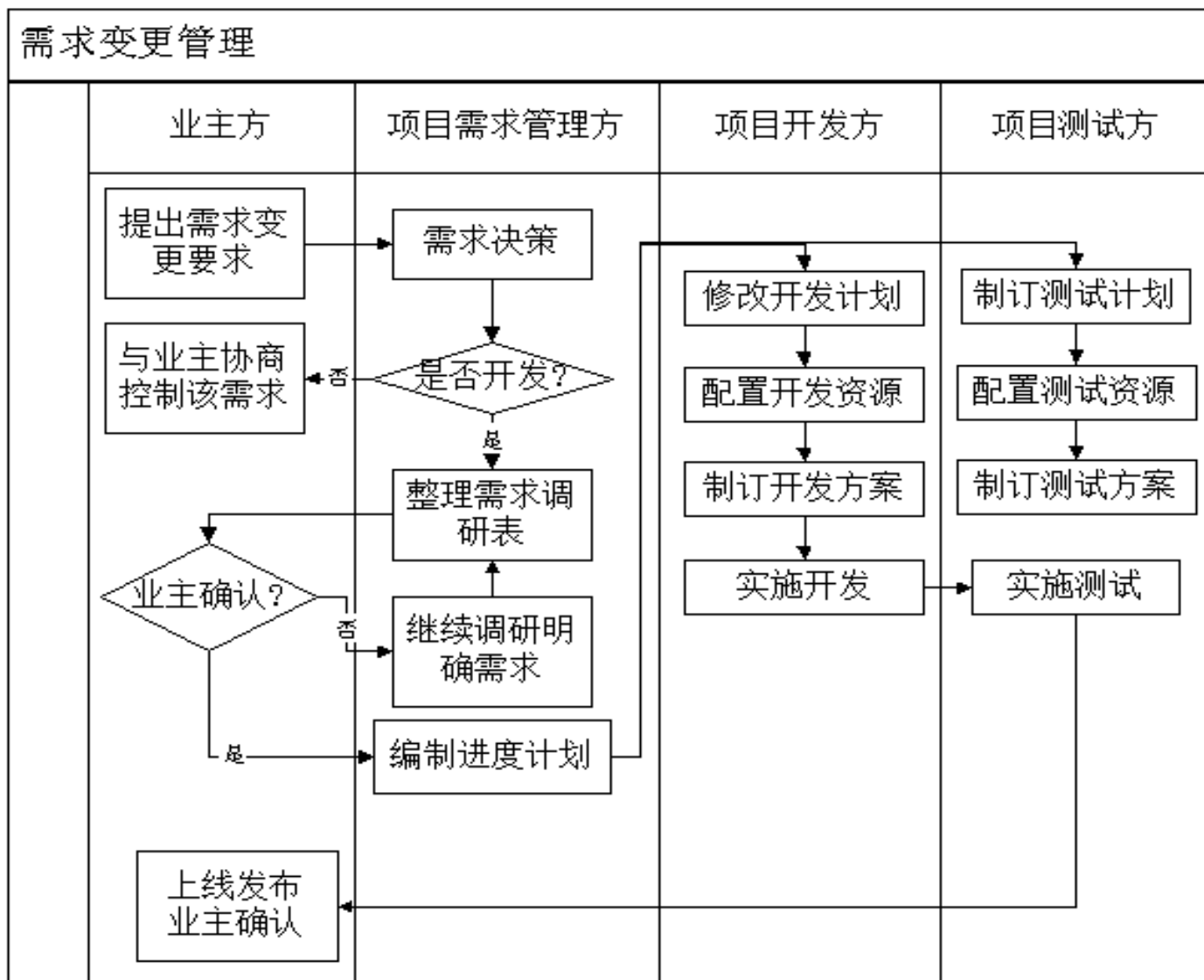
四

案例分析

## MED需求管理案例

- 需求确认: 软件需求规格
- 需求变更: 变更控制流程

# 变更控制流程



## 软件需求管理过程

- 需求获取
- 需求分析
- 需求规格编写
- 需求验证
- 需求变更

## 需求建模的基本方法

- 原型方法
- 基于数据流建模-结构化分析法
- 基于UML建模-面向对象的用例分析法
- 敏捷需求分析