

①

迭代易管理变更的原因：

“迭代周期模型”：项目被拆分为多个短时间盒的迭代（如 Scrum 的“Sprint”，周期 2 - 4 周），每个迭代完成“需求→开发→测试→交付可工作成果”的小闭环。当变更出现时，由于每个迭代仅聚焦“部分功能”，变更只需调整当前或下一个迭代的计划，无需修改整个项目的大规模设计（对比传统瀑布模型“一次性规划所有需求，变更时需推翻前期大量工作”，迭代的调整成本极低）。如本章 PPT 里的“迭代变更影响范围示意图”也会直观显示：变更在单个迭代内的影响是局部的，这是迭代易管理变更的核心逻辑。

本章敏捷过程是否都具迭代性：

是的。本章涉及的敏捷方法（如 Scrum、极限编程 XP、敏捷统一过程等），PPT 会强调“迭代是其核心特征”。以 Scrum 为例，“Sprint 迭代流程”：每个 Sprint 包含计划会议、每日站会、评审会、回顾会，通过迭代实现“增量交付”；XP 的“小发布”实践，也把开发拆成多个小迭代周期，快速交付和调整。“迭代”是敏捷“响应变化、逐步完善”的关键载体，没有迭代的过程很难体现敏捷核心，因此本章讨论的敏捷过程几乎都具备迭代性。

一次迭代完成项目的敏捷过程是否存在：

理论上存在，但场景极特殊。“迭代的适用场景”：当项目需求 100% 明确、规模极小、复杂度可忽略时，一个迭代就能覆盖“全流程”。比如开发“简易天气查看小程序”，只需“显示当前城市天气”功能，团队能在 1 个迭代周期（如 1 周）内完成“需求分析→代码开发→测试→上线”。但现实中，多数项目（如

电商平台、管理系统）需求复杂或易变，必须多次迭代逐步构建，因此“一次迭代完成”只是极端特例。

②

假设项目作业是“在线购物平台”，最关键功能为“商品购买与支付”。结合 PPT 里“用户故事三要素（角色、功能、价值）”和“具体性要求”，则编写出的用户故事如下：

“作为一名忙碌的上班族（角色），我想要在选中心仪商品后，能快速进入下单页面，填写收货地址、选择配送时间，并通过支付宝 / 微信等多种支付方式完成付款（功能），以便我能在碎片化时间里高效完成购物，不用反复操作或因支付方式单一而耽误时间（价值）。”