

- 描述需求的序列图，描述的是“内外对话”。
- 描述设计的序列图，描述的是“内部协作”。

# 1、划分原理

用例是需求，架构师设计。用例驱动的架构设计，从用例到模块划分结构，关键过渡是一组对象的相互协作。所谓协作，是一组对象为了实现某个目的而进行的交互。

从“需求层”到“设计层”的总体思维路径，简称“两环节、四步骤”：

- 需求分析环节
  - 一方面用例图定义系统能提供给外部Actor的功能，此步在先。
  - 另一方面，用例规约（用例图可不够）进一步将笼统的“功能”明确定义为“能够为用户带来价值的交互序列”，但仍保持系统作为“黑盒”，此步在后。
- 架构设计环节
  - 一方面，打破“黑盒”，识别一个用例背后有哪些类，以及设计类之间的交互，此步为先。
  - 另一方面，梳理通过多个用例识别出的这些类，并将它们划分到不同模块，此步在后。

## 2、第1步：实现用例需要哪些类

### 2.1、鲁棒图-发现实现用例需要哪些类

首先，识别最明显的职责。

接下来，开始考虑职责间的关系，并发现新职责。

继续同样的思维方式，增量建模。

### 2.2、序列图-明确类之间的交互关系

画UML序列图时应注意“先大局，后细节”。

## 3、第2步：这些类应该划归哪些模块

用例驱动设计有别于所有“自顶向下”设计方法的最大不同之处，包含了“自底向上”的设计思想。这种“从好多类，到少数模块”的过程，显然是一种“归纳”思维。

# 3、优缺点

### 3.1、优点

由简入繁，从研究一个个功能的实现作为切入点，易上手。  
为后续详细设计也提供不少铺垫。

### **3.2、缺点**

用例多，从用例到类，再到模块，设计工作量大。

如果用例驱动方法运用得太死板，可能会掉入“先详细设计，后架构设计”的陷阱。