



第六部分 软件实施、维护与演化

- 6.1 敏捷开发
- 6.2 软件实施、维护与演化



6.2 软件实施、维护与演化

- 软件交付工作
- 软件演化法则
- 软件维护
- 软件再工程



项目接近尾声

当开发阶段接近尾声的时候，项目组的气氛也最高涨！



项目如何交付？



项目交付工作

项目验收交付时，还有三项工组在等着：**实施、培训、验收**



验收后的项目才正式进入“维护”阶段



项目实施

项目实施是将软件系统部署到客户方的计算机系统中，协助客户准备基础数据，使软件系统顺利上线运行。

- **保证软件符合需求，质量过关**

全面做好测试工作（集成测试、功能测试、性能测试）

- **制定实施计划**

要发布的代码版本、数据库创建方式、基础数据准备方式

- **准备好程序代码和相关文档**

用户手册以及其他系统文档（如需求说明书、设计文档等）



客户培训

在系统安装完成、基础数据准备齐全之后，应该组织客户培训，使其掌握对软件系统的使用和操作。



- 选择合适的培训人员

经验丰富、了解业务和系统

- 准备好培训内容

不要临时抱佛脚

- 制定培训计划

与客户沟通协调，安排时间



项目验收

客户对系统进行验收测试，包括范围核实（用户需求是否全部实现）和质量核实（质量属性是否满足要求）。



客户在验收报告上签字，一切尘埃落定。对于大中型项目，还会有一个签字验收仪式。



6.2 软件实施、维护与演化

- 软件交付工作
- 软件演化法则
- 软件维护
- 软件再工程



软件变化原因

软件的变化是不可避免的

- 软件在使用过程中，新的需求不断出现
- 商业环境在不断地变化
- 软件中的缺陷需要进行修复
- 计算机硬件和软件环境的升级需要更新现有的系统
- 软件的性能和可靠性需要进一步改进

关键：采取适当的策略，有效地实施和管理软件的变化！



Lehman 法则

持续变化	在用的程序持续地经历变化，或逐渐变得不可用。
递增复杂性	程序的不断修改将导致结构恶化，增加了复杂性。
程序演化法则	程序演化服从统计上的确定趋势和恒定性。
组织稳定守恒	编程项目总体活动统计上是不变的。
熟悉程度守恒	后续发行对于整个系统功能不会产生很大改变。



软件演化策略

软件维护

为了修改软件的缺陷或者增加新功能而对软件进行修改；软件的修改通常发生在局部，一般不会改变整个结构。

软件再工程

为了避免软件本身退化而对软件的一部分进行重新设计、编码和测试，以便提高软件的可维护性和可靠性等。



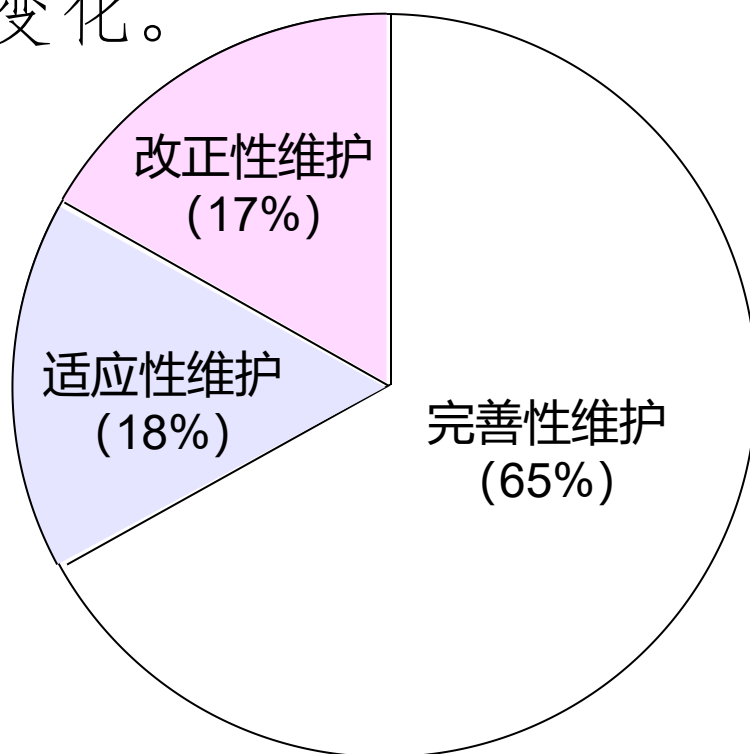
6.2 软件实施、维护与演化

- 软件交付工作
- 软件演化法则
- 软件维护
- 软件再工程



软件维护类型

软件维护是软件被投入运行使用后人们对软件产品所进行的修改，变更通常是修改现有的组件或增加新的组件，一般不涉及体系结构的重大变化。

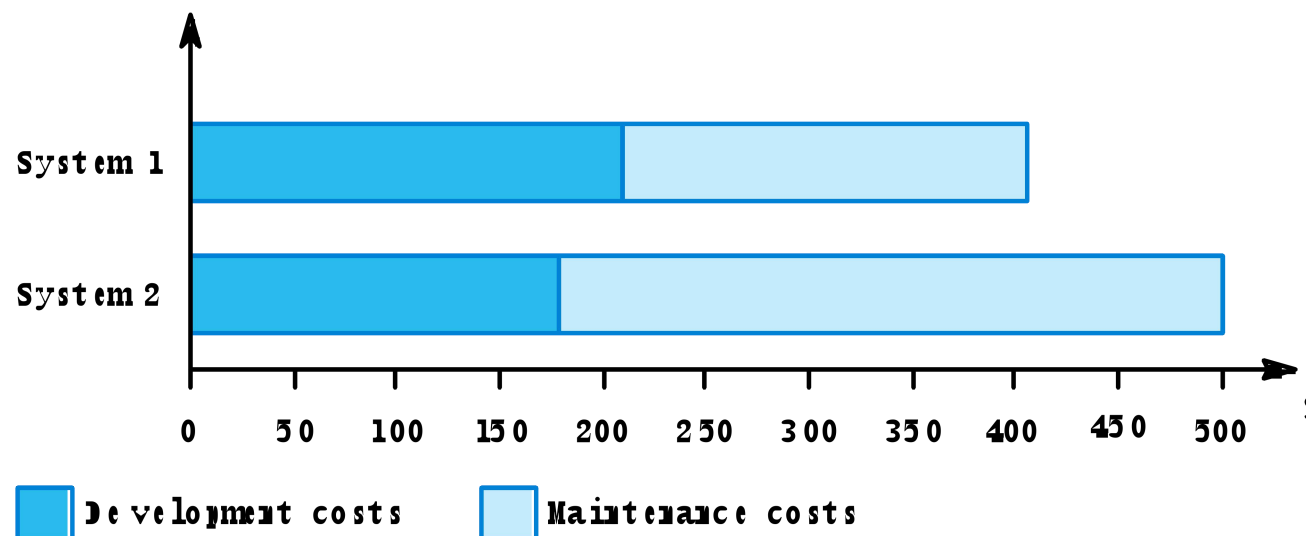


- **改正性维护**：修改软件缺陷或不足
- **适应性维护**：修改软件使其适应不同操作环境，主要包括硬件变化、操作系统变化或者其他支持软件变化等
- **完善性维护**：增加或修改系统功能，使其适应业务的变化

软件维护成本

软件维护成本是很昂贵的

- 业务应用系统：维护费用与开发成本大体相同
- 嵌入式实时系统：维护费用是开发成本的四倍以上





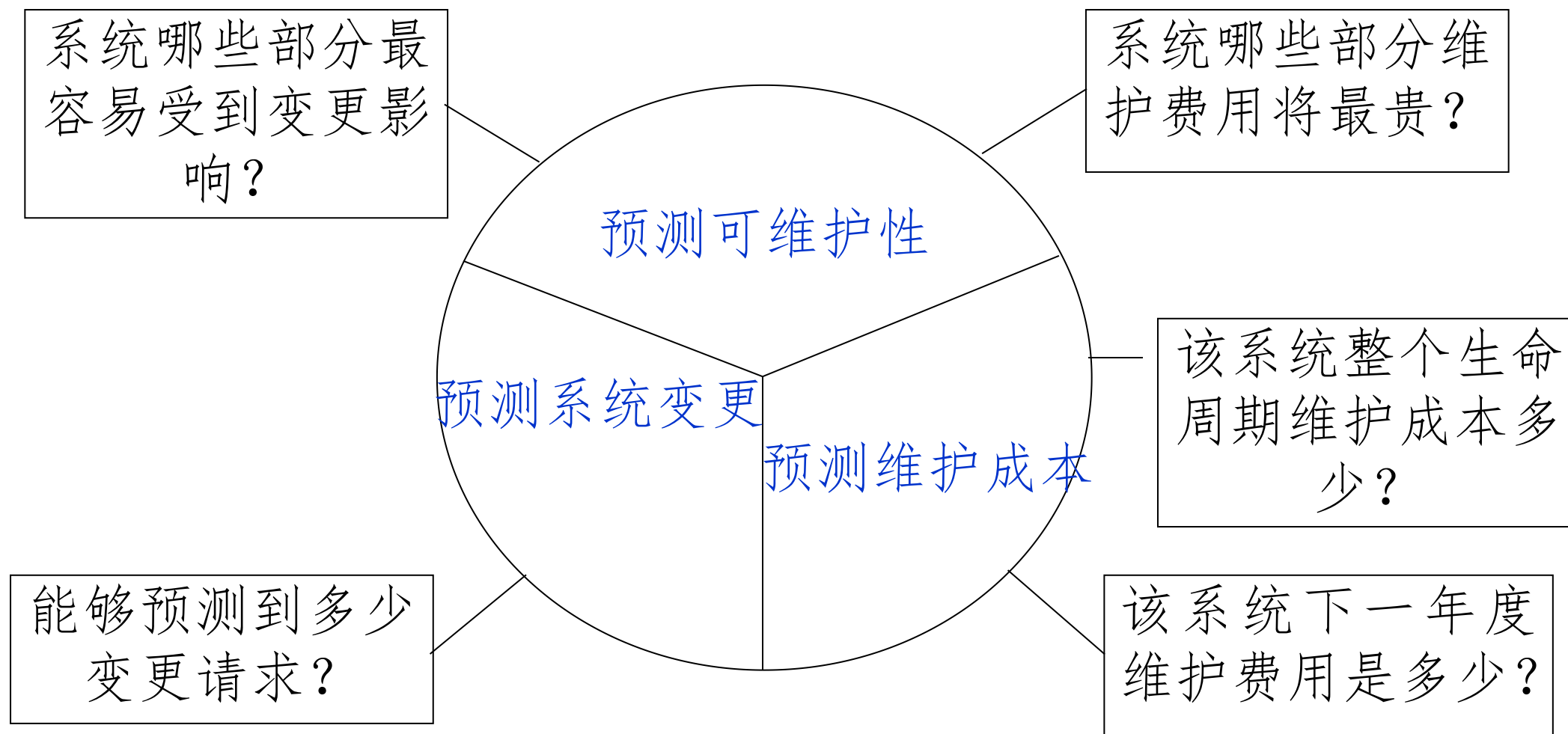
影响维护成本的因素

- **团队稳定性**：系统移交后开发团队会解散，人员分配到其他项目中，负责维护的人员通常不是原开发人员，需要花时间理解系统。
- **合同责任**：维护合同一般独立于开发合同，这样开发人员有可能缺少为方便维护而写软件的动力。
- **人员技术水平**：维护人员有可能缺乏经验，而且不熟悉应用领域。
- **程序年龄与结构**：程序结构随年龄增加而受到破坏，不易理解和变更。



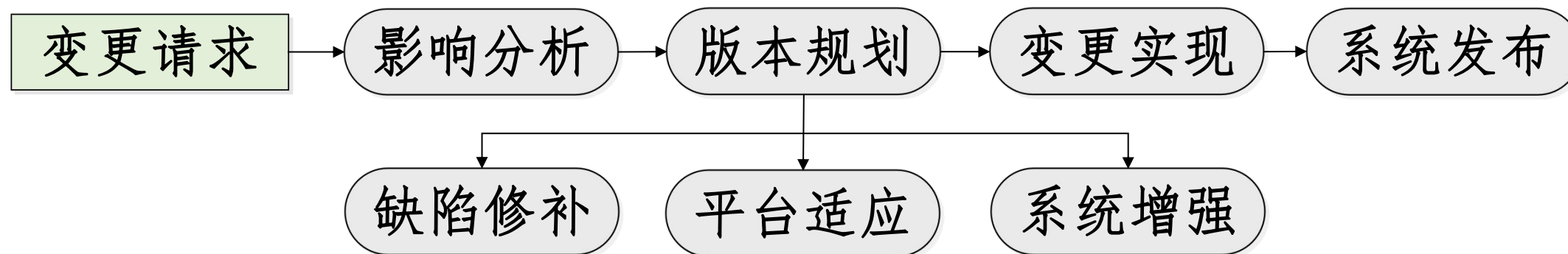


软件维护预测

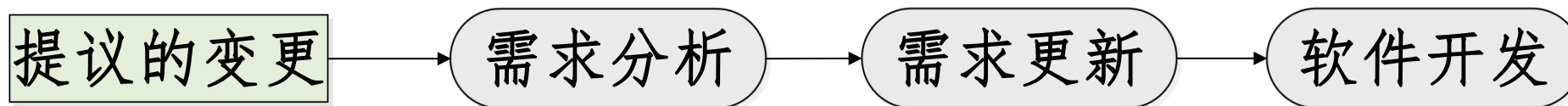




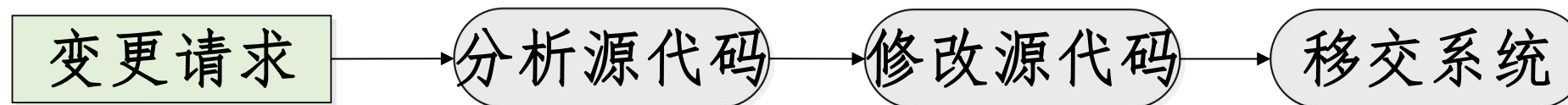
软件维护过程



变更
实现



紧急
修补





6.2 软件实施、维护与演化

- 软件交付工作
- 软件演化法则
- 软件维护
- 软件再工程



软件再工程

软件再工程

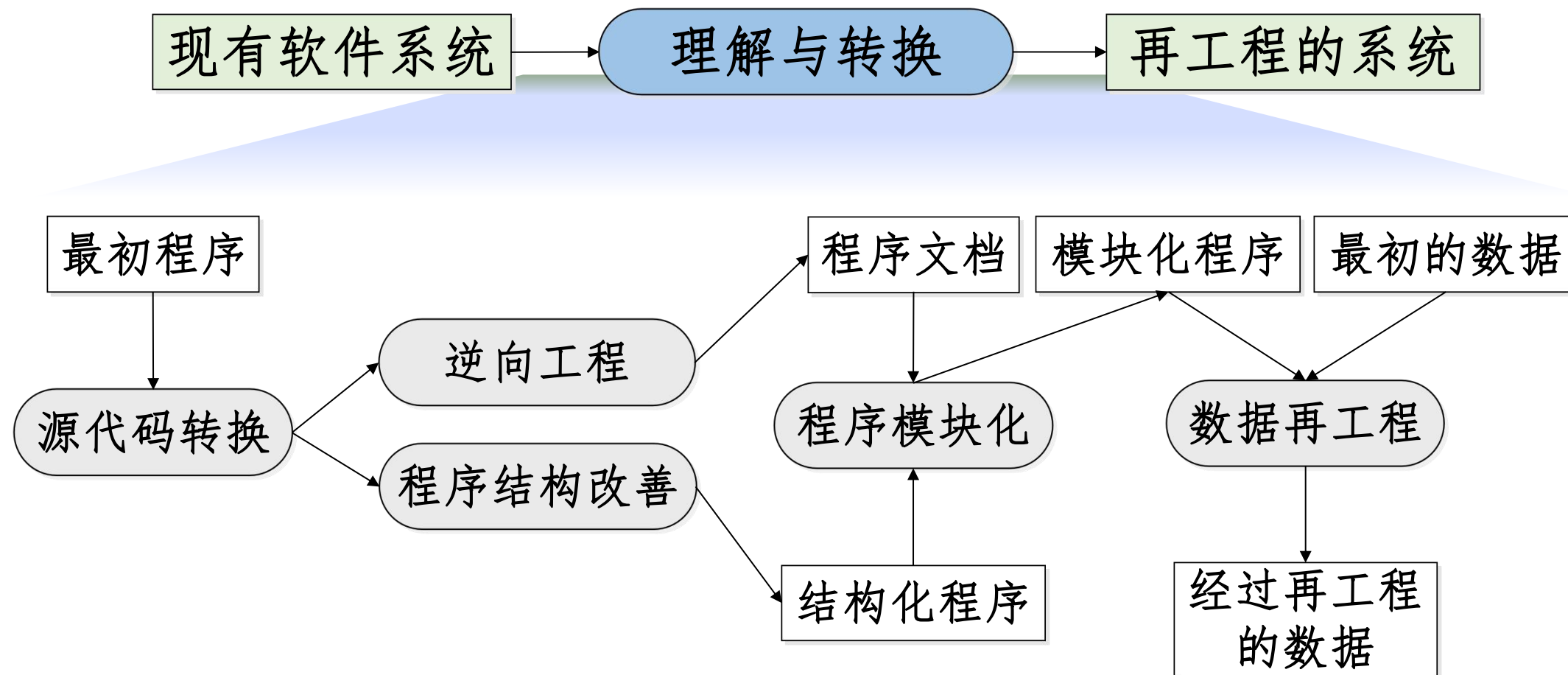
- 重新构造或编写现有系统的一部分或全部，但不改变其功能
- 在大型系统中某些部分需要频繁维护时，可应用软件再工程
- 再工程的目的是努力使系统更易于维护，系统需要被再构造和再文档化

优势：

- 减少风险：重新开发一个在用的系统具有很高的风险，可能会有开发问题、人员问题和规格说明问题
- 降低成本：再工程的成本比重新开发软件的成本要小得多

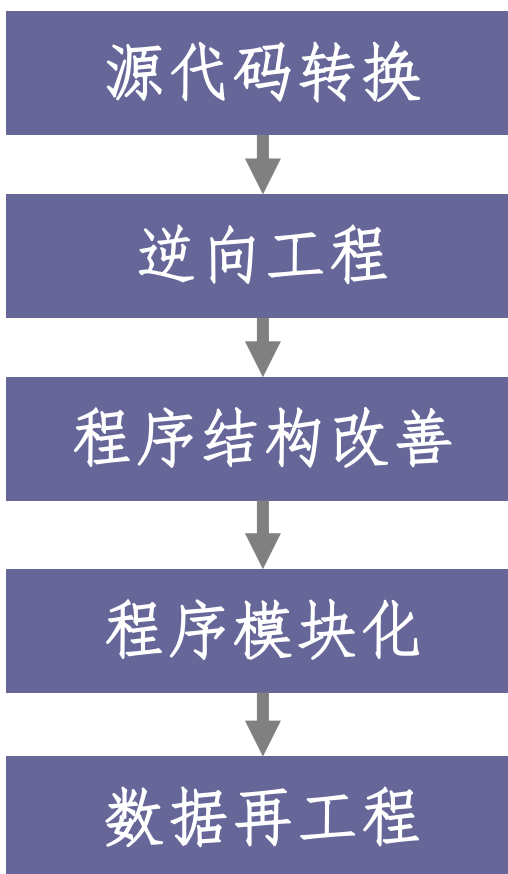


再工程过程





再工程过程



代码从原有的程序设计语言转换到一种新语言

分析程序并抽取信息记录其结构和功能

分析和修改程序的控制结构，使其更易读和好理解

重新组织程序的结构

改变程序处理的数据以反映程序的变更

逆向工程

- 逆向工程是以复原软件的规格说明和设计为目标的软件分析过程
- 大多数情况下，逆向工程弥补缺乏良好文档的问题
- 开发阶段的文档与维护阶段的文档可能是不一致的
- 开发阶段编写的程序文档在维护阶段是非常有用的

