

- 如何做：对应该把需求分成哪些小的迭代来实现达成共识，要把需求拆分为可开发的粒度。
- 何时做：对完成需求的整体路线图达成共识。

第2点说的是刻意地忽略。不是所有需求都必须马上得到澄清，一开始就澄清所有需求只是一厢情愿，既没有必要，也不符合渐进认知的规律。敏捷开发方法倡导“合适的详细”这一原则，意味着可以刻意地忽略以下需求。

- 近期不会开发的低优先级需求。
- 某些尚未得到澄清，且不影响即将开发的功能的需求。

“刻意地忽略”意味着大家知道哪些事情是暂时被忽略的，这很重要。正是知道“已知的未知”，在遇到这些问题时，开发人员、产品人员和测试人员才会自发地凑在一起互相讨论，避免了大家都以为自己已经知道，于是自行其是，进而造成不必要的错误。

3.4.2 表示法

一图胜千言。本节介绍两种常用的表示操作步骤的UML符号体系：UML顺序图和UML活动图。

用UML顺序图表示操作步骤

UML顺序图擅长表示多个参与者之间存在互动的操作步骤。图3.7中的UML顺序图表示的是订餐用例的操作步骤。

UML顺序图包括如下重要元素。

- 参与者：图3.7中的同学、餐品预订系统和校园一卡通都是参与的角色。角色前的冒号遵从UML对象的命名惯例，冒号前是对象的名称，冒号后是对象的类型，即“对象名称：对象类型”。在图3.7中，由于对象名称不重要，所以予以省略。箭头代表了参与角色之间的交互情况。
- 交互：在图3.7中，从一个参与者指向另一个参与者的箭头反映了二者之间的交互。站在参与者角度，所有指向自己的箭头均反映了自己的职责或操作步骤。例如，餐品预订系统的职责就包括选择食堂、添加餐品到购物车、确认餐品列表等。如果分析的是系统，那这些职责多数会呈现为系统的一个用户界面，也可能作为一种接口接受调用。
- 参数列表：图3.7中的订餐成功（取餐码）中的（取餐码）就是参数列表。在分析过程中，写明参数是一种好习惯，因为它可能会启发出更多细节，如选择食堂（食堂ID）、添加餐品到购物车（餐品ID、餐品数量）等。