需求基础

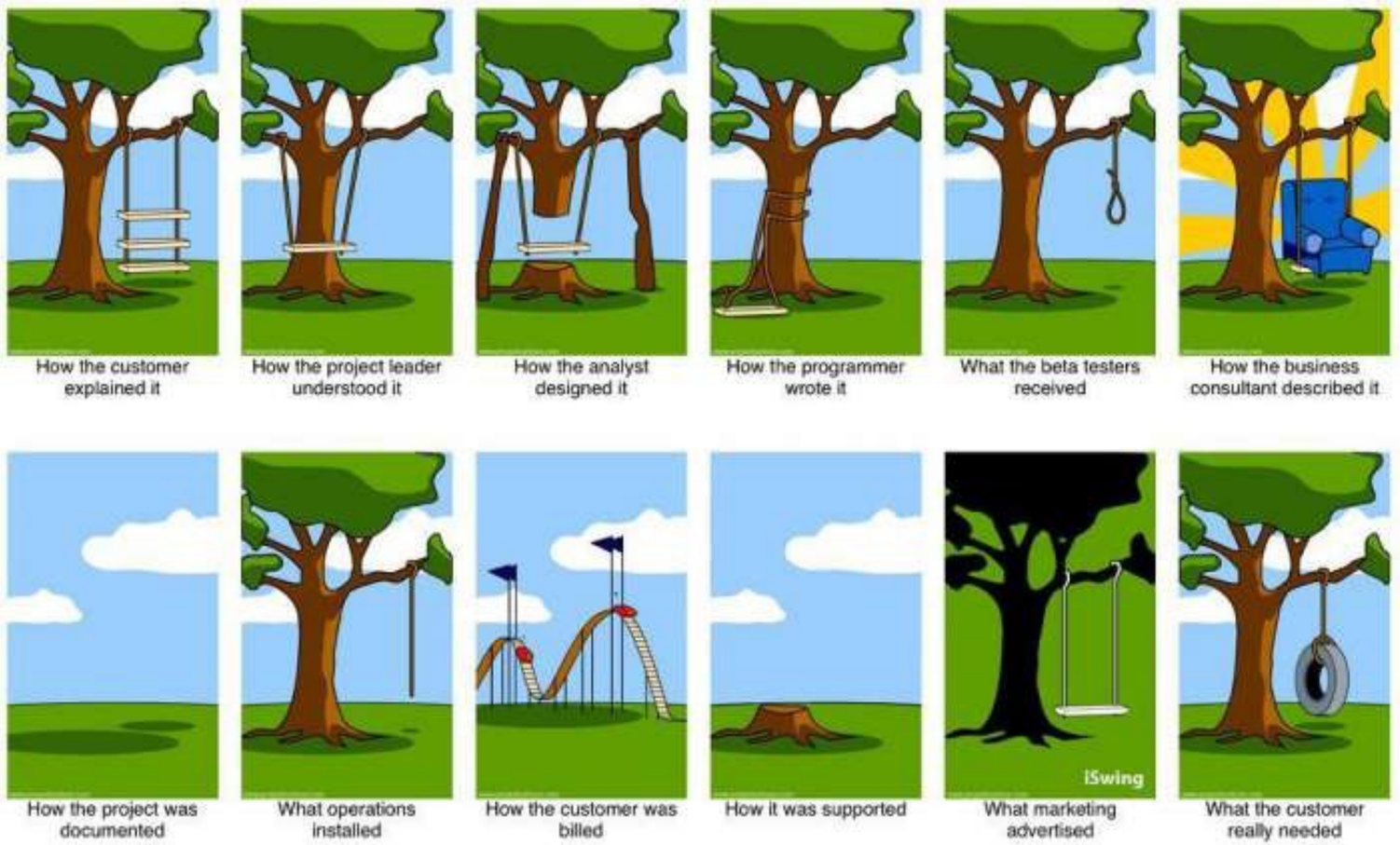
刘钦

预习测试-1

- 需求的分类有哪些?
- 以下需求是否正确？如果不正确请改正?
  - 在使用选课系统时，学生必须要在一刻钟内完成一个选课处理的所有操作。

预习测试-11

- 需求的层次性是什么?
- 以下需求是否正确？如果不正确请改正?
    - 所有的用户查询都必须在很快的时间内完成。



What the customer wanted

主要内容

- 需求工程
- 需求基础
- 需求分类

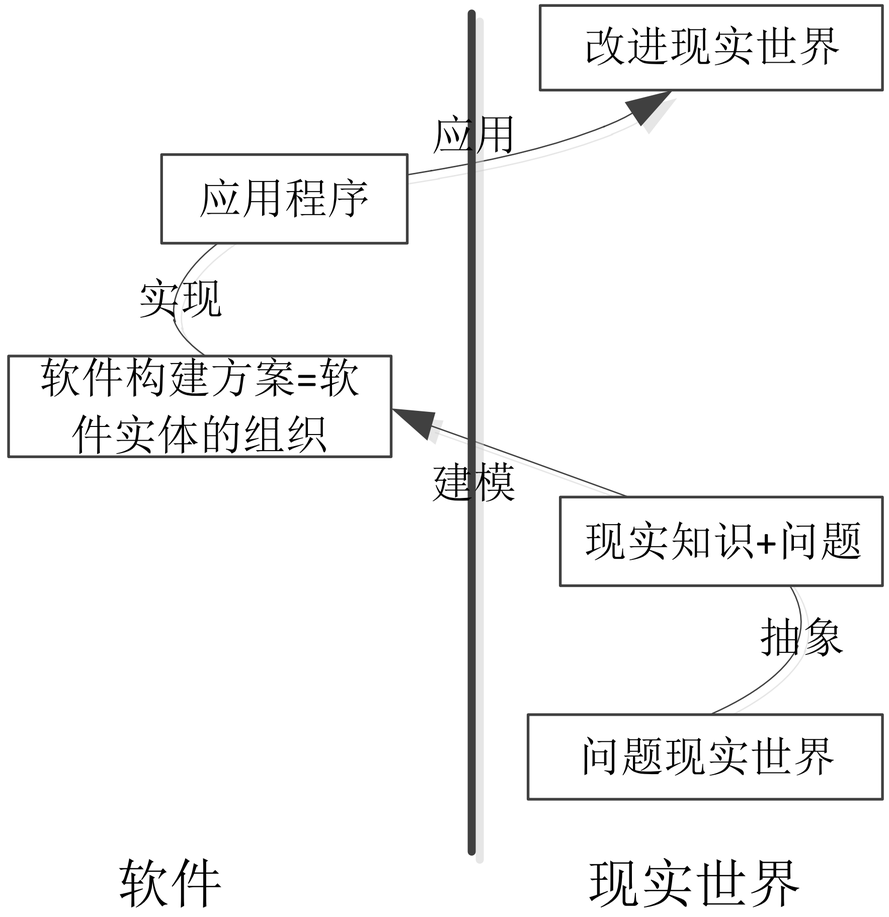
主要内容

- 需求工程
  - 需求开发
    - 需求获取
    - 需求分析
    - 规格说明
    - 需求验证
  - 需求管理
- 需求基础
- 需求分类

为什么要开发需求?
如何得到需求?

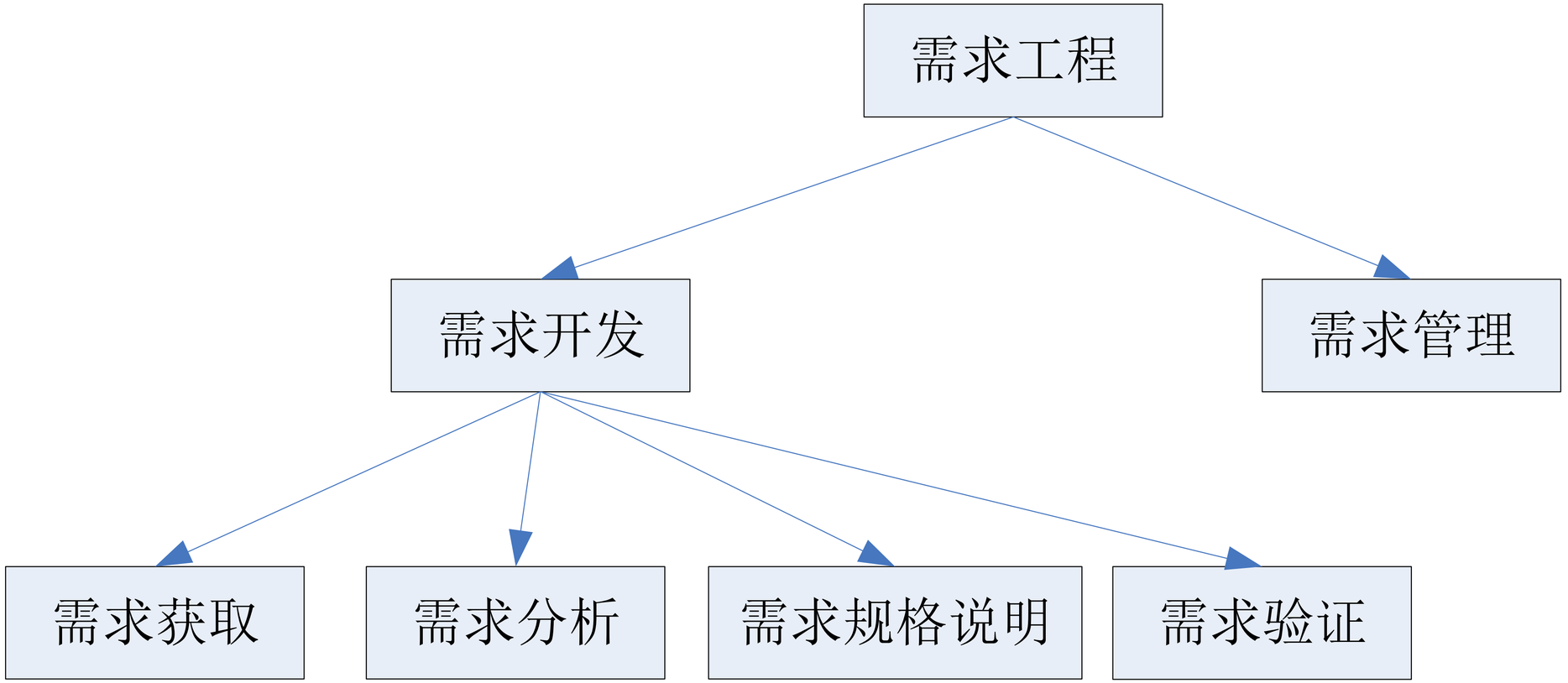
软件建立的依据?

- 单纯的软件系统是不能解决问题的，它只有和现实世界之间形成有效互动才能实现问题的解决

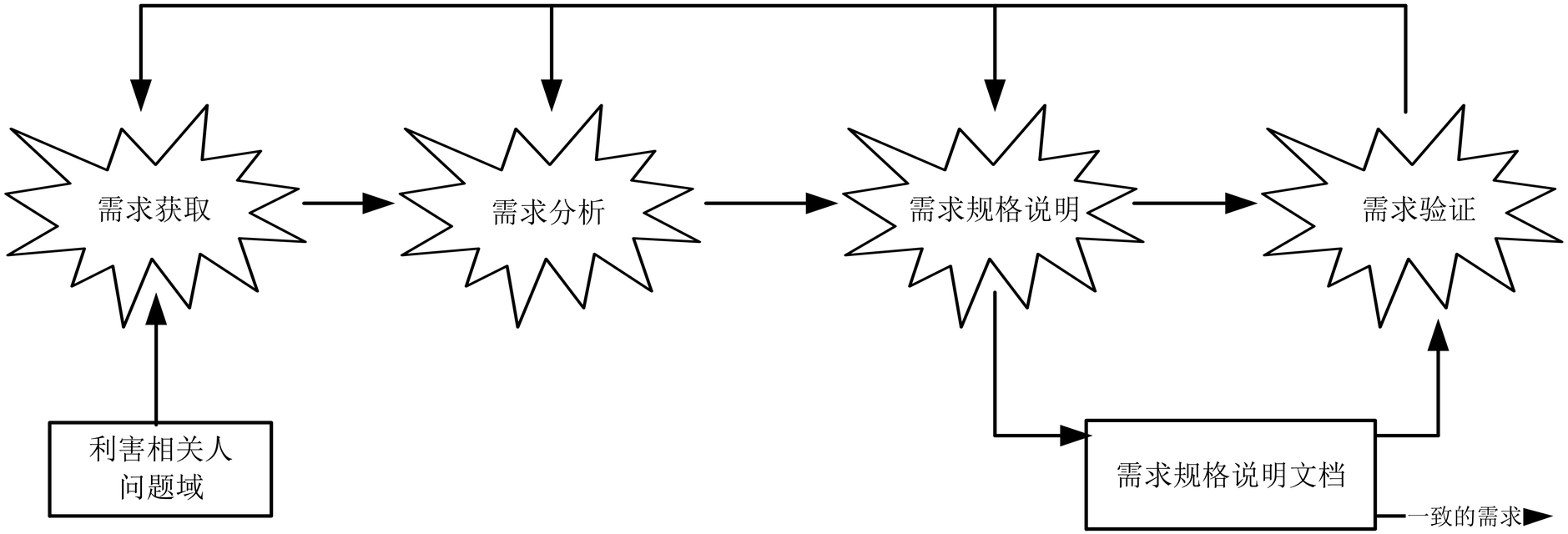


需求工程

- 概念：
  - 所有需求处理活动的总和。它收集信息、分析问题、整合观点、记录需求并验证其正确性，最终描述出软件被应用后与其环境互动形成的期望效应。
- 三个主要任务：
  - 需求工程必须说明软件系统将被应用的应用环境及其目标，说明用来达成这些目标的软件功能，也即要同时说明软件需要“做什么”和“为什么”需要做。
  - 需求工程必须将目标和功能反映到软件系统当中，映射为可行的软件行为，并对软件行为进行准确的规格说明。
  - 现实世界是不断变化的世界，因此需求工程还需要妥善处理目标和功能随着时间演化的变动情况。



需求工程的活动



需求开发过程模型

需求获取

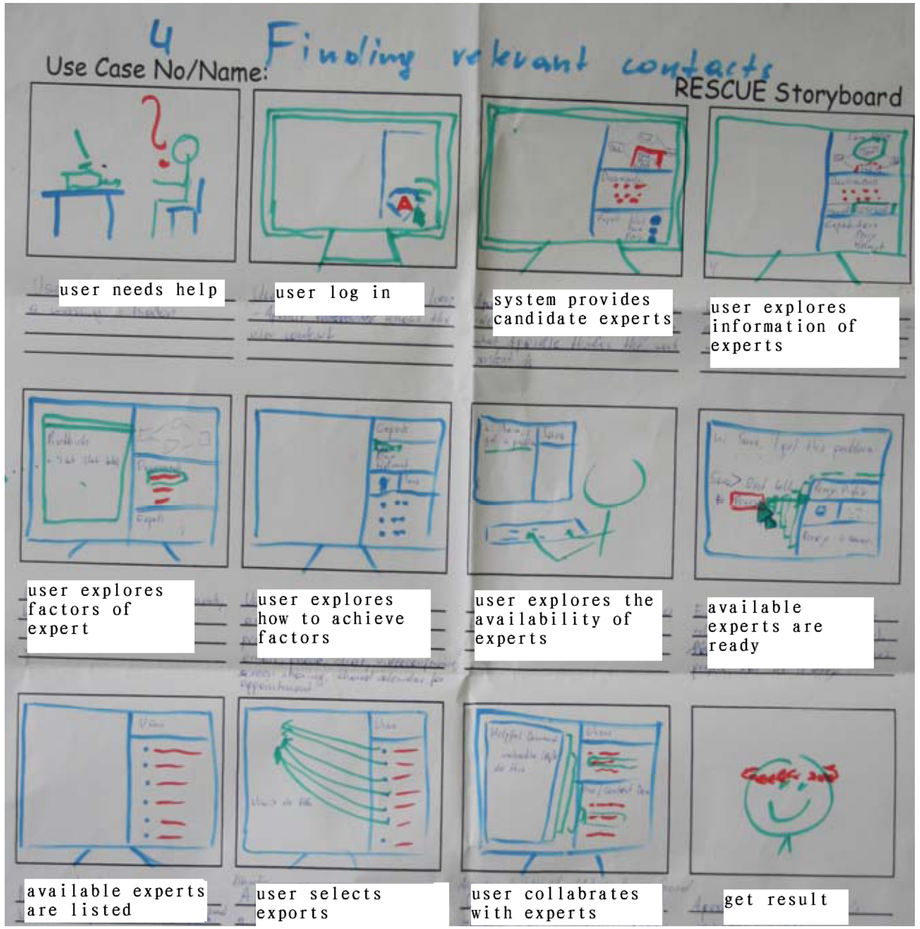
- 从人、文档或者环境当中获取需求的过程
- 要利用各种方法和技术来“发现”需求
- 目标分析
  - ⑴根据问题确定目标
  - ⑵通过分析利害关系人确定目标

需求获取的常见困难

- 用户和开发人员的背景不同，立场不同　背景不同，很难考虑周全。
  - “床边B超,肝胆胰脾”
- 普通用户缺乏概括性、综合性的表述能力
  - 不聪明的记者
- 用户存在认知困境
  - 平板电脑
- 用户越俎代庖　不要歧视笨蛋甲方，虽然他们又菜又装，但是他们懂业务。
  - 双机热备
  - 我们就是要求系统能够。。。，至于怎么实现是你开发者的事
- 缺乏用户参与
  - 不愿参与的医生

用户需求获取的方法

- 面谈
- 问卷
- 文档分析
- 头脑风暴
- 专题讨论
- 原型



以Story board作为原型

重新认识需求获取

- 用户和开发人员的背景不同，立场不同
  - “床边B超,肝胆胰脾” －－ 消除默认知识
- 普通用户缺乏概括性、综合性的表述能力
  - 不聪明的记者 －－ 专业的需求人员
- 用户存在认知困境
  - 平板电脑 －－ 原型
- 用户越俎代庖
  - 双机热备 －－ 需求是开发人员开发出来的，不是用户提出来的
  - 我们就是要求系统能够。。。，至于怎么实现是你开发者的事 －－ 协商
- 缺乏用户参与
  - 不愿参与的医生 －－ 为用户参与提供方便

需求分析

- 通过建模来整合各种信息，以使得人们更好的理解问题。
- 为问题定义出一个需求集合，这个集合能够为问题界定一个有效的解决方案。
- 检查需求当中存在的错误、遗漏、不一致等各种缺陷，并加以修正。

需求分析

- 一、边界分析
  - 定义项目的范围
  - 系统边界的定义要保证系统能够和周围环境形成有效的互动
  - 系统用例图通常被用来定义系统的边界
- 二、需求建模
  - 建模是为展现和解释信息而进行的抽象描述活动
  - 常用的技术包括类图、顺序图、状态图等建模技术

需求规格说明

- 在系统用户之间交流需求信息
- 要简洁、精确、一致和易于理解

- 需求工程师在这个阶段的重要工作包括:
  - 一、定制文档模版
  - 二、编写文档

需求验证

- 需求规格说明文档至少要满足下面几个标准：
  - 文档内每条需求都正确、准确的反映了用户的意图；
  - 文档记录的需求集在整体上具有完整性和一致性；
  - 文档的组织方式和需求的书写方式具有可读性和可修改性。

验证的方法

- 同级评审
- 原型
- 模拟

需求管理

- 保证需求作用的持续、稳定和有效发挥
  - 在需求开发活动之后，设计、测试、实现等后续的软件系统开发活动都需要以围绕需求开展工作
- 进行变更控制
  - 纳入和实现合理的变更请求，拒绝不合理的变更请求，控制变更的成本和影响范围

Outline

- 需求工程
- 需求基础
  - 什么是需求?
  - 需求的层次
- 需求分类

什么是需求?

需求

- IEEE对需求的定义为[IEEE610.12-1990]：
  - ⑴用户为了解决问题或达到某些目标所需要的条件或能力；
  - ⑵系统或系统部件为了满足合同、标准、规范或其它正式文档所规定的要求而需要具备的条件或能力；
  - ⑶对⑴或⑵中的一个条件或一种能力的一种文档化表述。

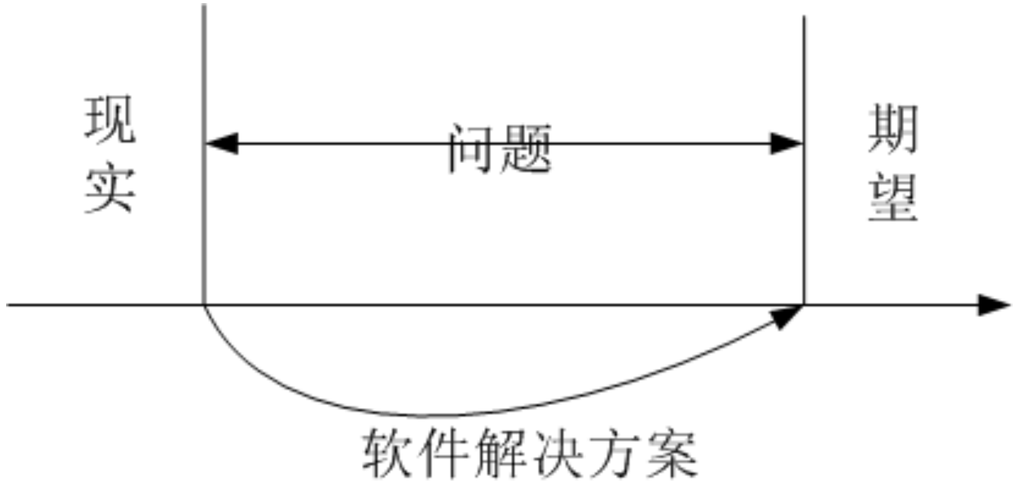
需求的表述

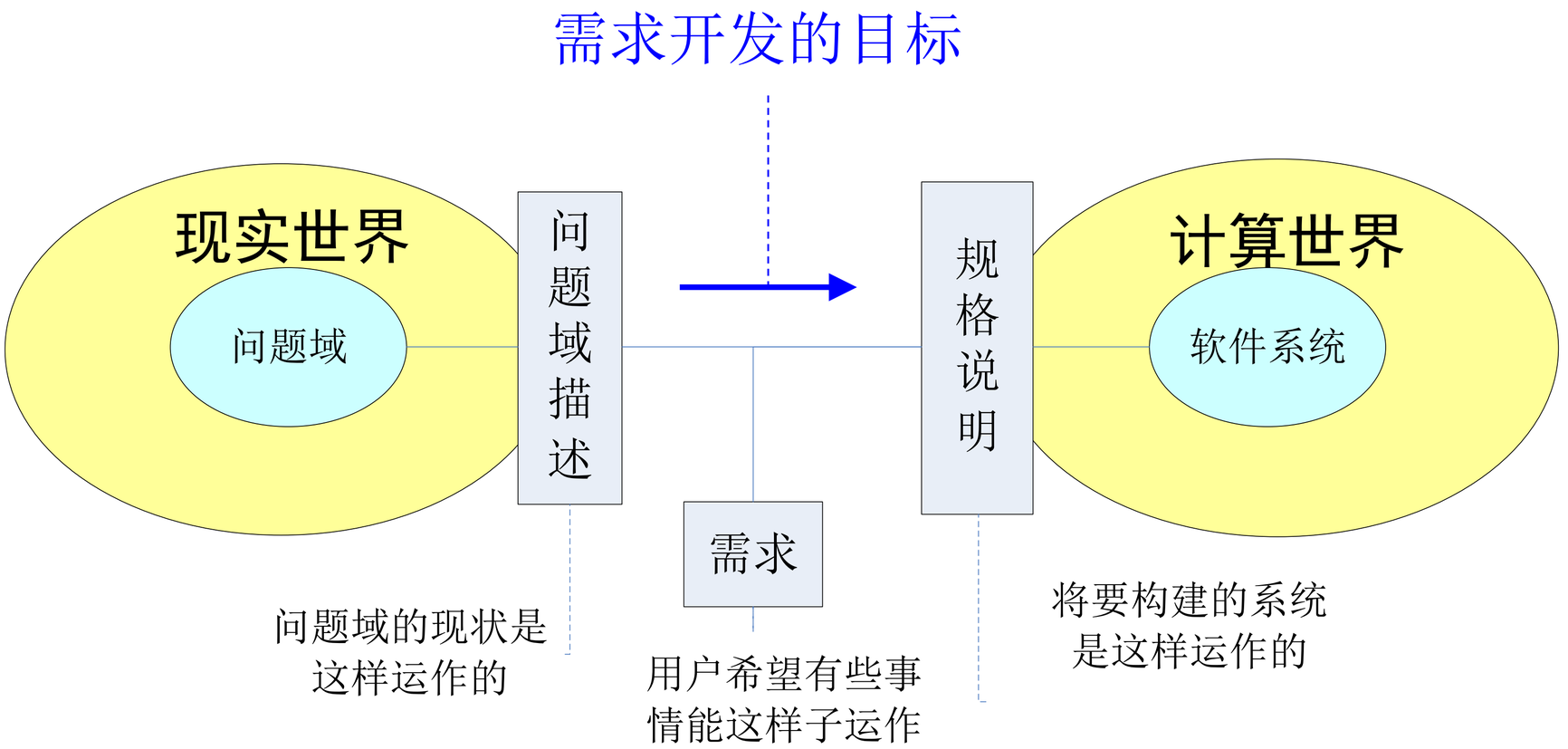
- 作为一种期望，需求通常被表述为“系统应该…”、“在…时，系统应该…”、“用户可以通过系统…”等，例如R1。

- R1：系统应该允许顾客退回已经购买的产品。

软件解决方案

- 需求是一种解决问题后所能达到的期望
- 一件事情的两面
  - 问题是糟糕的一面
  - 需求是理想的一面




需求开发的目标
现实世界
问题域
问题域描述
规格说明
计算世界
软件系统
需求
问题域的现状是
这样运作的
用户希望有些事
情能这样子运作
将要构建的系统
是这样运作的

需求开发的目标

需求

- 是一种期望
- 源自现实又高于现实
- 需求是多变和可调整的，项目可以依据实际情况调整需求的实现程度。

问题域

- 现实世界运行规律的一种反映
- 需求的产生地，也是需求的解决地。
- 最终的软件产品要在现实中部署，它能够部分影响问题域，但不能任意改变现实
    - 软件开发必须尊重问题域，不能因为技术原因妄自修改现实世界的实际情况。

问题的解决

- 基础
  - 模拟与共享现象
- 方法
  - 直接与间接
- 解决方案
  - 需求规格说明

规格说明

- 软件产品的方案描述，它以软件产品的运行机制为主要内容。
- 它不是需求但实现需求，不是问题域但需要与问题域互动。
- 规格说明要以关注对外交互的方式描述软件解决方案，它既需要从软件产品的角度而不是用户的角度进行描述，又不能太多地涉及软件产品的内部构造机制。

超市销售系统

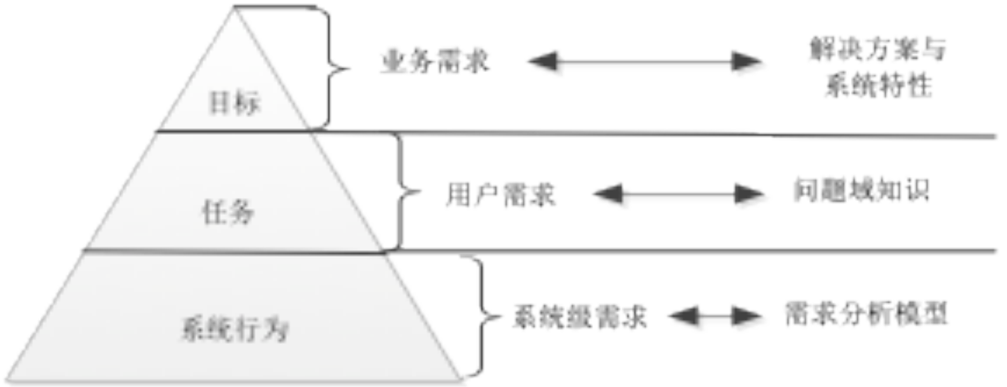
- 问题
  - 超市的成本太高
- 需求
  - 超市的成本应该降低
- 问题域
  - 超市的成本主要由人力成本和库存成本组成
- 规格说明
  - 库存预测
  - 库存报警

需求层次性

R2：在系统使用*3*个月后，销售额度应该提高*20%*。

R3：系统要帮助收银员完成销售处理。

R4：收银员输入购买的商品时，系统要显示该商品的描述、单价、数量和总价。



业务需求

- 系统建立的战略出发点，表现为高层次的目标（Objective），它描述了组织为什么要开发系统
- 为了满足用户的业务需求，需求工程师需要描述系统高层次的解决方案，定义系统应该具备的特性（Feature）
- 参与各方必须要对高层次的解决方案达成一致，以建立一个共同的前景（Vision）
- 特性说明了系统为用户提供的各项功能，它限定了系统的范围（Scope）

案例

- R2：在系统使用3个月后，销售额度应该提高20%
- 可以建立高层次的解决方案，其系统特性如SF1～SF4所示。
  - SF1：管理VIP顾客信息。
  - SF2：提供VIP顾客服务，增加回头率。
  - SF3：使用多样化的特价方案，吸引顾客购买，增加销售额。
  - SF4：使用多样化的赠送方案，吸引顾客购买，增加销售额。

用户需求

- 执行实际工作的用户对系统所能完成的具体任务的期望，描述了系统能够帮助用户做些什么
  - 直接用户
  - 间接用户（通用软件的销售人员和售后支持人员）
- 对所有的用户需求，都应该有充分的问题域知识作为背景支持
- 特性
  - 模糊、不清晰（允许适度的用形容词和副词）
  - 多特性混杂 （功能和非功能的混杂）
  - 多逻辑混杂 （一个任务需要多次系统交互才能完成）

该软件管理工具软件必须帮助项目管理者进行开发管理工作，以通过CMMI-4的评估。“项目管理者”是用户

案例

- SF1：管理VIP顾客信息
- 针对每一个系统特性，都可以建立一组用户需求。例如对SF1，可以建立用户需求组如UR1.1～UR1.7，它们中每一条都是用户完成具体任务所需要的功能：
  - UR1.1：系统应该允许客户经理添加、修改或者删除会员个人信息。
  - UR1.2：系统应该记录会员的购买信息。
  - UR1.3：系统应该允许客户经理查看会员的个人信息和购买信息。
  - UR1.4：系统应该允许客户经理查看所有会员的统计信息。

补充问题域知识

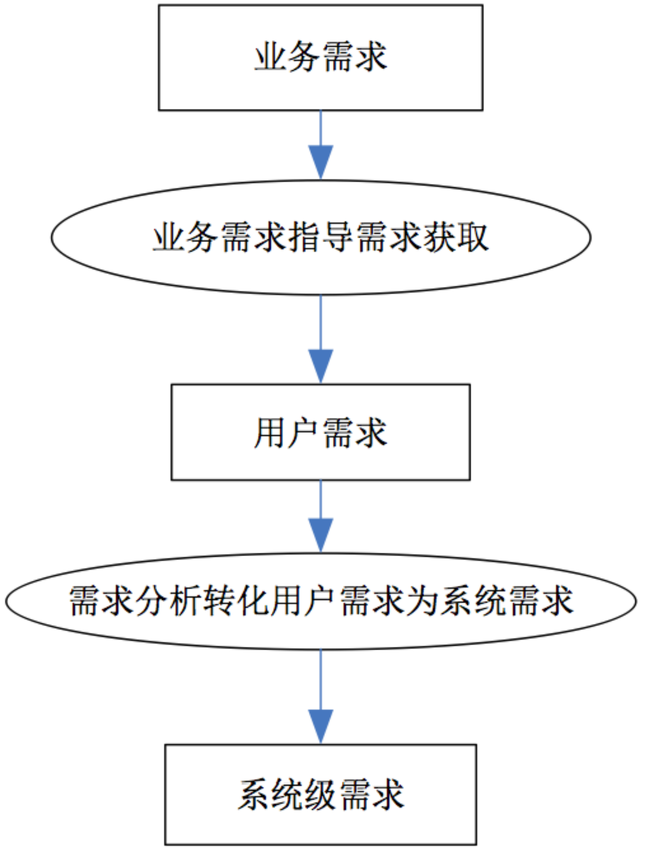
- 用户需求表达了用户对系统的期望，但是要透彻和全面的了解用户的真正意途，仅仅拥有期望是不够的，还需要知道期望所来源的背景知识。
- 因此，对所有的用户需求，都应该有充分的问题域知识作为背景支持。
  - UR1.1：系统应该允许客户经理添加、修改或者删除会员个人信息。
- 例如对UR1.1，需要补充问题域知识如下：
  - 会员的个人信息有：客户编号、姓名、联系方式、积分。

系统需求

- 用户对系统行为的期望，每个系统级需求反映了一次外界与系统的交互行为，或者系统的一个实现细节
- 描述了开发人员需要实现什么
- 将用户需求转化为系统需求的过程是一个复杂的过程
  - 首先需要分析问题领域及其特性，从中发现问题域和计算机系统的共享知识，建立系统的知识模型；
  - 然后将用户需求部署到系统模型当中，即定义系列的系统行为，让它们联合起来实现用户需求，每一个系统行为即为一个系统需求。
  - 该过程就是需求工程当中最为重要的需求分析活动，又称建模与分析活动。

案例

- UR1.3：系统应该允许客户经理查看会员的个人信息和购买信息。
- 对用户需求UR1.3，可以依据任务中的交互细节将之转化为系统级需求SR1.3.1～SR1.3.4。
  - SR1.3.1在接到客户经理的请求后，系统应该为客户经理提供所有会员的个人信息。
  - SR1.3.2在客户经理输入会员的客户编号时，系统要提供该会员的个人信息。
  - SR1.3.3在客户经理选定一个会员并申请查看购买信息时，系统要提供该会员的历史购买记录。
  - SR1.3.4经理可以通过键盘输入客户编号，也可以通过读卡器输入客户编号。



从功能需求的层次性看需求开发

课堂练习

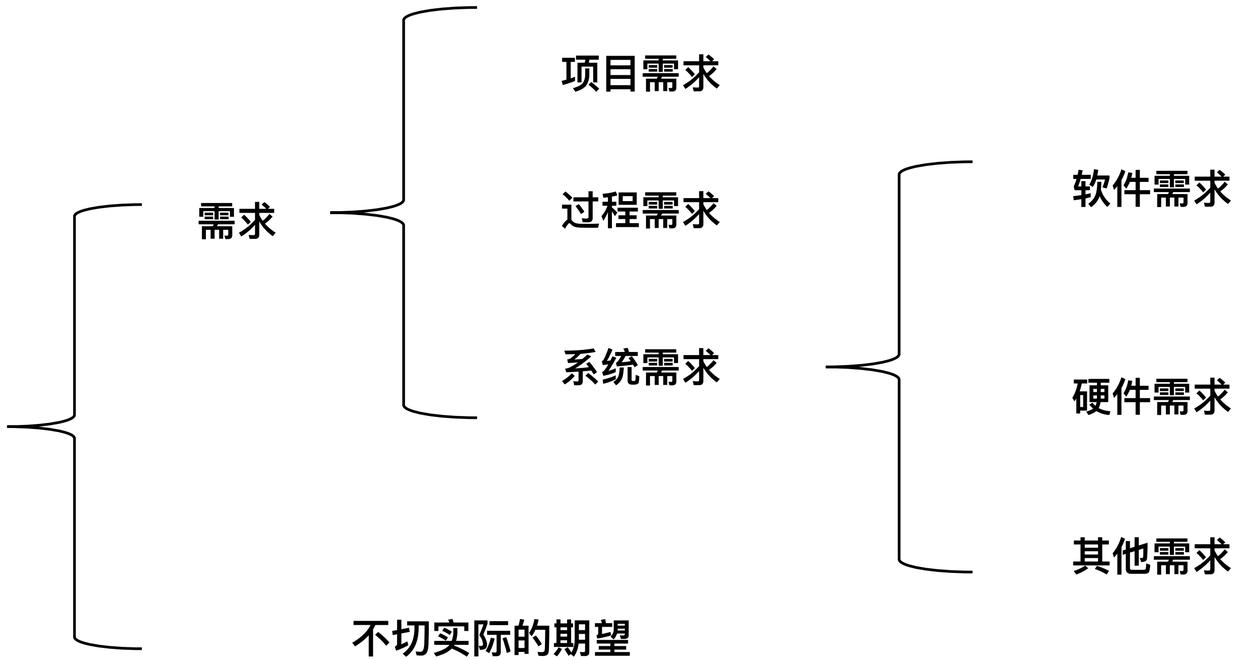
- NBA数据分析应用
- 业务需求 — NBA球队老板希望知道谁是2014-2015赛季最大心脏的投手?
- 用户需求?
- 系统级需求?

Outline

- 需求工程
- 需求基础
- 需求分类
  - 需求谱系
  - 功能需求
  - 性能需求、质量属性、对外接口、约束

需求如何分类?

需求的图谱



案例

- 项目需求
  - R5：项目的成本要控制在60万元人民币以下。
  - R6：项目要在6个月内完成。
- 过程需求
  - R7：在开发中，开发者要提交软件需求规格说明文档、设计描述文档和测试报告。
  - R8：项目要使用持续集成方法进行开发。
- 其他需求
  - R9：系统要购买专用服务器，其规格不低于….。
  - R10：系统投入使用时，需要对用户进行1个星期的集中培训。

- 不切实际的期望
  - R11：系统要分析会员的购买记录，预测该会员将来一周和一个月内会购买的商品；
  - R12：系统要能够对每月的出入库以及销售行为进行标准的财务分析；
  - R13：在使用系统时，收银员必须要在2个小时内完成一个销售处理的所有操作。
- 正确的形式
  - R14：如果一个销售处理任务在2个小时内没有完成，系统要撤销该任务的所有已执行操作

需求的分类IEEE

- 功能需求（Functional Requirement）：
  - 和系统主要工作相关的需求，即在不考虑物理约束的情况下，用户希望系统所能够执行的活动，这些活动可以帮助用户完成任务。功能需求主要表现为系统和环境之间的行为交互。
- 性能需求（Performance Requirement）：
  - 系统整体或系统组成部分应该拥有的性能特征，例如CPU使用率、内存使用率等。
- 质量属性（Quality Attribute）：
  - 系统完成工作的质量，即系统需要在一个“好的程度”上实现功能需求，例如可靠性程度、可维护性程度等。
- 对外接口（External Interface）：
  - 系统和环境中其他系统之间需要建立的接口，包括硬件接口、软件接口、数据库接口等等。
- 约束
  - 进行系统构造时需要遵守的约束，例如编程语言、硬件设施等

非功能性需求

功能需求

- 最常见、最主要和最重要的需求
- 能够为用户带来业务价值的系统行为
- 最需要按照三个抽象层次进行展开
- 软件产品产生价值的基础

性能需求

- 需要进行专门模拟和测试
  - 速度（Speed），系统完成任务的时间，例如PR1。
    - PR1：所有的用户查询都必须在10秒内完成。
  - 容量（Capacity），系统所能存储的数据量，例如PR2。
    - PR2：系统应该能够存储至少100万个销售信息。
  - 吞吐量（Throughput），系统在连续的时间内完成的事务数量，例如PR3。
    - PR3：解释器每分钟应该至少解析5000条没有错误的语句。
  - 负载（Load），系统可以承载的并发工作量，例如PR4。
    - PR4：系统应该允许50个营业服务器同时从集中服务器上进行数据的上传或下载。
  - 实时性（Time-Critical），严格的实时要求，例如PR5。
    - PR5：监测到病人异常后，监控器必须在0.5秒内发出警报。

系统应该能够存储3年的交易数据。
“能够”表明了这是在系统设计之前就考虑到的，
属于性能需求。
相对的，
“系统需要存储1年内的销售记录。”是数据需求

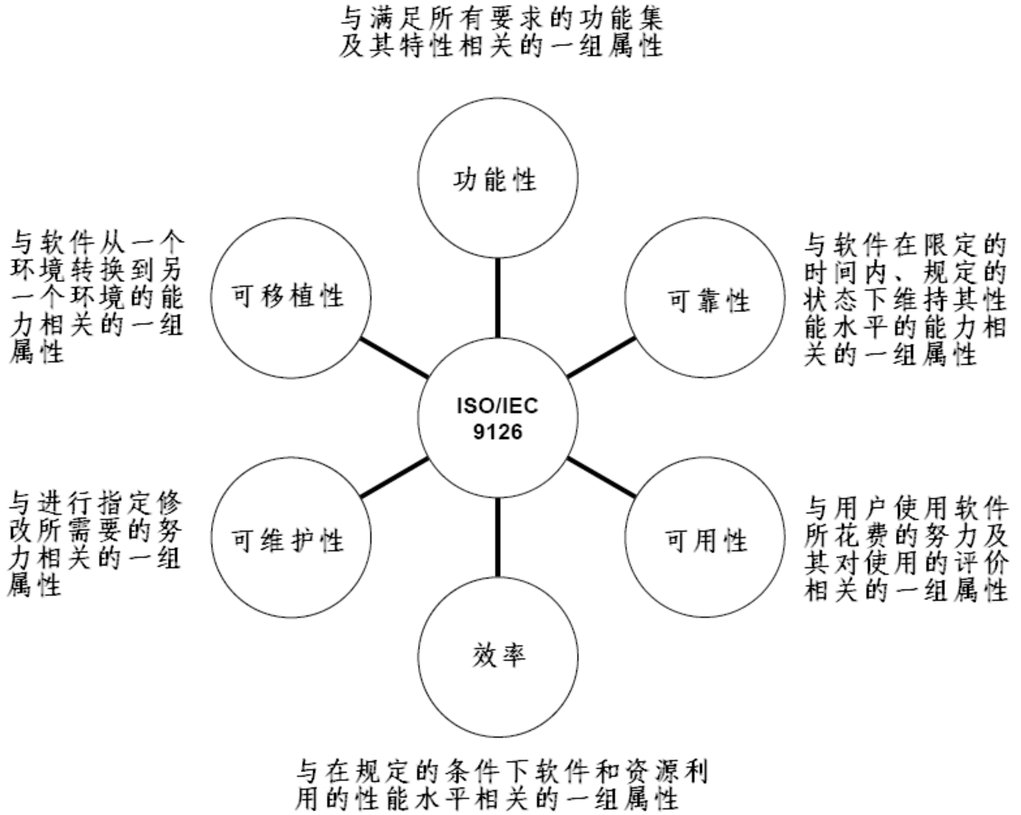
需求的灵活性

- PR6：98%的查询不能超过10秒。

- PR7：
  - （最低标准）在200个用户并发时，系统不能崩溃；
  - （一般标准）在200个用户并发时，系统应该在80%的时间内能正常工作；
  - （理想标准）在200个用户并发时，系统应该能保持正常的工作状态。

质量属性

- 系统为了满足规定的及隐含的所有要求而需要具备的要素称为质量
- 质量属性是为了度量质量要素而选用的特征
- 质量模型就是能够为质量需求的描述和评价提供工作基础的特征集及特征之间的联系
- 质量属性的重要性
  - 对设计的影响很大
  - 对越复杂的系统越为重要
  - [Robert19901]： 真实的现实系统中，在决定系统的成功或失败的因素中，满足非功能属性往往被满足功能性需求更为重要。



质量属性 - ISO/IEC 9126

常见质量属性

- 可靠性（Reliability）：在规格时间间隔内和规定条件下，系统或部件执行所要求能力的能力。
  - QA1：在进行数据的下载和上传中，如果网络故障，系统不能出现故障。

- 可用性（Availability）：软件系统在投入使用时可操作和可访问的程度或能实现其指定系统功能的概率。
  - QA2：系统的可用性要达到98%。

- 安全性（Security）：软件阻止对其程序和数据进行未授权访问的能力，未授权的访问可能是有意，也可能是无意的。
  - QA3：VIP顾客只能查看自己的个人信息和购买记录；
  - 收银员只能查看，不能修改、删除VIP顾客的信息。

常见质量属性

在存储设备发生故障时，
系统要在10秒内发现。
没有与用户交互，
不算用户需求。

- 可维护性（Maintainability）：软件系统或部件能修改以排除故障、改进性能或其他属性或适应变更了的环境的容易程度，包括可修改性（Modifiability）和可扩展性（Extensibility）。
  - QA4：如果系统要增加新的特价类型，要能够在2个人月内完成。

- 可移植性（Portability）：系统或部件能从一种硬件或软件环境转换至另外一种环境的特性。
  - QA5：集中服务器要能够在1人月内从Window 7操作系统更换到Solaris 10操作系统。

- 易用性（Usability）：与用户使用软件所花费的努力及其对使用的评价相关的特性。
  - QA6：使用系统1个月的收银员进行销售处理的效率要达到10件商品/分钟。

质量属性的开发

- 用户并不能明确地提出他们对产品质量的期望
  - 并不了解软件系统的开发过程，也就无从判断哪些质量属性会在怎样的程度上给设计带来多大的影响，也无法将他们对软件系统的质量要求细化成一组组的可量化的质量属性
- 需求工程师
  - 质量属性大都是和功能需求联系在一起的，因此需要对照软件的质量属性检查每一项功能需求，尽力去判断质量属性存在的可能性
    - 形容词和副词通常意味着质量属性的存在
  - 对于一些不和任何功能需求相联系的全局性质量属性，需求工程师要在碰到特定的实例时意识到它们的存在

对外接口

使用扫描仪扫描文件，传递回的数据为pdf格式 文件。

- 解系统和其他系统之间的软硬件接口
  - 接口的用途
  - 接口的输入输出
  - 数据格式
  - 命令格式
  - 异常处理要求
- 用户界面

约束

数据库是系统运行的基本环境，
不是普通的其他软件系统（如支付宝），
所以应该是约束而不是接口

- 总体上限制了开发人员设计和构建系统时的选择范围
  - 系统开发及运行的环境
    - 包括目标机器、操作系统、网络环境、编程语言、数据库管理系统等
    - C1：系统要使用Java语言进行开发。
  - 问题域内的相关标准
    - 包括法律法规、行业协定、企业规章等。
  - 商业规则
    - 用户在任务执行中的一些潜在规则也会限制开发人员设计和构建系统的选择范围

数据需求

- 功能需求的补充
  - 如果在功能需求部分明确定义了相关的数据结构，那么就不需要再行定义数据需求
- 数据需求是需要在数据库、文件或者其他介质中存储的数据描述，通常包括下列内容：
  - 各个功能使用的数据信息；
  - 使用频率；
  - 可访问性要求；
  - 数据实体及其关系；
  - 完整性约束；
  - 数据保持要求。

商品的标识由0-24位字母、数字混合组成的字符串。

实例

- 例如，连锁超市销售系统可以使用数据需求DR1和DR2。
  - DR1：系统需要存储的数据实体及其关系为图6-14的内容。
  - DR2：系统需要存储1年内的销售记录和退货记录。

案例 -- 注册使用Google Maps API

- 通过 Google Maps API，您可以将 Google Maps 嵌入自己的网页中。一个 Google Maps API 密钥只对一个“目录”或域有效。有关详细信息，请参见此常见问题解答。您需要拥有 Google 帐户才能获得 Google Maps API 密钥，并且 API 密钥会与您的 Google 帐户关联。

- 以下是适用于非法律专业人士的针对条款内容的一些重点概述：
    - 对每天可使用 Google Maps API 生成的网页浏览没有限制。有关详细信息，请参见此常见问题解答。
    - 每天可发出的地址解析请求数有限制。有关详细信息，请参见此常见问题解答。
    - Google Maps API 不包含广告。如果我们决定更改此策略，将会通过公告列表至少提前 90 天通知您。
    - 如果您将 Google Maps API 与其他 API 结合使用，您还应查看其他 API 的条款。请特别注意，Google Javascript 地图 API 中的 GoogleBar 使用 AJAX Search API，而 AJAX Search API 有自己的条款。
    - 您的服务必须可让最终用户免费访问。要在其他类型的应用程序中使用 Google Maps 绘制技术，请使用 Google Maps API Premier。有关详细信息，请参见此常见问题解答。
    - 您不得更改或遮掩地图上的徽标或版权归属内容。
    - 您必须指明您的应用程序是否使用传感器（例如 GPS 定位器）确定用户的位置。
    - 您可以使用网站或软件应用程序中的 API（Google Static Maps API 除外）。对于网站，请注册可在其中找到您的实施方案的网址。对于其他软件应用程序，请注册可以在其中下载您的应用程序的页面的网址。
    - Google 将定期升级 API。要获取更新通知，请订阅公告列表。
    - 请记住，我们保留随时暂停或终止您使用服务的权利，因此，请仔细阅读常见问题解答和论坛帖子以确定您的网站是否符合使用条款，然后再进行 API 集成。

约束

- 提交的地址解析请求次数是否有限制?

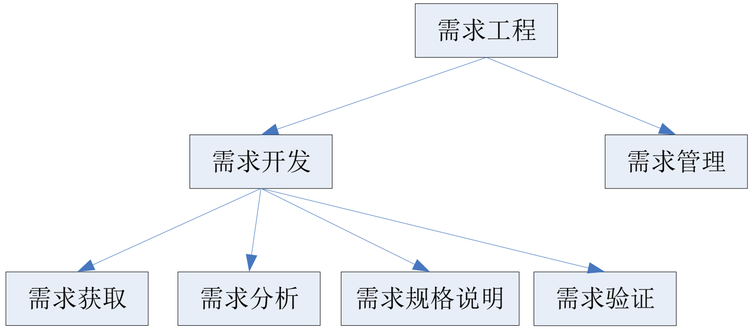
  - 如果在 24 小时时段内收到来自一个 IP 地址超过 15,000 个地址解析请求，或从一个 IP 地址提交的地址解析请求速率过快，Google 地图 API编码器将用 620 状态代码开始响应。如果地址解析器的使用仍然过多，则从该 IP 地址对 Google 地图 API 地址解析器的访问可能被永久阻止。

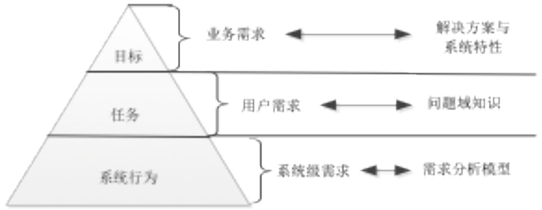
项目实践——快递物流系统

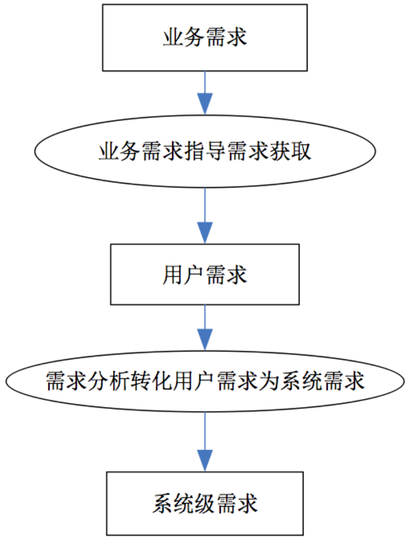
- 有没有
  - 性能需求?
  - 质量需求?
  - 对外接口?
  - 约束?
  - 数据需求?

本章重点总结

- 需求工程
- 需求
- 需求层次性
- 需求的分类







功能需求（Functional Requirement）：

- 和系统主要工作相关的需求，即在不考虑物理约束的情况下，用户希望系统所能够执行的活动，这些活动可以帮助用户完成任务。功能需求主要表现为系统和环境之间的行为交互。

性能需求（Performance Requirement）：

- 系统整体或系统组成部分应该拥有的性能特征，例如CPU使用率、内存使用率等。

质量属性（Quality Attribute）：

- 系统完成工作的质量，即系统需要在一个“好的程度”上实现功能需求，例如可靠性程度、可维护性程度等。

对外接口（External Interface）：

- 系统和环境中其他系统之间需要建立的接口，包括硬件接口、软件接口、数据库接口等等。

约束

- 进行系统构造时需要遵守的约束，例如编程语言、硬件设施等

非功能性需求

下一章预习要点

- 为什么要需求分析?
- 面向对象需求分析的模型包括哪些?
- 如何获得概念类图?
- 需求分析方法细化和明确需求