

## 软件工程与实践

软件工程与实践课程组  
电子科技大学信息与软件工程学院

## □第三章 需求分析

- ✓ 第一部分 软件需求工程基础
- ✓ 第二部分 获取软件需求
- ✓ 第三部分 分析软件需求



# 第二部分 获得软件需求

## 1. 获得软件需求的方法

- ✓ 软件需求的方式和过程

## 2. 明确问题及软件解决方案

- ✓ Problem → Software Solution

## 3. 导出和构思软件需求

- ✓ 如何从利益相关方导出和构思软件需求

## 4. 描述初步的软件需求

- ✓ 自然语言描述和可视化建模

## 5. 确认和验证初步软件需求



# 1.1 何为获取软件需求?

- 软件需求不会无缘无故的产生，也不应是无源之水、无本之木，任何软件需求都有其出处以及动机，即它从何而来、目的是什么
- 获取软件需求是指得到或产生软件需求
- 获取软件需求是需求分析的**基础**，是软件开发的**首要工作**

# 软件需求从何而来？

## 口软件开发的动机

- ✓ 解决特定行业和领域的问题，体现意义和价值
- ✓ 示例：“MiNote”是为了帮助用户编写和管理便签；“12306”是为了帮助旅客解决购买火车票费时费力的问题；“ElderCarer”是为了解决空巢老人在家无人看护的问题。

## 口解决软件利益相关方的诉求

- ✓ 软件系统的受益者，软件需求的提出者
- ✓ 利益相关方有哪些，他们会对软件提出什么样的期望和要求
- ✓ 示例：“12306”利益相关者是旅客，服务于旅客期望和要求

# 软件开发者也可以充当软件的利益相关方

## □许多软件系统的需求来自**软件工程师**，而非最终的用户

- ✓示例：如开源软件，微信软件，12306软件等等

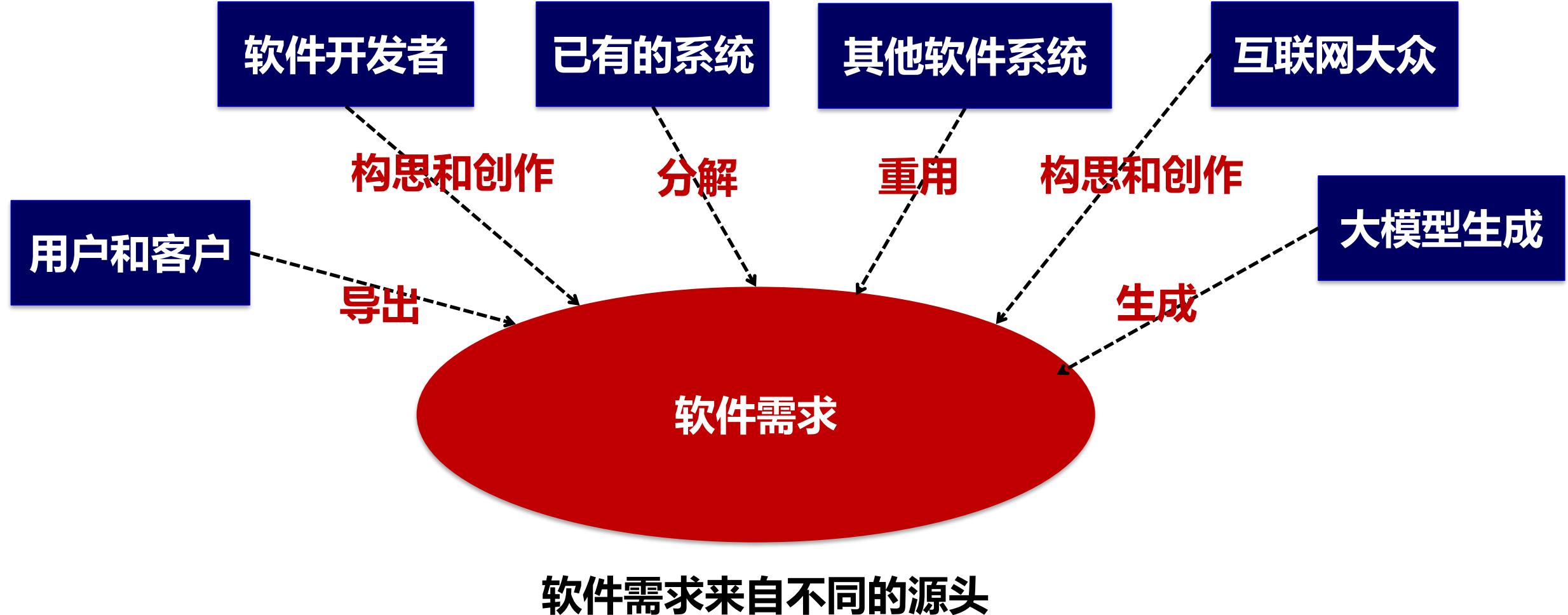
## □他们充当软件的**用户或客户**，构思和提出软件需求

- ✓原因是找不到实际的用户来提出需求

## □开源软件的许多需求来自**软件开发者**，而非最终用户

- ✓提出软件的功能性构思及需求，并以Issues的形式提出

# 1.2 获取软件需求的方式



# 思考和讨论

- 微信和“12306”软件的需求从何而来？
- 开源软件的需求从何而来？
- 飞机控制软件系统的需求从何而来？



# 要获取的软件需求要素

- **Who**, 谁会关心该项软件需求，他们有何特点和诉求
- **What**, 软件需求的内涵是什么
- **Why**, 为什么需要软件需求，想解决什么样的问题
- **Where**, 软件需求归属于哪些子系统
- **How**, 软件需求包含哪些行为，它们是如何来解决问题的
- **When**, 什么时候需要该项软件需求

要说清楚一项软件需求，就需要说清楚有关需求的以上6个方面内容

# 获取软件需求面临的困难

## □ 需求想不清

- ✓ 软件需求并非简单地实现应用领域中的业务流程，而是要**将业务流程进行改造，形成基于软件的解决方案**
- ✓ 软件的利益相关方虽然对业务流程非常熟悉，但是要对其进行改造、形成软件需求实则较为困难

## □ 需求道不明

- ✓ 不清楚软件需求的具体内涵，很难清晰、准确和详实地讲明白软件需求是什么，从而无法获取足够多的软件需求信息
- ✓ **软件需求不明确，需求内容不详实，软件需求质量低等**

## □ 需求想不到

- ✓ 一些软件系统是全新的，在现实世界没有可模仿和参照的对象。软件利益相关方和需求工程师要构思出软件需求将变得更加困难

# 1.3 获取软件需求方法

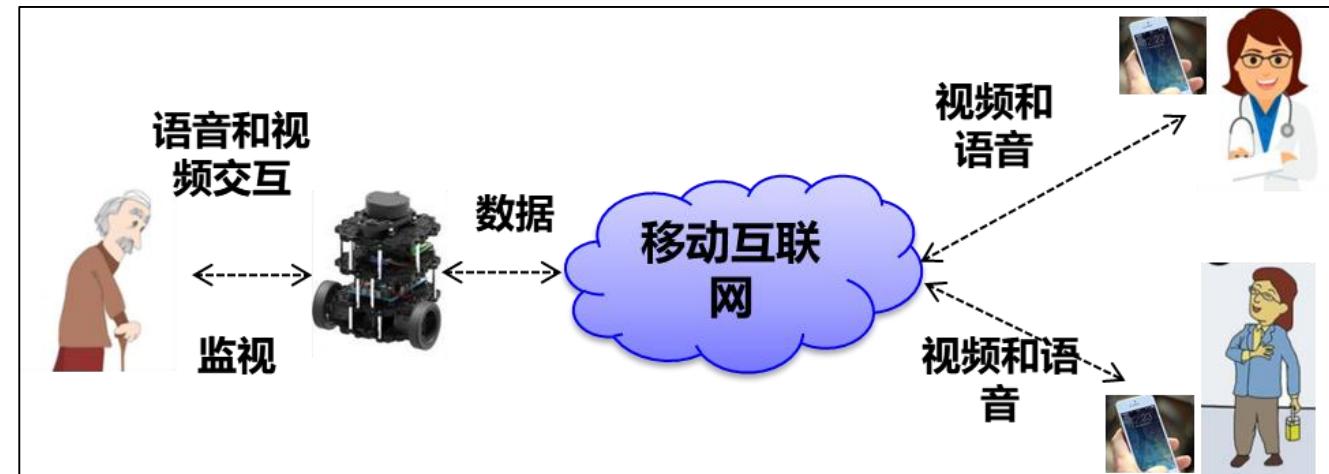
- 访谈和会议(Interview)**
- 调查问卷(Questionnaire)**
- 现场观摩**
- 分析业务资料**
- 软件原型**
- 群体化方法**
- 大脑风暴构思 (Brainstorming)**
- 大模型生成 .....**

# 访谈和会议

- 召开相关的会议
- 走访相关的人员



参加的人员包括：用户、客户、需求分析人员、质量保证人员等等



从访谈和参与会议的对象中获得软件需求

# 调查问卷

## □设计调查问卷

- ✓了解业务流程、实际操作过程
- ✓用户的期望和要求

## □对谁进行调查

- ✓应用领域中扮演不同角色的用户

## □如何利用调查问卷

- ✓从调查问卷中了解业务流程
- ✓从调查问卷中掌握用户需求

如设计一个问卷，调查车站旅客对12306软件的期望和要求



# 示例：通过调查问卷来获取软件需求

□设计调查问卷，询问旅客对12306软件需求及其优先级

- ✓ 需要哪些功能
- ✓ 哪些功能优先

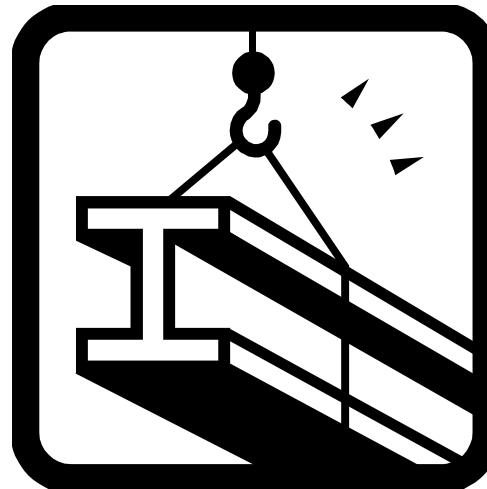
□在火车站寻找旅客开展问卷调查，根据调查的结果确定软件需求及其优先级



# 现场观摩

- 业务的过程、步骤和输出
- 业务的工作流程及细节

如观摩售票窗口的购票过程



从现场观摩中了解业务流程，进而获得软件需求

# 分析业务资料

- 收集重要的业务资料
- 阅读和分析业务资料

如银行国际交易的业务材料，火车调度的业务资料



通过阅读业务资料来了解业务及其流程，进而获得软件需求

# 大脑风暴

- 将与软件系统相关的一帮人组织在一起，非正式、开放地、甚至没有明确主题的散漫讨论，从中**捕捉软件需求的灵感和认识**
- 该方法适合于那些需要对软件需求进行**开放构思和自由创作**的软件系统



# 软件原型

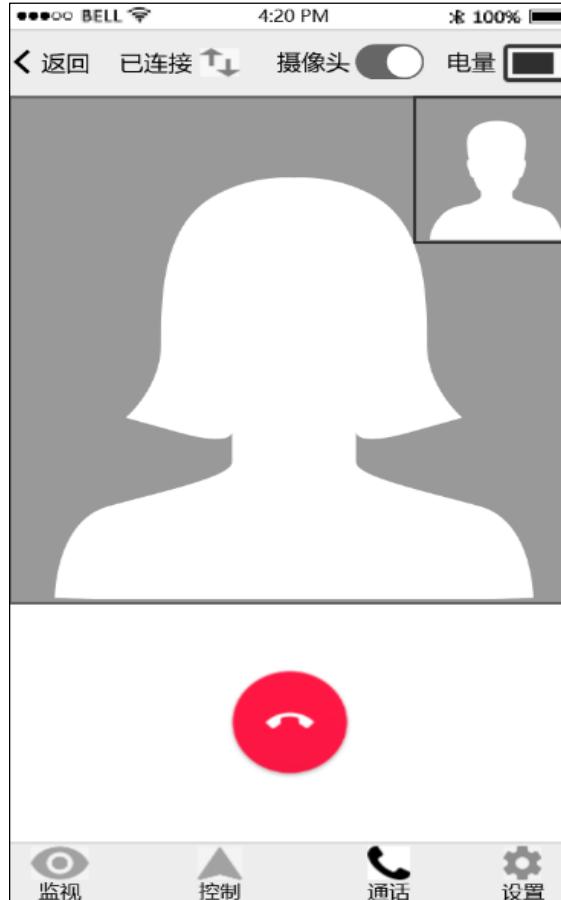
□需求工程师根据用户初步需求描述，快速构造出一个可运行的软件原型

- ✓ 展示业务操作流程以及每一个步骤用户与软件之间的交互
- ✓ 用户可以通过操作和使用该软件，分析需求工程师是否正确地理解了他们所提出的软件需求，发现软件原型所展示的软件需求中存在的问题，导出尚未发现、新的软件需求

□软件原型作为需求工程师和用户之间的交流媒介，有助于直观地展示软件需求，激发用户投入到需求讨论和导出之中，因而是一项极为有效的需求获取和分析方法

# 示例：基于软件原型来获取软件需求

## □ “空巢老人看护软件”的原型



- 让用户使用该软件原型
- 听取用户对原型的意见和建议
- 用户意见就是潜在的软件需求

# 群体化方法

- 让互联网上的**海量和开放群体**参与到获取软件需求的工作中来，提出他们对软件需求的想法，并通过组织、汇聚和筛选，从中遴选出有价值的软件需求
- 集思广益，吸纳更多的人参与到需求创作和构思之中，有助于获得超出开发团队和用户常规想定的软件需求
- 开源软件的需求就是采用群体化的方法来获取的

# 借助大模型技术

- 通过大模型工具(如豆包, deepseek, ChatGPT)生成软件需求
- 给大模型提示 (Prompt)
- 理解和遴选生成的软件需求

# 成立需求分析的联合工作小组

## □ 小组成员

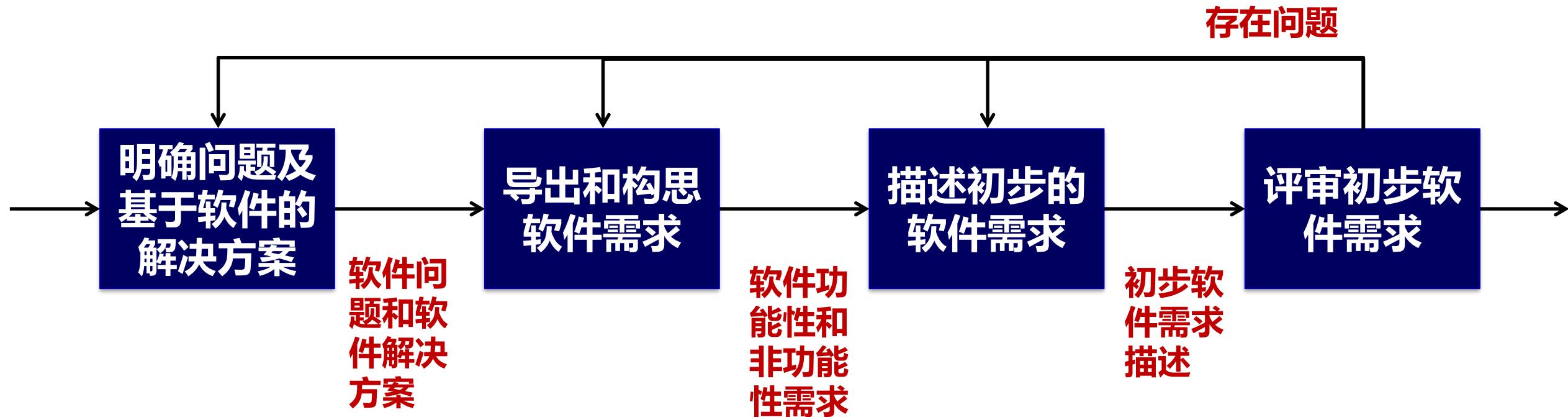
- ✓ 利益相关者 (如用户)
- ✓ 软件需求分析人员
- ✓ 软件质量保证人员等

## □ 目的

- ✓ 共同工作
- ✓ 加强联系
- ✓ 促进交流
- ✓ 增进合作



# 1.4 获取软件需求的过程



# 内容

## 1. 获取软件需求的方法

- ✓ 软件需求的方式和过程

## 2. 明确问题及软件解决方案

- ✓ Problem → Software Solution

## 3. 导出和构思软件需求

- ✓ 如何从利益相关方导出和构思软件需求

## 4. 描述初步的软件需求

- ✓ 自然语言描述和可视化建模

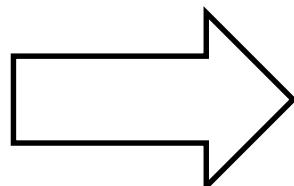
## 5. 确认和验证初步软件需求



## 2.1 软件的目的

- 每一个软件都试图去解决特定领域中的问题，并提供基于软件的问题解决方案
- 软件需求必须服从和服务于软件欲解决的问题，只有这样软件需求才有意义和价值

12306软件



买票难、买票费时  
等问题

# 明确软件要解决的问题

## □清晰地界定软件欲解决什么样的问题

- ✓ 与特定领域及其业务相关联
- ✓ 或提高业务工作效率，或解决业务瓶颈问题，或提升业务服务水平和质量等等

## □示例

- ✓ “12306” 软件与铁路旅客服务这一领域及业务相关联，国家铁路集团有限公司投资研制该软件的目的是要改变落后的旅客服务和业务模式，提高旅客服务质量，降低旅客服务成本

# 定义软件欲解决问题

## □开展调研分析，切忌拍脑袋凭空想问题

- ✓ 铁路旅客存在买票难的问题

## □不断反复论证，寻找适合软件解决的问题

- ✓ 通过软件来支持旅客在线上买票

## □寻求有意义、有价值的问题

- ✓ 买票、退票、改签、打印等等

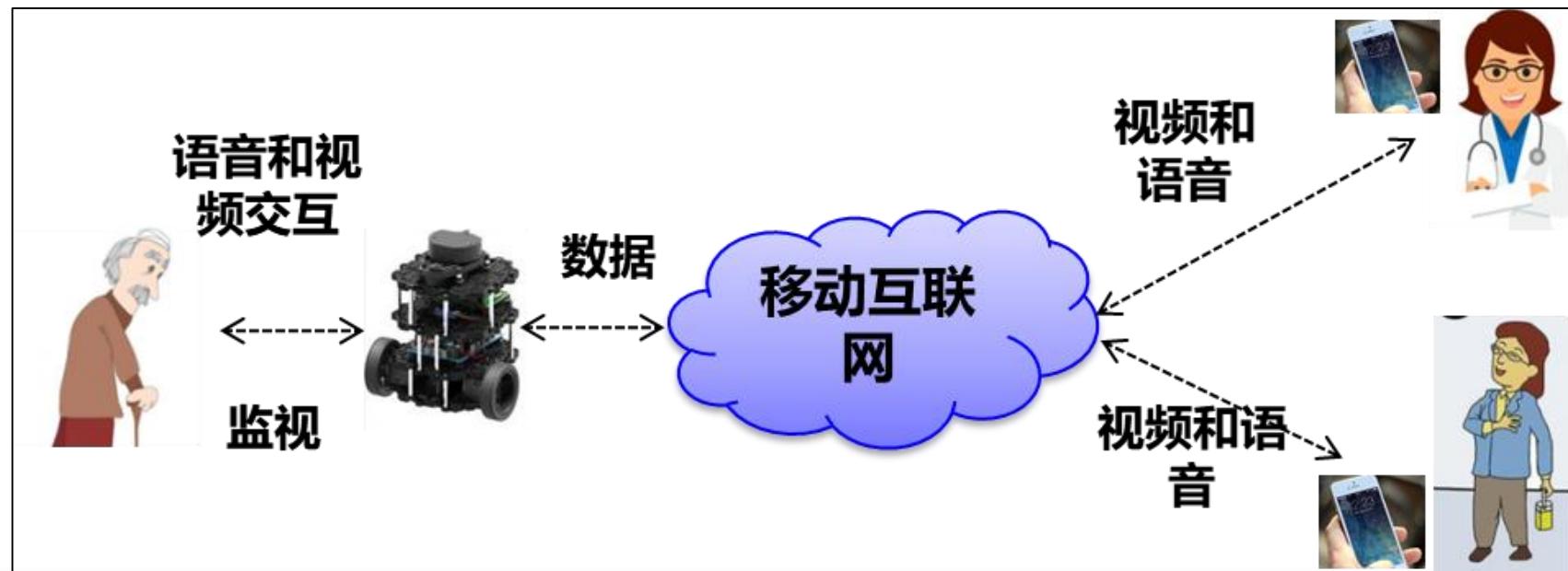


# 明确基于软件的问题解决方案

## □ 软件可以起到什么作用

- ✓ 完成各种计算，作为一种粘合剂来连接不同的设备和系统，实现不同设备和系统之间的交互和协同，从而来解决问题

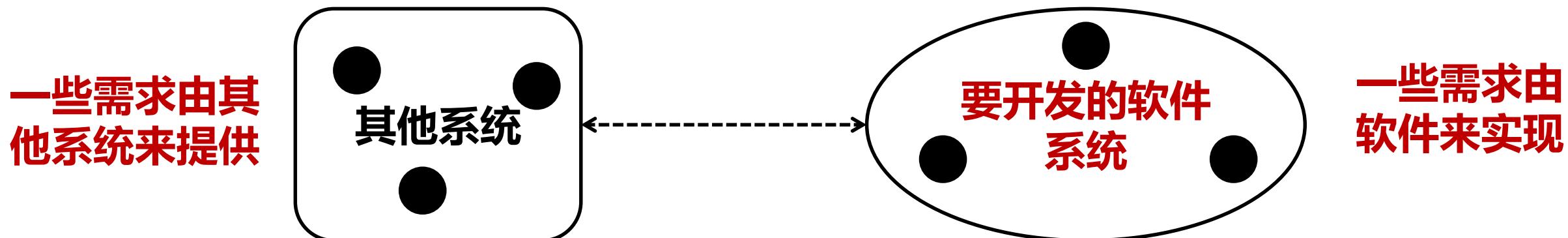
## □ 发挥软件的优势，为问题解决提供新颖和有效的途径



# 明确软件的范围和边界

## □ 描述软件的范围，确定软件的边界

- ✓ 软件的范围说明了软件需要完成哪些业务领域中的功能
- ✓ 软件的边界描述了软件的界限，即哪些要素属于软件，哪些不属于软件；哪些需求要由软件来完成，哪些需求由其他设备和系统来完成



# 思考和讨论

- 微信软件想解决什么样的问题？
- 微信如何借助软件来解决问题？
- “空巢老人看护” 软件的范围和边界是什么？
- 12306软件的范围和边界是什么？



# 内容

## 1. 获取软件需求的方法

✓ 软件需求的方式和过程

## 2. 明确问题及软件解决方案

✓ Problem → Software Solution

## 3. 导出和构思软件需求

✓ 如何从利益相关方导出和构思软件需求

## 4. 描述初步的软件需求

✓ 自然语言描述和可视化建模

## 5. 确认和验证初步软件需求



### 3.1 识别软件的利益相关方

- 软件需求来自于软件的**利益相关方**，要获取软件需求，首先要搞清楚软件系统有哪些**利益相关方**
- 软件系统的利益相关方可以表现为特定的**人群和组织**，也可以是一类**系统**
- 不仅**软件用户或客户**可以是软件的利益相关方，软件的**开发者**也可以成为软件的利益相关方。

# 示例：“空巢老人看护软件”的利益相关方

## □老人 (Elder)

- ✓与系统通过语音方式进行交互，命令系统为其完成某些事务

## □医生 (Doctor)

- ✓突发或紧急情景时接受呼叫，与系统进行交互以获取老人状况

## □家属 (Family Member)

- ✓突发或紧急情景时接受呼叫，与系统进行交互以获取老人的状况

## □管理员 (Administrator)

- ✓对软件系统进行必要的配置和管理

## □机器人 (Robot)

- ✓其运动受软件系统的控制

# 示例：“Mini-12306” 软件的利益相关方

## □旅客(Passenger)

- ✓ 希望提供购票、查询车次、退票、改签等功能和服务

## □售票员(TicketSeller)

- ✓ 希望能帮助旅客完成购票、查询车次、退票、改签等服务

## □系统管理员Administrator)

- ✓ 设置系统的参数等相关信息，如可提前购买多少天的车票等

## 3.2 导出软件的功能性需求

需求工程师可以通过与利益相关方的交互，听取他们对软件的**期望和要求**，从他们那里导出软件需求

采用多种方法来导出软件需求

- ✓ 与用户或客户的面谈
- ✓ 分析业务资料
- ✓ 观察业务流程
- ✓ 进行问卷调查
- ✓ 软件原型等



# 构思软件的功能性需求

- 需求工程师需要**充当软件利益相关方的角色**，站在他们的视角，来**构思软件需求**
- 针对软件需求开展创作，结合软件解决方案，提出可有效促进问题解决的**软件需求**
- 需求工程师可以采用**头脑风暴构思、群体化方法、问卷调查、软件原型**等多种方式来开展需求构思工作

需求工程师需要积极和主动地参与到需求获取的过程中

# 示例：构思“空巢老人看护软件”功能性需求

## □老人的视角

- ✓自主跟随老人，以对老人进行持续跟踪和感知，获取老人信息。
- ✓提醒服务，提醒老人按时服药、身体检查等事宜

## □家属和医生

- ✓监视老人状况，通过智能手机在远端监视老人在家的状况，获得老人的视频、图像和语音等方面的信息
- ✓通知异常情况，将老人的突发异常信息发送给老人家属和医生。
- ✓远程控制机器人，通过智能手机在远端来控制机器人移动，以从不同的角度和距离来获取老人的图像、视频和语音信息
- ✓视频/语音双向交互，实现老人、医生和家属间的视频和语音交互

# 示例：构思“Mini-12306”功能性需求

## □旅客和售票员的视角

- ✓注册和登录
- ✓查询车次
- ✓购票、退票、改签等

## □系统管理员的视角

- ✓设置系统参数
- ✓如设置售票员的账号和密码、设置车票提前预售的天数

### 3.3 导出和构思软件的非功能性需求

#### □ 非功能性需求包括软件质量要求和软件开发的约束性要求

- ✓ **质量要求**，如软件运行性能、可靠性、易用性、安全性、私密性等属于外部质量要求，软件可扩展性、可维护性、可互操作性、可移植性等内部要求
- ✓ **约束性要求**，包括开发进度要求、成本要求、技术选型等

#### □ 软件的非功能性需求变得越来越重要

- ✓ 在某些情况下它们直接决定了软件**是否能用和可用、是否好用和易用、是否高效和可靠运行、是否便于维护和演化**等。

# 示例：“空巢老人看护软件”的非功能需求

类别	非功能性需求项	需求描述
性能	EC-Req-Performance-01	所有界面操作的响应时间应小于1s
可靠性	EC-Req-Reliability-01	软件系统每周7天、每天24小时可用；在机器人和网络无故障的前提下，系统正常运行时间的比例在95%以上
	EC-Req-Reliability-02	系统任何故障都不应导致用户已提交数据的丢失。发生故障后，系统需在10分钟内恢复正常使用
易用性	EC-Req-EasyUse-01	老人只需通过语音方式与系统进行交互；家属、医生和管理人员通过操作手机App来使用本系统
	EC-Req-EasyUse-02	用户无需专门培训只需通读安装手册即可完成安装
安全性	认证需求EC-Req-Safety-Authentication-01	所有用户（包括家属和医生）均需通过用户名、密码相结合的方式经系统验证通过后方可使用本软件系统
	权限控制需求EC-Req-Safety-Authorization-001	医护人员只能在老人处于紧急状况、需要求助时可控制机器人查看老人的状况，家属可以在任何时候查看老人的所有信息
运行环境约束	客户端EC-Req-Env-Client-001	客户端APP运行在Android 4.4及以上版本
	服务器端EC-Req-Env-Server-001	服务器端的软件运行在Ubuntu 14.04及以上版本；本软件运行时占用的内存空间不得超过128MB

# 示例：“Mini-12306”的非功能需求

非功能性需求项	非功能性需求描述
性能	<ul style="list-style-type: none"><li>– 用户界面操作的响应时间不超过1秒</li></ul>
可靠性	<ul style="list-style-type: none"><li>– 软件系统每周7天、每天24小时可用</li><li>– 系统正常运行时间的比例在99%以上</li><li>– 任何故障都不应导致用户已提交数据的丢失</li><li>– 发生故障后系统需在5分钟内恢复正常使用</li></ul>
易用性	<ul style="list-style-type: none"><li>– 界面简洁直观，易于操作</li></ul>
安全性	<ul style="list-style-type: none"><li>– 系统能够抵御网络攻击，保护数据不被非法窃取</li></ul>
私密性	<ul style="list-style-type: none"><li>– 系统中的用户数据不被非授权人员访问和获取</li></ul>
可移植性	<ul style="list-style-type: none"><li>– App软件能够方便地移植到其他的移动操作系统环境</li></ul>
运行环境约束	<ul style="list-style-type: none"><li>– App须运行在Android 4.4、鸿蒙3.0及iOS 17以上版本</li></ul>
本地化与国际化	<ul style="list-style-type: none"><li>– 支持中文和英文两种用户界面</li></ul>

# 借助互联网大众来构思软件需求

- “三人行必有我师焉，三个臭皮匠顶上一个诸葛亮”
- 利用开源社区中的开发者大众来推动需求构思，获得多样化和有新意的软件需求

# 基于大模型技术及工具生成软件需求(1/2)

- 根据用户的提示，生成软件需求的描述
- 创建与项目目标一致的需求描述，进而生成用户使用场景的案例，并推荐软件系统的产品功能
- 开发者以此为基础，将抽象、笼统的概念转化为明确、详实的软件需求

HO

You

我想要开发一款类似于“铁路12306”的购票应用软件，请你帮我分析该软件系统的所需的功能，并分条列出。

ChatGPT

开发一款类似于“铁路12306”的购票应用软件，需要考虑用户体验、安全性、高效性等因素。以下是一些核心功能和特性，这些可以作为软件系统设计的基础：

## 1. 用户账号管理

- 用户注册与登录：支持邮箱、手机号等多种注册方式。
- 身份验证：包括实名认证，可能还需要与国家铁路局的数据库进行对接验证。
- 账号安全：密码保护、验证码、二次验证等。
- 个人信息管理：允许用户更新个人信息，如联系方式、密码等。
- 账户余额与积分系统：用于购票支付和积分兑换。

## 2. 票务查询

- 车次查询：根据出发地、目的地、出发日期等条件查询车次。
- 票价信息：显示各种座位类型（如硬座、软卧等）的价格。
- 实时余票查询：显示各车次的剩余票数。
- 车站信息查询：提供车站的基本信息，如车站设施、位置图等。

## 3. 在线购票

- 快速订票：为用户提供快速购票流程，包括选择车次、座位类型、填写乘客信息等。
- 购票限制：根据国家规定和铁路局的要求，设置购票的限制和规则。
- 支付方式：支持多种支付方式，如信用卡、借记卡、支付宝、微信支付等。
- 订单管理：用户可以查看、修改、取消订单，并且能够打印电子车票或发票。

## 4. 座位选择与预订

- 动态座位图：显示车厢布局和座位状态，允许用户选择具体座位。
- 多人订票：支持一次性为多位乘客订票，并尽量安排相邻座位。

## 5. 实时信息与通知

- 车次动态：提供列车实时位置、预计到达时间等信息。
- 旅行提醒：出发前通过短信或应用内推送提醒用户车次信息。
- 紧急通知：如有突发事件或变更，能够及时通知到用户。

## 6. 客户服务与反馈

- 常见问题解答：提供一个解答常见问题的区域，比如退票政策、儿童购票等。
- 客户反馈：用户可以反馈问题或提出建议。
- 在线客服：提供在线聊天或电话客服，以帮助解决用户的疑问和问题。

## 7. 安全与隐私保护

- 数据加密：确保用户数据和交易信息的安全。
- 隐私政策：清晰地说明应用如何收集、使用和保护用户的个人信息。

## 8. 多语言支持

- 支持多种语言，尤其是在多民族国家，以提升不同用户的体验。

这些功能是构建一个完整、高效、用户友好的铁路票务购票系统的基础。在实际开发过程中，还需根据用户反馈和技术进步不断调整和优化。

# 基于大模型技术及工具生成软件需求(2/2)

- 进一步细化和完善需求
- 提供软件需求的细节
- 生成需求的用户故事

HO

You

请针对上述功能中的“在线购票”模块的功能需求进行分解和细化，给出具体的子功能点，以多级列表的形式呈现。

ChatGPT

在线购票模块是购票应用软件中最核心的部分，涉及到用户的操作体验、支付安全、订单管理等多个方面。以下是“在线购票”模块功能需求的分解和细化：

## 1. 快速订票

- 选择出发和到达站点
  - 提供搜索和选择功能，支持模糊匹配。
  - 显示最近搜索历史或热门目的地选择。
- 选择出发日期和时间
  - 日历控件选择日期。
  - 时间段选择，如早上、下午、晚上。
- 选择车次和座位类型
  - 显示不同车次的发车时间、到达时间、历时、票价和余票信息。
  - 选择座位类型（硬座、软座、硬卧、软卧等）。
- 填写乘客信息
  - 输入或选择乘客姓名、身份证号等信息。
  - 支持添加儿童、学生等特殊乘客并应用相关优惠。
- 确认订单信息
  - 显示所选车次、座位、乘客信息和总价。
  - 提供修改订单信息的选项。

## 2. 购票限制

- 实名认证要求
  - 要求乘客完成实名认证才能购票。
- 购票数量限制
  - 根据车次和用户等级，限制每次购票的最大数量。

## 3. 支付方式

- 选择支付方式
  - 支持多种支付方式：信用卡、借记卡、第三方支付（如支付宝、微信支付）等。
- 完成支付
  - 提供安全的支付环境，确保交易信息加密。
  - 支付成功后，显示支付结果和订单详情。

## 4. 订单管理

- 查看订单
  - 显示当前和历史订单的详细信息，包括车次、乘客、座位、价格等。
- 修改订单
  - 在规定时间内允许修改部分订单信息，如乘客信息。
- 取消订单
  - 在规定时间内允许取消订单并根据退票政策退款。
- 电子车票和发票
  - 提供电子车票下载和打印功能。
  - 根据需求提供电子发票申请和下载。

这样的功能细化有助于开发团队理解需求、分配任务，并确保每个细节都能得到妥善处理，从而提高整个购票系统的用户满意度和效率。

# 合法和非法的软件需求

## □并非软件利益相关方提出的每项期望和要求都是软件需求

✓如果他们提出的要求与软件及其欲解决的问题无关、没有实际的意义和价值、或者不存在技术等方面可行性，那么这些要求不应成为软件需求

## □软件需求工程师要有一双“火眼睛”，能够鉴别和发现真正有意义和有价值的重要软件需求，剔除“无意义和无价值”的软件需求

# 软件创意示例1：无人值守图书馆

## 要解决的问题

- ✓ 找寻图书困难

## 软件创意

- ✓ 线上借阅
- ✓ 线下机器人获取
- ✓ 自动找寻



### 线上基于智能终端的借阅



### 线下基于机器人的服务

国防科学技术大学  
参赛队员：张元良 卢腾 冯辰冬 曹书锦 杨东须  
指导老师：毛新军 刘汉泽  
院系单位：国防科学技术大学计算机学院

无人值守的图书借阅系统

颠覆你的借阅习惯  
开创图书借阅新时代

产品背景：

- 读者自助借书、找书、归书、借书仍然是当前图书馆实现图书借阅的主要流程和模式，但是由于图书摆放的无序性、读者搬运书籍后的随意性等情况下，读者和图书馆员往往面临着需要花费大量的时间和体力去搬书的问题。针对这一问题，我们将自主机器人引入到图书馆借阅流程，并结合移动终端的移动计算技术，推出无需人工参与的全新的借书模式，将读者从繁琐的找书过程和搬书过程中解放出来。

创新价值：

- 创新图书借阅模式；
- 创新机器人与移动终端的交互；
- 机器人自主完成图书归位任务；

实现功能：

- 读者通过移动终端发送借书请求；
- 机器人接收收书任务，自主移动到图书所在地点，根据书行为，并送达至预定的存放地点；
- 机器人拿取书本时识别读者，读者接收到反馈信息后即可前往取书；

技术路线：

- 基于Android平台开发消费者的手机软件app；
- 基于nao机器人平台开发面向队长任务的机器人服务功能；
- 基于后台管理平台开发用于接收和分派读者收书请求的后台系统。

无人值守的图书借阅系统

机器人送进书架      路标标记的书架      访问开馆后台系统

机器人送进书架      路标标记的书架      访问开馆后台系统

机器人送进书架      路标标记的书架      访问开馆后台系统

# 软件创意示例2：多无人机联合搜寻软件系统

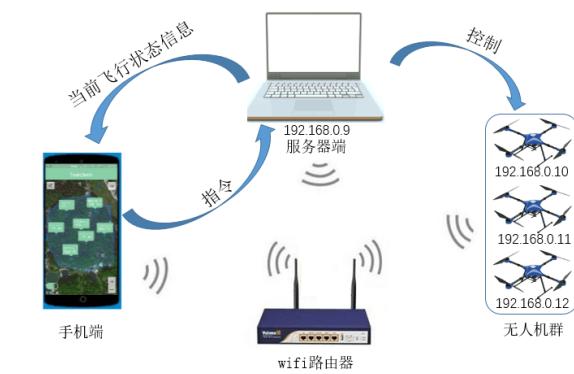
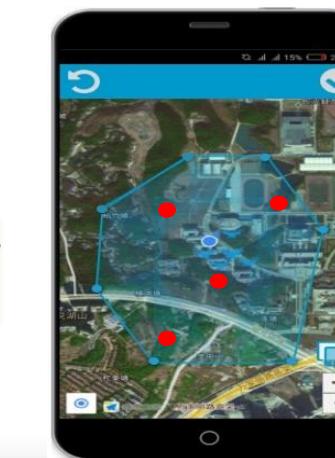
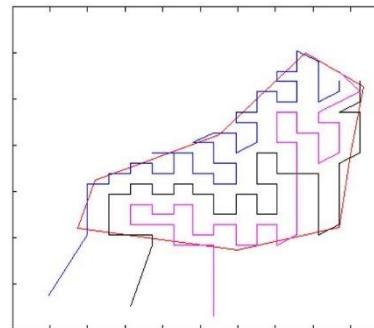
## 要解决的问题

- ✓ 快速定位被埋人员



## 软件创意

- ✓ 借助手机定位
- ✓ 采用无人机编队搜索
- ✓ 自动生成位置



# 内容

## 1. 获取软件需求的方法

- ✓ 软件需求的方式和过程

## 2. 明确问题及软件解决方案

- ✓ Problem → Software Solution

## 3. 导出和构思软件需求

- ✓ 如何从利益相关方导出和构思软件需求

## 4. 描述初步的软件需求

- ✓ 自然语言描述和可视化建模

## 5. 确认和验证初步软件需求



## 4.1 描述初步软件需求

### □ 初步的软件需求

✓之所以称之为是**初步的**，是因为这些软件需求还很**粗略**，只是一个初步的需求轮廓，**不够具体和详尽，可能有遗漏，会存在不一致和相冲突等问题**，后续还需要对其开展进一步的精化和分析

### □ 描述需求

✓**记录下来、描述清楚**，形成相关的软件文档  
✓便于不同人员（如需求工程师、用户、客户等）之间的**交流和讨论**，及时发现需求理解上存在的偏差，支持后续的需求分析工作

# 初步需求的描述方法

- ① 自然语言描述
- ② 软件原型描述
- ③ 用例图描述

## 4.2 自然语言描述

### 口自然语言是最为常用的需求描述手段

- ✓ 描述软件的**功能性需求、质量需求和开发约束需求**等
- ✓ 详细刻画需求的具体内容和细节
- ✓ 可为各方所理解，便于交流和讨论

### 口自然语言描述的局限性

- ✓ **有二义**，不同的人看完描述后可能会有不同的理解
- ✓ **不直观**，很难从中厘清软件系统到底有哪些功能性需求和非功能性需求，这些需求之间存在什么样的关系

# 示例：用自然语言描述的初步软件需求(1/2)

## □功能性需求描述

- ✓ 软件系统需要对老人在家的状况进行分析，以判断是否出现突发异常情况。一旦出现异常情况，就需要通知老人家属和医生

## □质量需求描述

- ✓ 老人通过语音方式与系统进行交互，系统正确理解老人语音指令的比率应达到90%以上

## □软件开发约束需求描述

- ✓ 客户端App软件须运行在Android 4.4及以上版本的操作系统

# 示例：用自然语言描述的初步软件需求(2/2)

## □功能性需求描述

- ✓ 软件系统应提供查询车次功能，旅客输入出发地和目的地及日期，系统应查询出满足条件的所有车次

## □质量需求描述

- ✓ 系统中的用户数据不被非授权人员访问和获取

## □软件开发约束需求描述

- ✓ 客户端App软件须运行在Android 4.4及以上版本的操作系统

## 4.3 软件原型描述

### 口优势

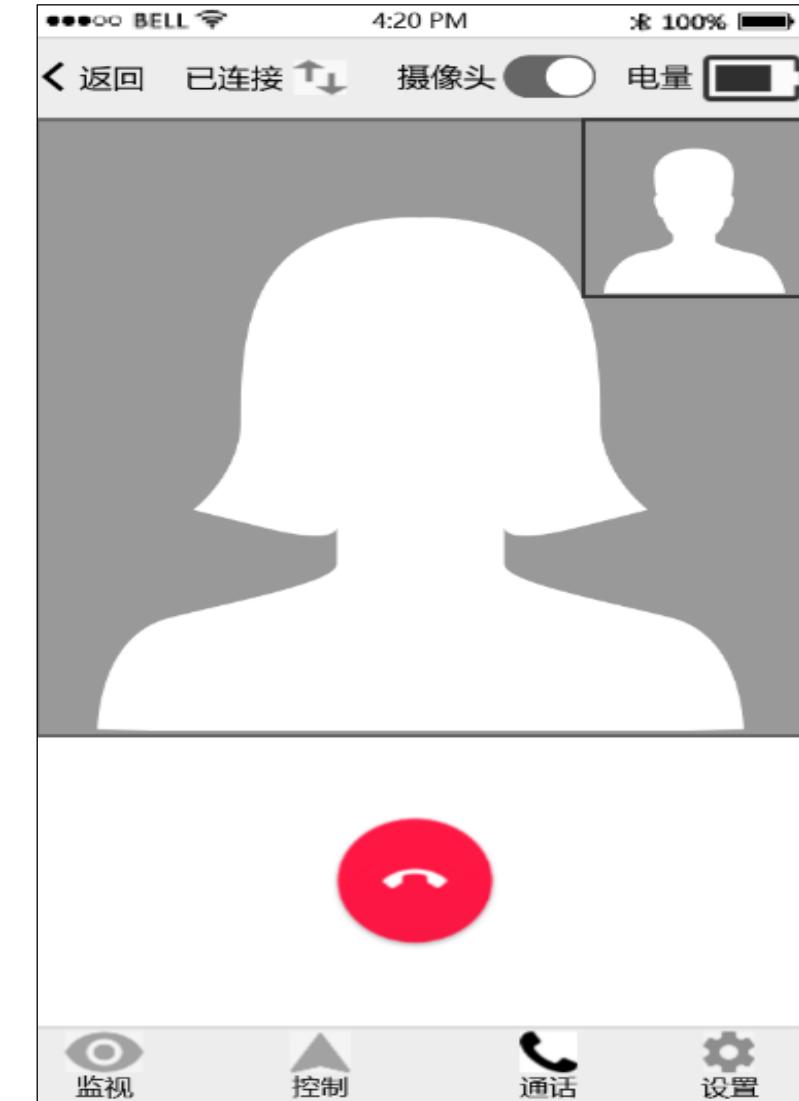
- ✓**直观、可展示和可操作**
- ✓有助于需求工程师与用户或客户之间的交流和沟通，便于在操作和使用软件原型的过程中帮助用户和客户确认和导出软件需求

### 口不足

- ✓以操作界面的形式展示软件需求的梗概，主要是软件与用户之间的输入和输出，业务的大致流程，**无法描述软件需求的具体细节**

# 示例：空巢老人看护软件的界面原型

- 界面的整体布局
- 显示的主要信息
- 提供的主要按钮



空巢老人看护软件  
前端APP的界面原型

# 示例：Mini-12306 APP的界面原型

- 界面的整体布局
- 显示的主要信息
- 提供的主要按钮



Mini-12306软件  
前端APP的界面原型

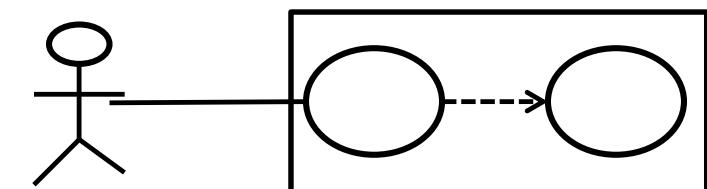
## 4.4 用例图描述

### 口用途

- ✓ 描述软件系统的**边界**以及软件**外部使用者**所观察到的**系统功能**
- ✓ “**观察到**” 是指外部使用者与系统存在**交互**，即信息输入和输出

### 口图的构成

- ✓ **执行者** (Actor) : 系统的外部使用者
- ✓ **用例** (Use Case) : 观察到的功能和交互
- ✓ **边**: 执行者与用例间、用例之间、执行者之间的关系



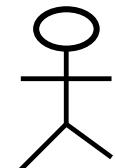
# 1. 执行者(Actor)

## □何为执行者

- ✓ 系统**之外的实体**，他们使用软件系统功能、与软件系统交换信息
- ✓ 可以是一类**用户**，也可以是其他**软件系统或物理设备**

## □执行者是UML中的类

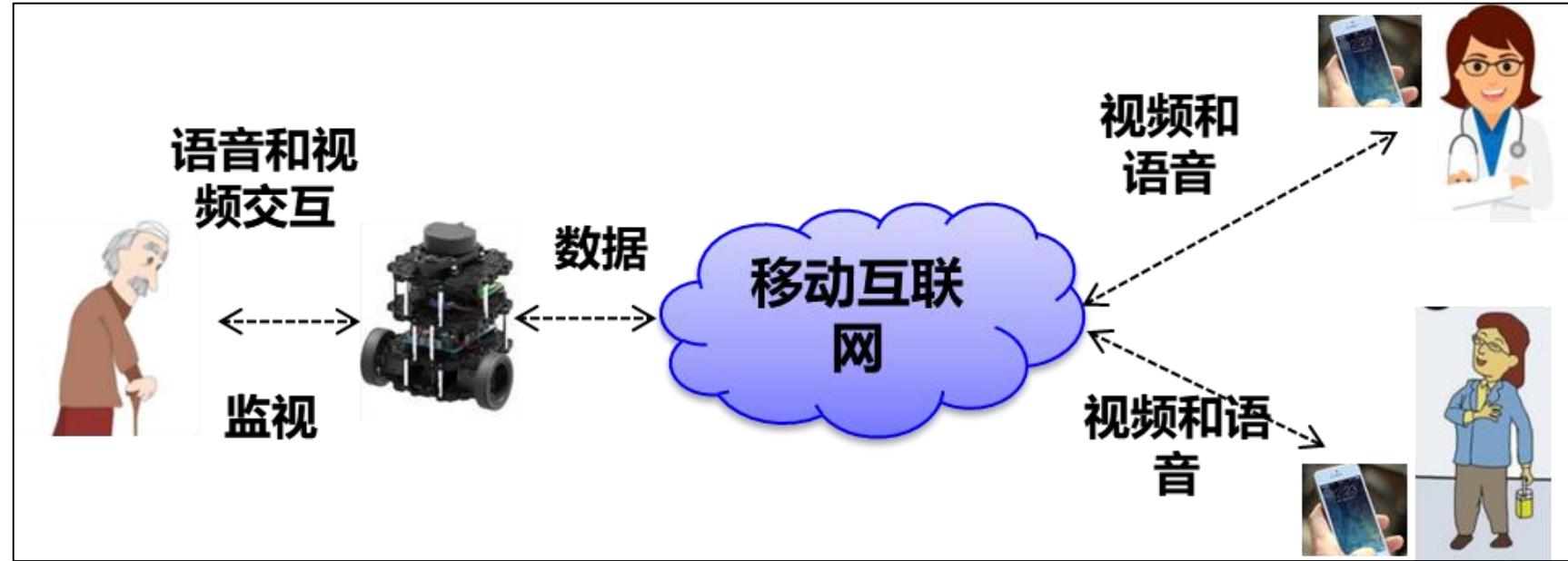
- ✓ 代表一类用户或者外部实体，而非具体的对象实例



Actor  
Name

# 示例：“空巢老人看护软件”的执行者

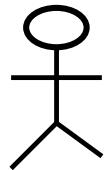
- 老人
- 医生
- 家属
- 机器人
- 管理员
- ...



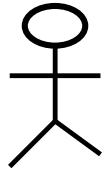
执行者通常对应于软件系统的利益相关方

# 示例：Mini-12306的外部执行者

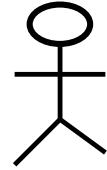
- 旅客
- 售票员
- 系统管理员



旅客



售票员



系统管理员

## 2. 用例(Use Case)

### 口用例

- ✓ 表示执行者为达成一项相对独立、完整的业务目标而要求软件系统完成的**功能**

Use Case

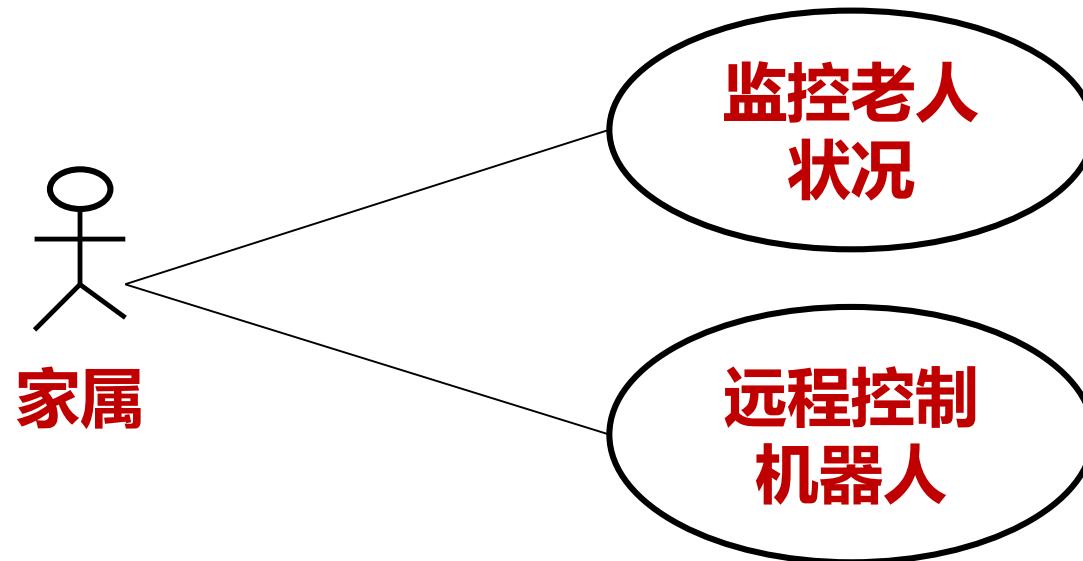
### 口用例的表现形式

- ✓ 表现为执行者与系统之间的**业务交互动作的序列**
- ✓ 对于执行者而言，交互目的或者效果在于达成其业务目标
- ✓ 对于待开发系统而言，交互的过程即是某项相对独立、完整的外部可见功能的实现过程

# 示例：用例

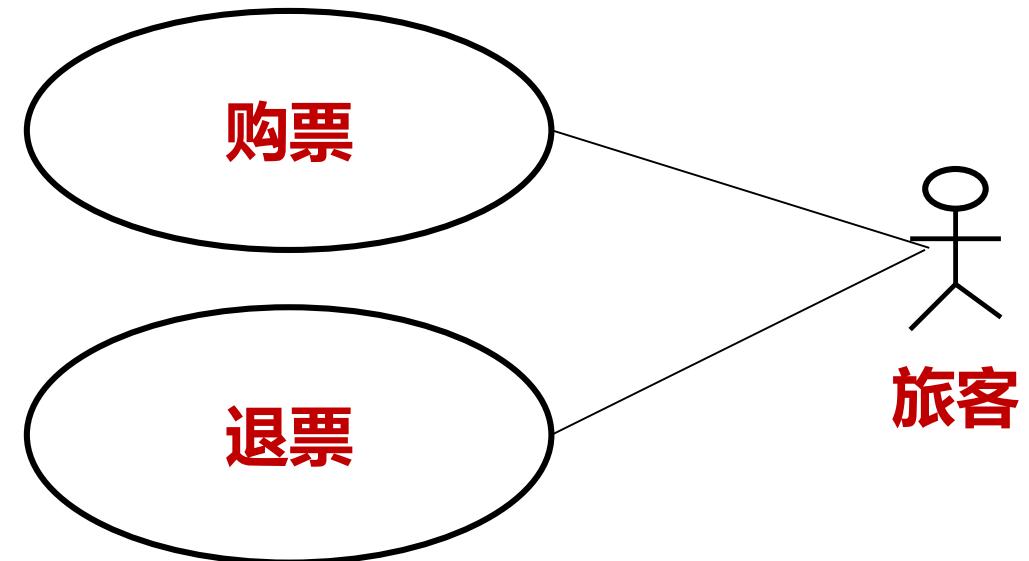
## □ “家属” 所看到的用例

- ✓ 监控老人状况
- ✓ 远程控制机器人



## □ “旅客” 所看到的用例

- ✓ 购票
- ✓ 退票

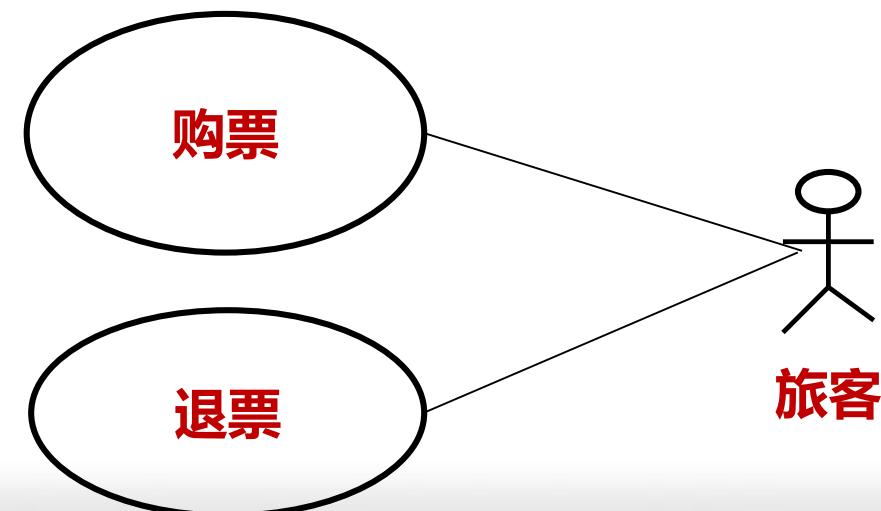
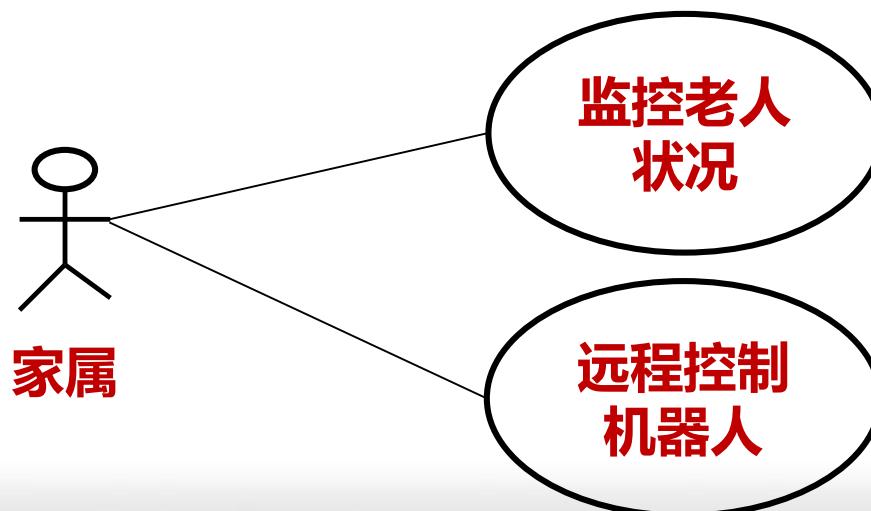


用名称表示执行者，用动名词表示用例

### 3. 执行者与用例间的关系

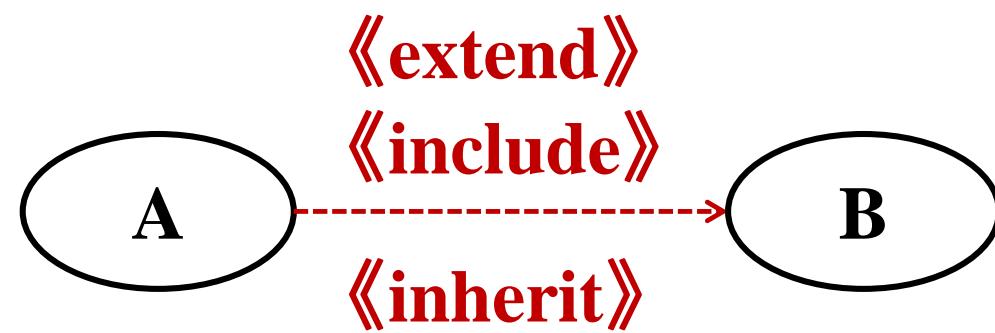
#### 口执行者与用例间关系的内涵

- ✓ 执行者与用例之间存在**交互**
- ✓ 执行者触发用例执行，向用例**提供信息或从用例获取信息**
- ✓ 触发用例执行的执行者称为**主动执行者**，仅从用例获取信息的执行者称为**被动执行者**
- ✓ 执行者与用例间的边通常为无向边



# 用例间的关系

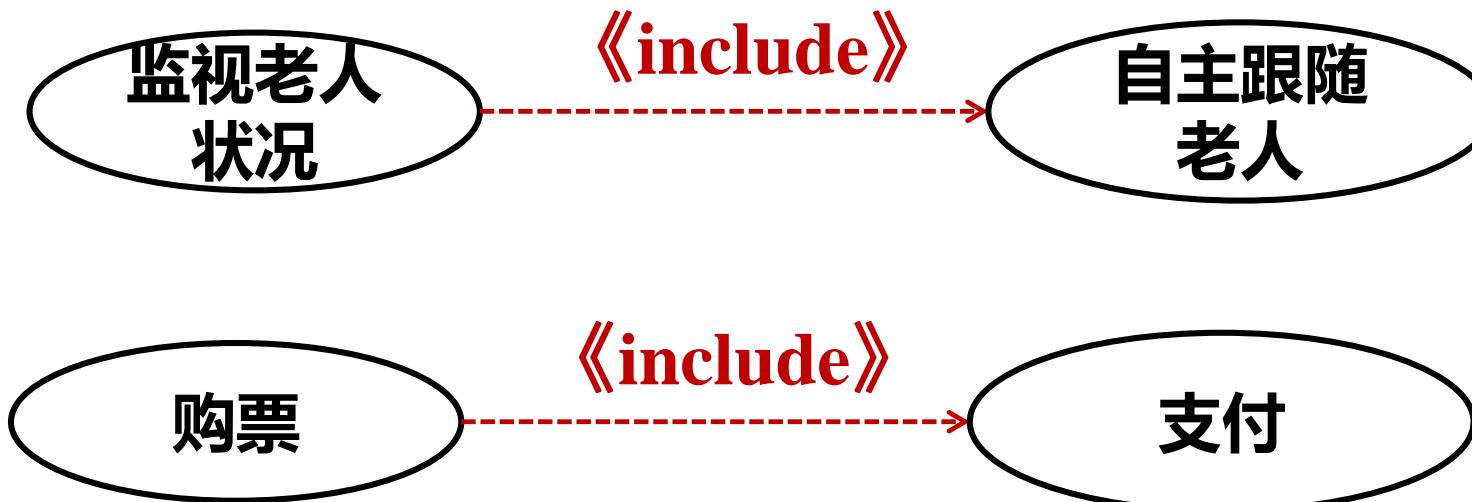
- ① 包含(Include)
- ② 扩展(Extend)
- ③ 继承(Inherit)



# 用例间的关系-包含关系

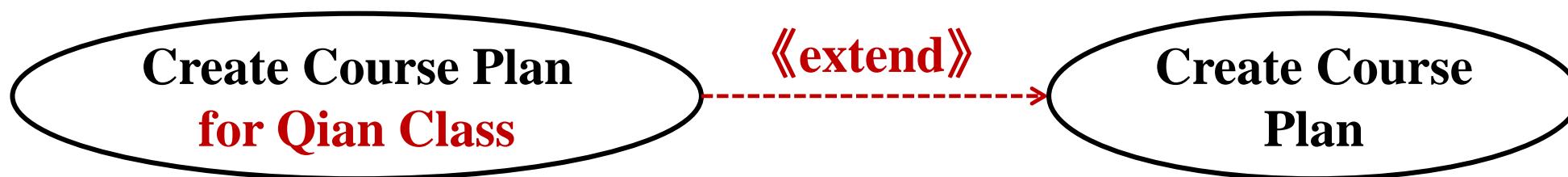
如果用例B是用例A的某项子功能，则称用例A包含用例B

- ✓ 包含关系用于提取多个用例中的公共子功能，以避免重复和冗余
- ✓ 体现了功能分解和组织的思想



# 用例间的关系-扩展关系

- 如果用例A与B相似，但**A的功能较B多**，A的动作序列是在B的动作序列中的某些执行点上插入附加动作序列而构成的，则称**用例A扩展用例B**
- 用来区隔正常业务处理功能和带有例外处理的功能
  - ✓ 避免例外处理逻辑搅乱或湮灭正常处理逻辑
  - ✓ 体现了功能增强



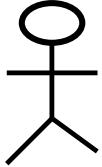
包含和扩展有何区别？



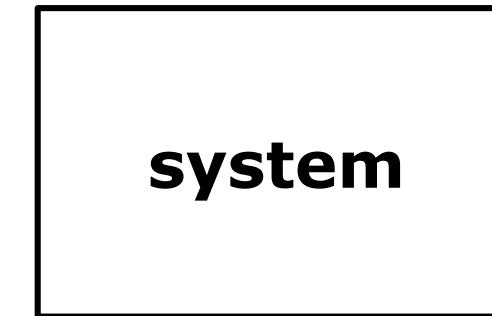
# 边界框

□表示整个软件系统或子系统的**边界**

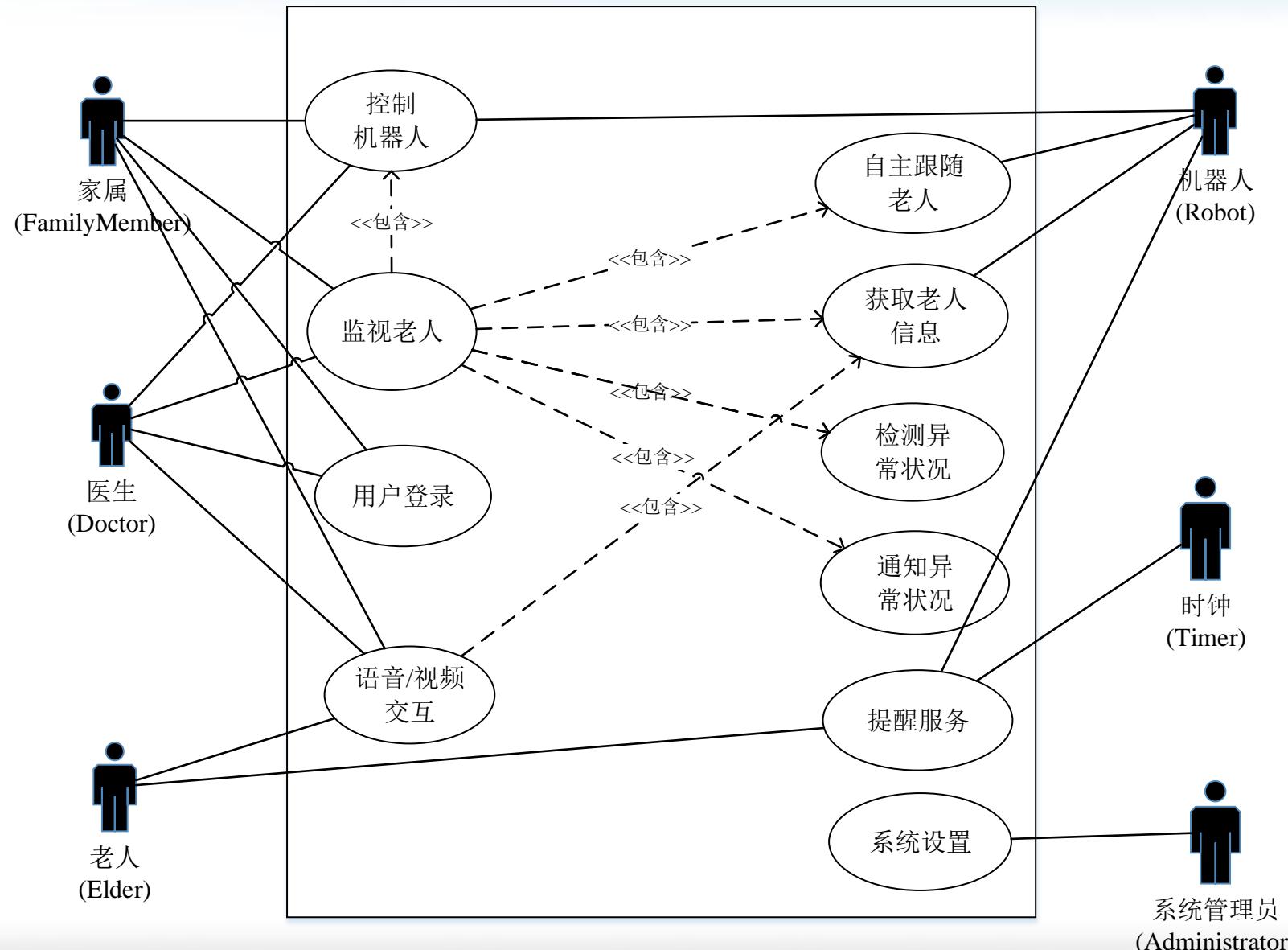
- ✓ 边界框内的用例构成了系统或子系统的内容，如用例
- ✓ 外面的是系统之外的执行者

  
Actor  
Name

Use  
Case

system

# 示例：“空巢老人看护软件”的用例图



- 图中有哪些actor
- 图中有哪些用例
- 图中有哪些类型的边，分别表示什么关系

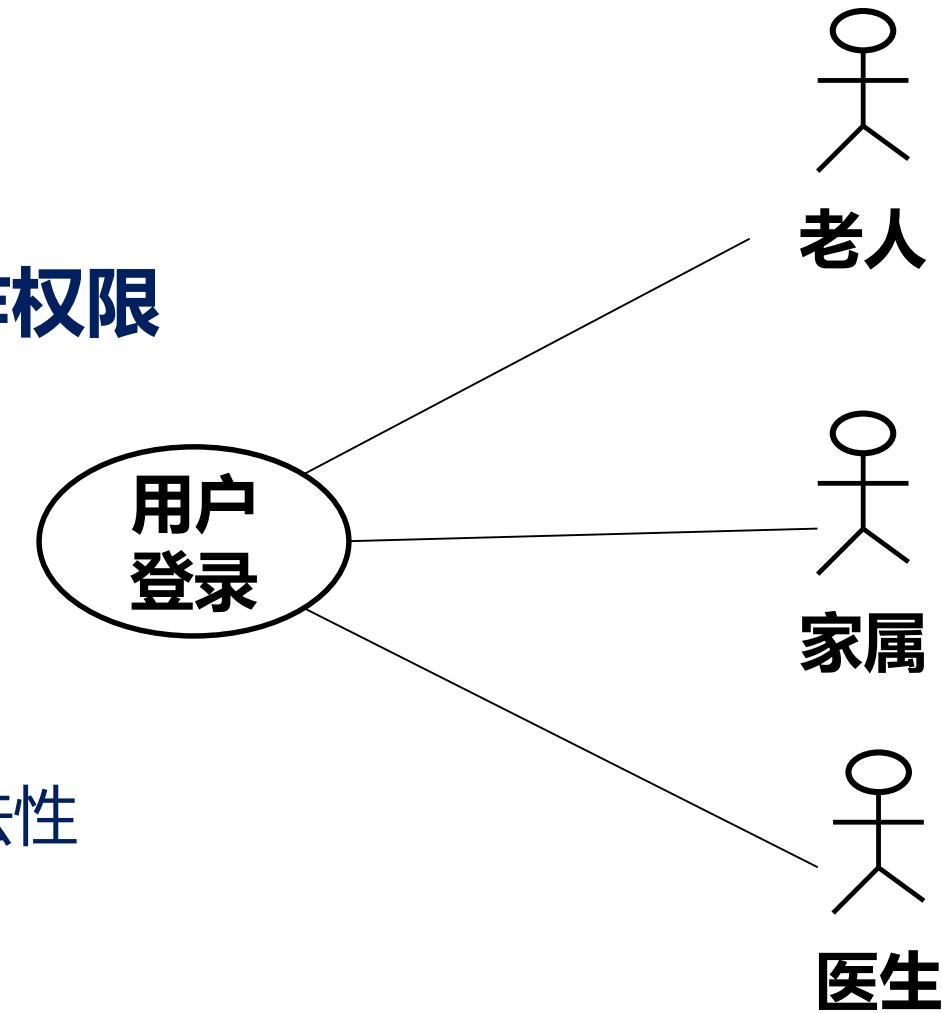
# 对用例的进一步详细描述

- 用例名
- 用例标识
- 主要执行者
- 目标
- 范围
- 前置条件
- 交互动作

为什么需要进一步描述用例的  
这些信息? 它有什么作用?

# 示例：用例描述(1/2)

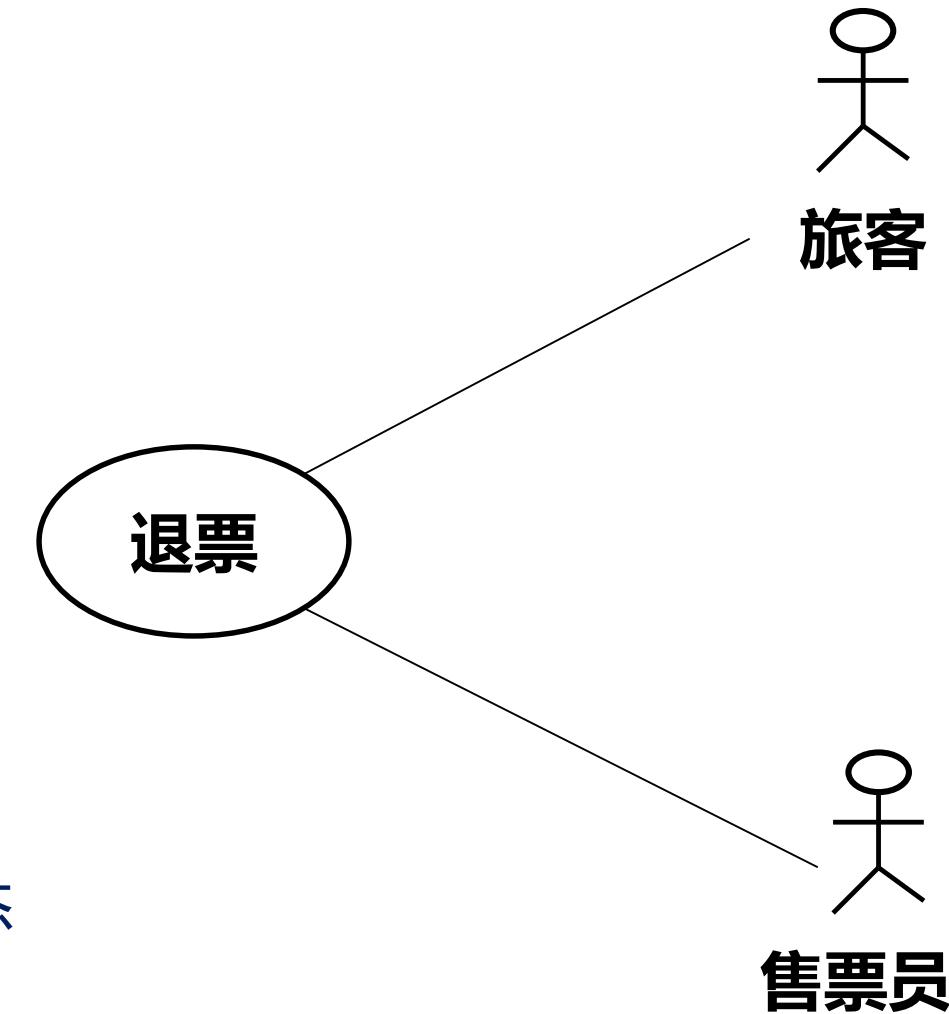
- **用例名：** 用户登录
- **用例标识：** UC-UserLogin
- **主要执行者：** 家属、医生
- **目标：** 通过合法身份登录系统以获得操作权限
- **范围：** 空巢老人看护软件
- **前置条件：** 使用App软件之时
- **交互动作**
  - (1) 用户输入账号和密码
  - (2) 系统验证用户账号和密码的正确性和合法性
  - (3) 验证正确和合法则意味着登录成功
  - (4) 提示登录是否成功的信息



# 示例：用例描述(2/2)

- **用例名：**退票
- **用例标识：** UC-RefundTicket
- **主要执行者：**旅客、售票员
- **目标：**将购买的车票退票
- **范围：** Mini-12306
- **前置条件：**已经登录系统
- **交互动作**

- (1) 提供待退票的车票
- (2) 验证车票的合法性
- (3) 将车票退票，设置车票为“未出售”状态
- (4) 将车票的费用退回到旅客的银行卡中



## 4.5 撰写“初步软件需求描述”文档

- 1. 软件背景介绍**
- 2. 欲解决的问题**
- 3. 软件解决方案**
- 4. 软件的功能性需求描述**
- 5. 软件的非功能性需求描述**
- 6. 可行性及潜在风险**

# 内容

## 1. 获取软件需求的方法

- ✓ 软件需求的方式和过程

## 2. 明确问题及软件解决方案

- ✓ Problem → Software Solution

## 3. 导出和构思软件需求

- ✓ 如何从利益相关方导出和构思软件需求

## 4. 描述初步的软件需求

- ✓ 自然语言描述和可视化建模

## 5. 确认和验证初步软件需求



# 5.1 输出的软件制品

## □ 软件原型

- ✓ 以**可运行软件的形式**展示了软件的业务工作流程、操作界面、用户的输入和输出等方面的功能性需求信息

## □ 软件用例模型

- ✓ 以**可视化图形符号的方式**刻画了软件系统的执行者、边界、用例以及它们之间的相互关系，描述了软件的功能性需求

## □ 软件需求文档

- ✓ 以**自然语言的方式**描述了初步软件需求，包括功能性和非功能性的软件需求

## 5.2 评审初步软件需求

- 中肯性**, 软件问题是否反映实际问题, 是否有意义和价值
- 合理性**, 基于软件的解决方案是否科学和合理。
- 完整性**, 软件需求是否覆盖了利益相关方的期望和要求
- 必要性**, 每一项软件需求是否有必要
- 溯源性**, 每一项软件需求是否都有其来源
- 准确性**, 描述是否清晰和准确地反映了软件需求的内涵
- 正确性**, 是否正确反映该需求提出者真实想法和关注点
- 一致性**, 软件需求文档、用例模型以及软件原型对软件系统需求的表述（包括术语等）是否一致

## 5.3 软件需求可行性分析 (1/2)

### □ 技术可行性

- ✓ 需要哪些技术，相关的技术是否成熟，现有技术能否支撑软件需求的实现，软件项目团队是否已经掌握了某些关键技术等等

### □ 设备可行性

- ✓ 基于软件的解决方案需要哪些设备或系统，软件项目团队是否已经具备这些设备和系统

### □ 进度可行性

- ✓ 软件用户或客户对软件提出什么样的进度要求，针对开发团队的人力资源及技术水平，能否遵循进度约束开发出满足这些需求的软件产品

# 软件需求可行性分析 (2/2)

## □成本可行性

- ✓ 基于软件项目成本约束，软件项目团队能够开发出满足软件需求的产品出来

## □商业可行性

- ✓ 软件需求是否有商业价值，能否获得用户的青睐以及市场的认可，针对这些软件需求的投入能否获得预期的收益

## □社会可行性

- ✓ 当前软件系统已经成为国家和社会的重要组成部分，需要从社会的角度，评估软件的各项需求是否违背社会道德、文化伦理、法律法规、行业标准等，或者与它们是否存在冲突

# 小结

## □ 软件需求来自于软件利益相关方

- ✓ 导出和构思软件需求

## □ 循序渐进地构思和导出软件需求

- ✓ 问题、基于软件的解决方法、软件利益相关方

## □ 描述初步的软件需求

- ✓ 自然语言、UML建模（用例图）、软件原型

## □ 软件需求的验证和确认

- ✓ 多方从多个方面来评审初步的软件需求

# 思考和讨论

- 你们如何获取项目的需求？
- 你们的项目画用例图了吗？如果没有，可以开始了



# 课后作业

- 安装staruml或者类似的UML软件，并尝试运行
- 5-1, 5-2, 5-5, 5-6, 5-7, 5-10, 5-17, 5-22