

软件过程及过程模型

软件过程 (Software Process)

- 软件过程是我们生产软件的方式。
- 软件工程实践是一系列原则、概念、方法和工具，这些都需要在软件计划和开发过程中考虑。
- 一个具体的软件工程实践需要根据情况定制。

软件生命周期模型 (Software Life-cycle Model)

- 生命周期模型是描述构建软件产品时应执行的步骤的描述。
- 所有软件开发方法在两个步骤上一样：获取需求和交付可运行的系统。

过程模型与生命周期模型

- 过程模型和生命周期模型是初学者容易混淆的概念。
- 过程模型以人为主体的，从开发过程来看；生命周期模型则从软件自身来看。
- 两者的本质是一致的，从不同视角看待软件开发。

生命周期模型类型

- 瀑布模型 (Waterfall)
- 进化树模型 (Evolution-Tree)
- 增量模型 (Incremental)

基线 (Baseline)

- 在每个情节的结尾，都有一个基线，即一套完整的工件。
- 工件是软件产品的组成部分。

瀑布式生命周期模型

- 瀑布模型可以代表Winburg案例研究，但无法显示事件的顺序。

增量和迭代

- 增量指产品功能或模块上的增加。
- 迭代指增量部分仍然使用相同的流程来开发。

五个核心 workflow

- 需求 workflow
- 分析 workflow
- 设计 workflow
- 实现 workflow
- 测试 workflow

通用过程模型

- 定义了五个框架活动：通信、计划、建模、构建、部署。
- 以及一系列在整个过程中应用的保护活动，如项目跟踪和控制、风险管理、质量保证、配置管理和技术审查。

关键概念

- 框架活动、软件工程行动、任务集、软件工作任务、工作产品、质量保证点、项目里程碑。

规范性生命周期模型

- 代码和修复生命周期模型 (Code-and-Fix)
- 瀑布生命周期模型 (Waterfall)
- 快速原型生命周期模型 (Rapid-Prototyping)
- 螺旋生命周期模型 (Spiral)

模型	模型的维度问题	易用性	对问题的刻画能力
Code-and-Fix	没有清晰阶段，编码与修复交替进行	易用性低	刻画能力弱，无法应对复杂需求或变化
Waterfall	线性、顺序的开发过程，阶段划分清晰	易用性中等	刻画能力强，适用于需求明确且稳定的项目

Rapid-Prototyping	快速原型迭代，多次反馈与调整，灵活性高	易用性高	刻画能力强，适应需求变化和不确定性
Spiral	迭代与风险评估结合，强调反馈与调整	易用性中	刻画能力强，适用于复杂、风险高、需求不稳定的项目