

需求分类

产品 / 过程

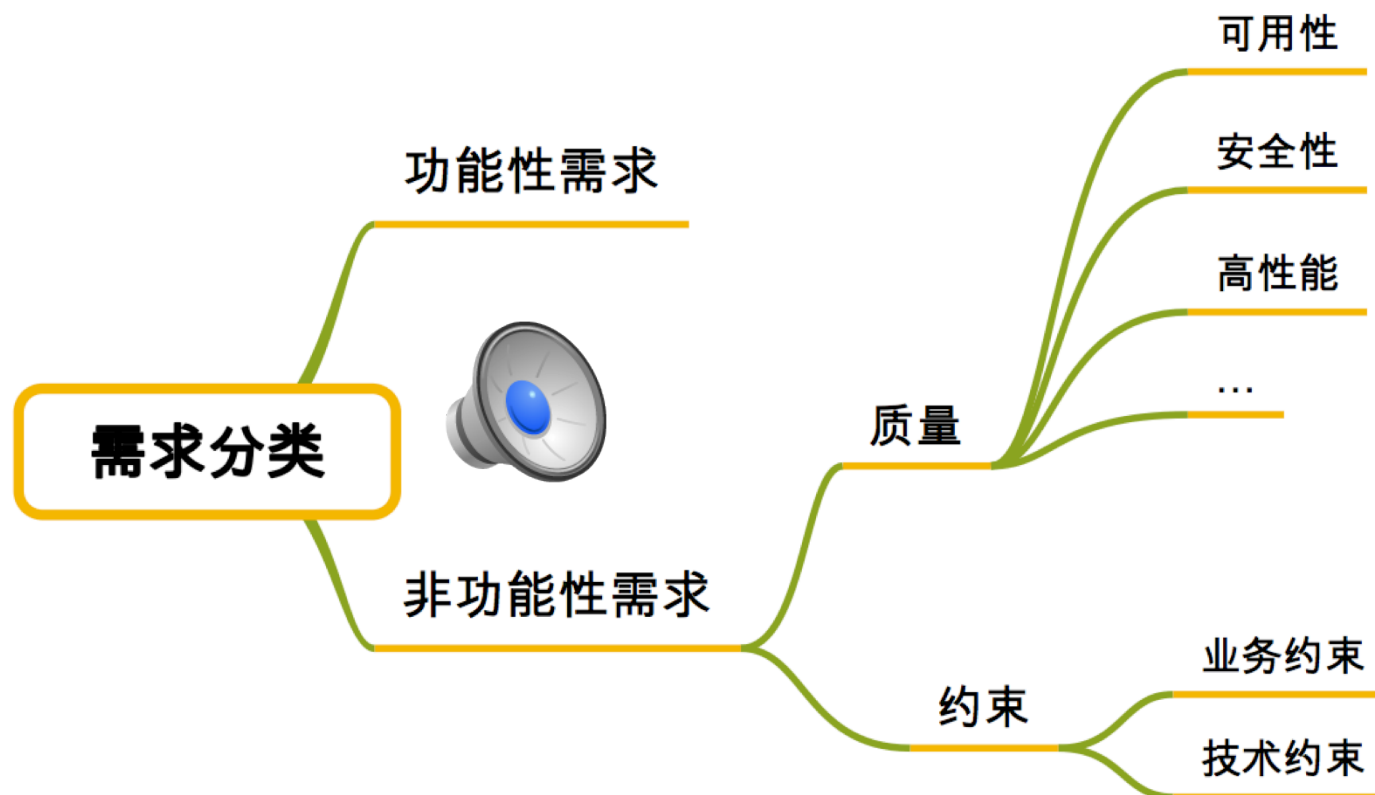
- 产品需求
- 过程需求

产品需求

- 功能性需求
- 非功能性需求

抽象层次详细程度

- 业务需求
- 用户需求
- 系统需求
- 软件设计规约

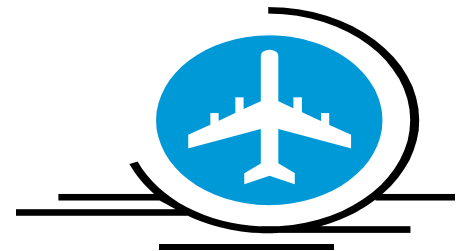


业务需求

- 业务需求是指那些可以帮助企业达成组织目标(包括策略目标)的需求项
- 企业的业务需求是关于企业业务的陈述，和这个需求如何被实现无关，无论是手动的还是通过系统来完成。
- 业务需求被叫做业务目标



- 例如：携程旅行的业务需求是卖飞机票
公司的目标是成为当人们想买飞机票时首先想到的公司



系统需求

- 系统需求的满足使得系统**实现预期的功能**，它从用户的角度描述系统在做什么，与系统是由什么硬件和软件实现无关。
- 企业选择实现系统的首要条件是：系统需求满足组织的业务需求
- 例如
 - 订票系统需要和用户数据库交互
 - **新的软件会使得汽车的启动速度加倍**
 - 我们新产品制造的低成本将会让我们有更高的市场份额并满足销售目标



软件需求



- 软件需求是指关于系统中软件部分的需求，它的满足帮助实现系统需求

举例：

- 订票系统软件通过标准的网络服务接口和航班信息交互
- 用户接口需要设置关于用户偏好的颜色和字体大小
- 系统的API需要同时支持C++和Java来让程序员访问系统服务



用户需求

系统的用户需求指的其满足会影响系统的用户接受程度的需求
有时候，用户需求被称作“用户接口需求”。



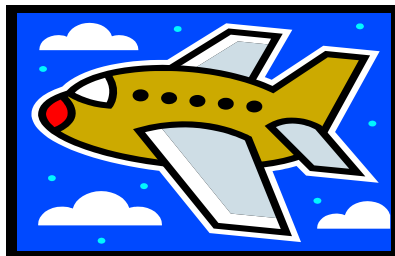
举例

- 订票系统的用户接口需要提供和系统交互的选项，通过控制命令或者WIMP接口
- 客户希望照相机背后有一个液晶屏幕，这样在拍照之后可以马上看到照片

功能性需求

系统的功能性需求是指满足系统需求需要提供的功能

有时，功能需求也被称为“行为需求”

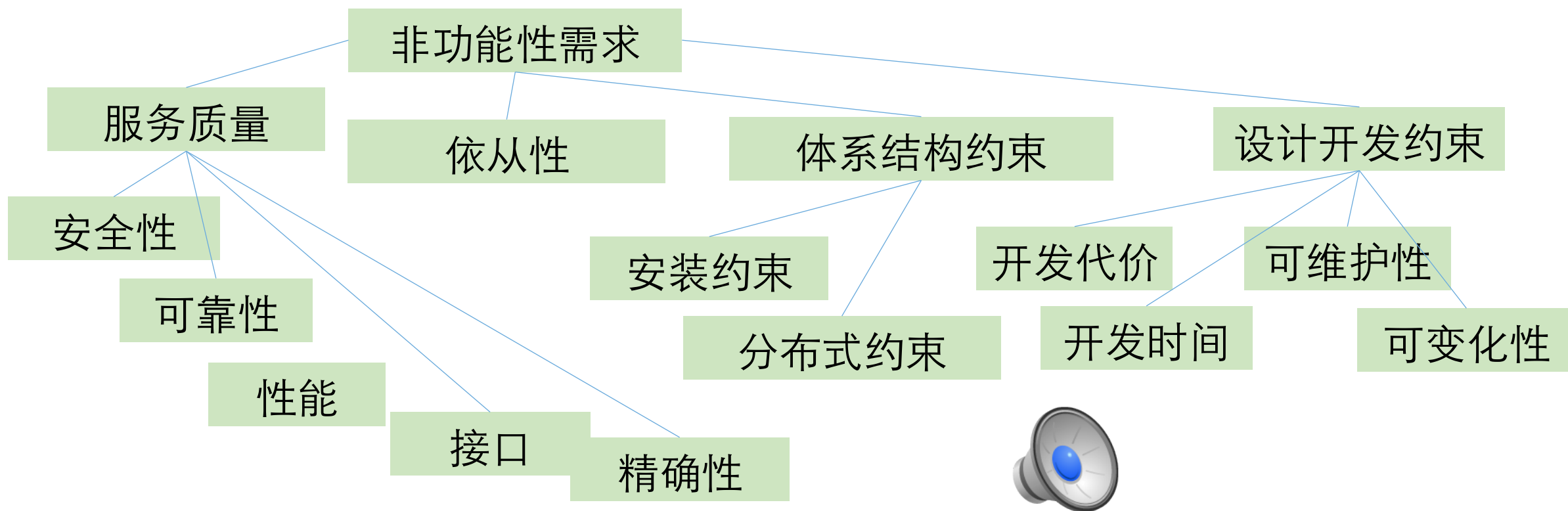


举例

- 订票系统需要提供一个在任何航班上预留座位的功能
- 订票系统需要提供一个通过信用卡付费的功能

非功能性需求 (Non-functional Requirements, NFRs)

- 非功能性需求定义软件系统以及软件开发过程为满足系统功能需求要满足的其他约束条件



质量需求

- 质量需求描述软件系统正常工作时需要满足的额外的、与质量相关的要求
- 在软件工程文献中，也称之为“**质量属性**”。
- 它应答的是关于“**提供的服务好到何种程度**”的问题（How well）

例如：

- 订票系统的用户界面要满足用户选择出发和终到站的过程不能超过四次按键
- 订票系统的订票请求响应时间要小于1分钟
- 图书馆的黑名单应随时响应馆员的查询操作
- 列车加速控制软件的平均无故障时间是 10^9 小时



依从性需求

- 依从性需求着重描述软件对国家法律、国际公约、社交法则、文化与政治习惯、标准等环境约束的满足要求

例如：

- 两列火车间的最小间距应满足国际铁路运输安全规范中的最坏情况停车距离
- 会议调度系统在缺省情况下，要排除目标市场所在地区的公众假期



体系结构设计需求

体系结构设计需求定义系统环境对待设计系统在结构上的约束

- **分布式约束**：要求软件系统组件满足目标组织由于地理自然分布导致的对系统设备节点的分布要求，以及数据的分布式存储与处理要求

例如：**会议调度系统应与分布在世界各地的参会者的邮件服务系统和电子日程管理系统协同工作**

- **安装约束**：要求软件系统能够在目标实现环境下正常运行

例如：**会议调度系统应在微软Window 8 和 Mac OS 10.10上**



设计开发约束

- 设计开发约束是对软件系统设计过程的约束
- 包括：开发成本、开发周期、产品特征的变化性、可维护性、可重用性、可移植性等。
- 例如：
 - 清华大学图书馆门户系统的开发成本不应高于200万元
 - 动车控制软件应在两年内投入使用
 - 会议日程安排系统应根据会议类型动态调整调度策略：
 - 专业研讨会vs.内部会议
 - 固定会址vs. 轮换会址
 - 参会人员的重要程度相同vs.不同



需求类型间存在一定的重叠

- 功能性需求与非功能性需求间的划分并非绝对的，可能存在一定的重叠
- 例如：在一个核电站的安全注入系统中，
 - 当反应堆正常启动或冷却时一旦发现冷却剂损失，将安全注入指示灯打开
- 这既是一个功能性需求，又是一条安全性需求
 - 防火墙管理软件的功能性需求也同样是安全型需求
 - 来电过滤常常被看做是功能性特征，同时它也和通话者的隐私需求相关
 - 对患者的病历服务发动拒绝服务攻击使得医生在手术期间无法访问患者的关键数据，这是关乎数据和患者人身双重安全的需求
 - 向列车高频发送加速请求，既是安全需求也是性能需求



需求分类的合理使用

- 关注特殊的需求特征
- 关注需求的语义特征
 - 明确指出系统必须支持的行为
 - 排除那些不可接受的系统行为
 - 明确指出系统的最好支持的行为
- 对那些适用范围受限的关注点和横切关注点区别对待
- 需求的分类主要用于为需求的抽取提供启发式的规则
 - 避免忽略某些关键的需求类型
 - 通过已知矛盾的需求类型发现具体需求间的矛盾和冲突



引出功能性需求的问题

功能

- 系统将做什么
- 系统什么时候做
- 有多种操作模式吗
- 需执行何种计算或数据转换
- 对外部刺激的响应是什么

数据

- 输入输出数据格式?
- 数据是否持久保存?



引出设计约束及过程约束的问题

物理环境

- 设备放在哪儿？
- 单节点还是多节点？
- 是否对环境有要求？ 温度、湿度、电磁干扰？
- 系统规模有何限制？
- 电源、供热或空调有何限制？
- 是否重用已有系统的构件？ 对编程语言的要求

接口

- 输入是来自一个还是多个其他系统？
- 输出是来自一个还是多个其他系统？
- 输入、输出格式是否预定义？

用户

- 谁使用系统？
- 几类用户？
- 每类用户技术水平如何？

过程

- 资源、材料
- 系统构造需要哪些人员、技能
- 多少文档？
- 在线还是本地？ 电子还是纸质
- 读者有哪些？
- 标准



引出质量需求的问题

[Roman, IEEE Computer85]:

- 性能
- 可靠性
- 安全性
- 可用性
- 可维护性
- 精度与精确性
- 交付时间及成本



- ***"-ilities"***: accessibility, adaptability, adjustability, affordability, availability, capability, compatibility, composability, comprehensibility, configurability, controllability, customizability, enhanceability, evolvability, expandability, extensibility, flexibility, inter-operability, learnability, maintainability, modifiability, portability, reconfigurability, reliability, repeatability, replaceability, reusability, scalability, standardizability, supportability, survivability, sustainability, testability, traceability, trainability, transferability, usability, variability, versatility, ...
- ***"-ities"***: additivity, distributivity, diversity, modularity, plasticity, safety, security, similarity, simplicity, ...
- ***Other***: accuracy, completeness, performance, responsiveness, user-friendliness, ...