

# 统一过程UP



- Ivar Jacobson, Grady Booch, James Rumbaugh  
统一方法
- Unified Process, Rational Unified Process
- 统一建模语言UML
- 融合传统软件过程模型的优点（混合过程模型）

# 统一过程UP

## 用例驱动

- 与客户沟通
- 从用户的角度描述系统

## 以架构为核心

- 强调体系结构的重要性

## 迭代并且增量

- 融合迭代和增量两种过程流的优点

# UP的三个视图

**动态视图**

• **分阶段（4）**

**静态视图**

•  **workflow（9）**

**实践视图**

• **最佳实践（6）**

# UP四个阶段（Life Cycle Phases）

开始

## Inception

- 与涉众合作，定义系统的业务需求（使用**用例**）
- 提出系统大致的架构（子系统及其功能）
- 制定开发计划

## Elaboration

- 精化、扩展初始用例
- 扩展体系结构（**分析、设计模型**）
- 评审、修订**项目计划**

# UP四个阶段

构建

Construction

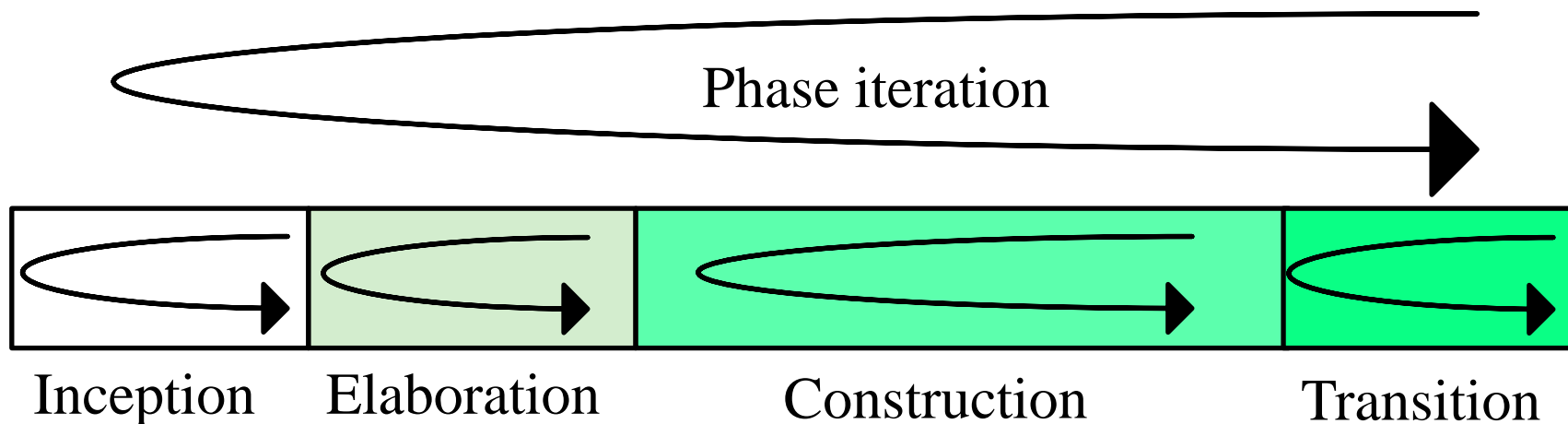
- **系统**开发和测试
- 基于分析、设计模型实现构件
- 单元测试、集成测试、验收测试

交付

Transition

- 在运行环境中部署系统
- 准备用户手册、问题指南、安装手册等

# UP四个阶段

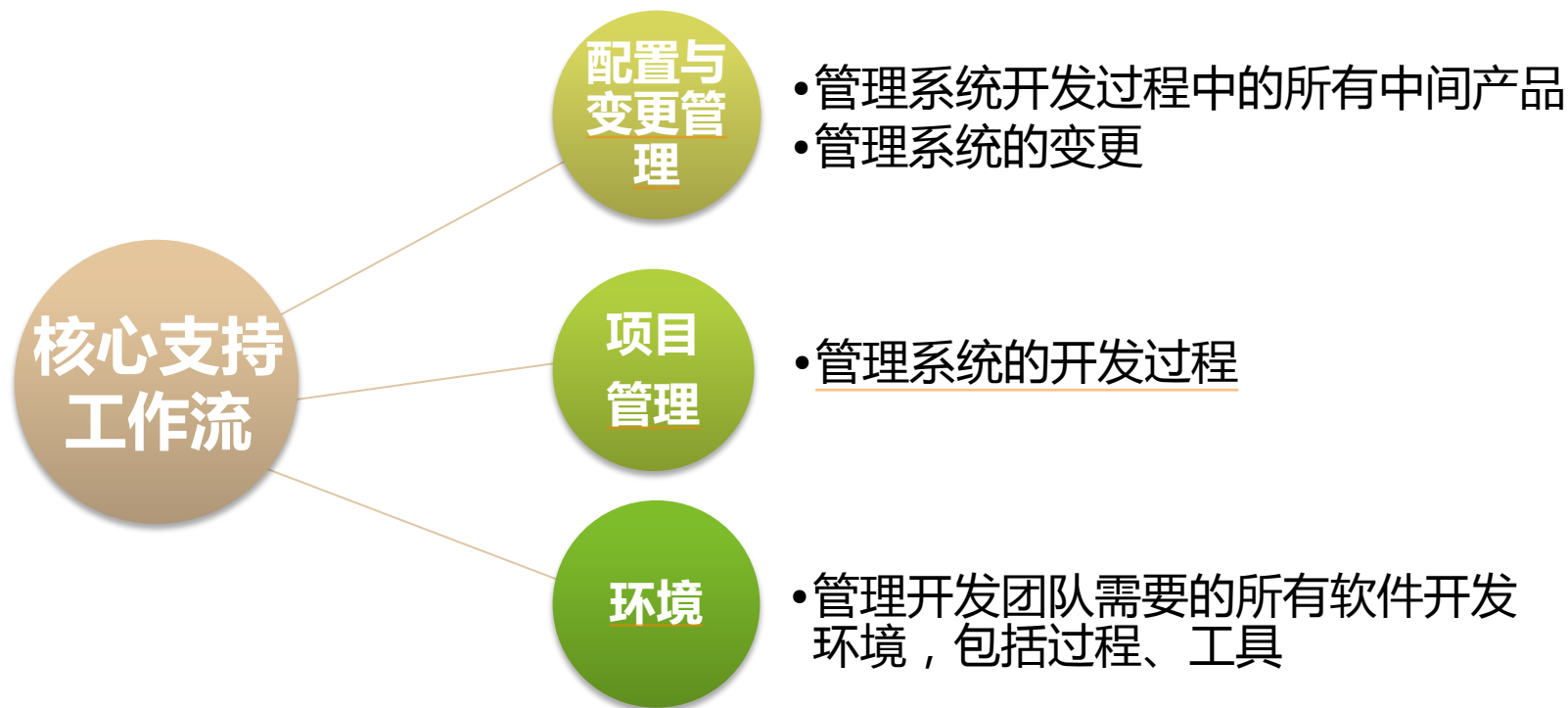


- 阶段内迭代 ( In-phase iteration )
- 跨阶段迭代 ( Cross-phase iteration )

# UP workflow (Workflow)

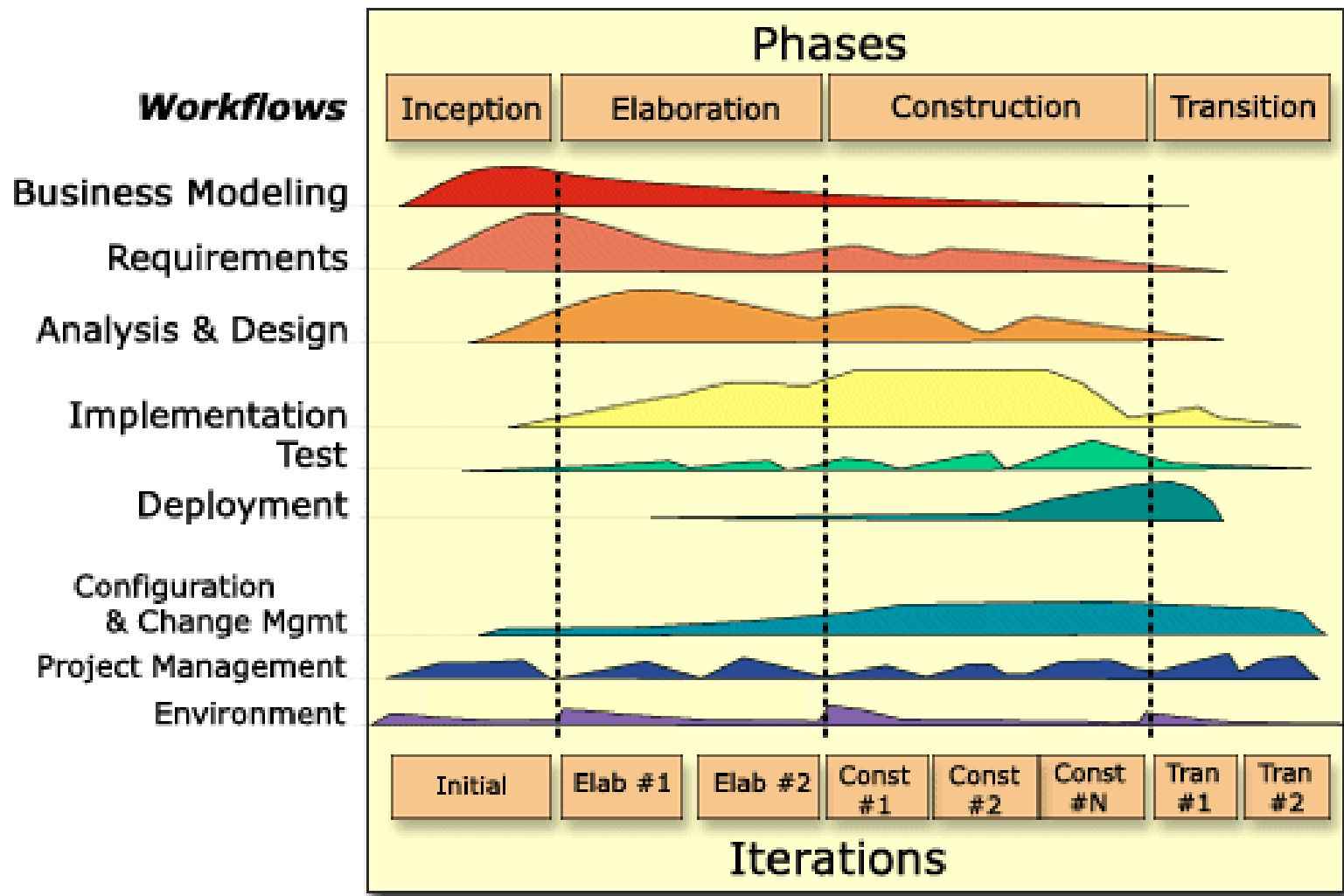


# UP workflow (Workflow)





# UP迭代模型图



# UP 最佳实践 (Best Practices)

UP所使用的最佳实践

最佳实践：以一种循环的、可预测的方式开发和维护高质量的软件产品

- 迭代式开发(Develop software iteratively)
  - 一系列细化、若干个渐进的反复过程而得出有效解决方案的迭代方法
  - 先开发优先级高的系统特性
- 管理需求 ( Manage requirements )
  - 获取用户的需求并把它们文档化
  - 跟踪需求的变更
- 使用基于构件的体系结构 ( Use component-based architectures )
  - 使用一系列可复用的构件来组织系统的体系结构

# UP 最佳实践

- 可视化建模 ( Visually model software )
  - 使用UML建立系统的可视化模型
  - 构件及构件间的关系
- 验证软件质量 ( Verify software quality )
  - 软件质量评估内建在贯穿于整个开发过程的所有活动中
  - 确保软件产品满足组织的质量标准
- 控制软件变更 ( Control changes to software )
  - 使用变更管理系统和配置管理工具管理软件的变更

# 总结

- No single “best” lifecycle
  - 取决于项目的细节和约束
- 通常使用某种形式的迭代方法
  - 2000年，美国军方软件开发标准（DOD 5000.2）推荐**迭代**  
**为软件开发优选模式**
- 在大项目里组合不同的生命周期模型