

- 软件开发生命周期指软件产品从开发到报废的生命周期，通常周期中包括了需求分析、软件设计、实现与调试、测试与验收、部署、维护等活动。
- 软件报废：不需要其功能；很难在合理的成本范围内继续演进开发原有软件。

- 简称软件过程。
- 软件开发过程是指一个软件产品开发的方法，它描述了软件开发中的活动和任务。简单的说，过程就是软件开发中的一系列活动，如果能够按照这些活动进行工作，我们就可以获得预想的结果。
- “过程决定质量”

- 瀑布模型核心思想是根据开发活动来分解项目。瀑布模型将软件生命周期划分为制定计划、需求分析、软件设计、程序编写、软件测试和运行维护等六个基本活动。
- 瀑布模型设计了一系列软件开发阶段，按时间顺序展开开发活动，从需求分析开始直到产品发布和维护，每个阶段都会产生循环反馈。如果在某个阶段出现了问题，那么最好“返回”上一个阶段并进行修改。项目开发过程像是从一个阶梯“流动”到下一个阶梯，这也是瀑布模型名称的由来。

- 问题2：在瀑布模型中，开发人员很难判断其前期的工作是否正确。计划、需求分析和设计中的错误往往要到产生了测试、集成时才能够发现，而在前期，我们没有足够的信息来进行这种判断。

- 按照软件工程中的贝姆定律（Boehm's Law），在开发过程中越晚修正缺陷，代价就会越高。

- 迭代式软件开发根据软件项目的不同功能子集来分解项目。
- 在迭代式软件开发中，整个开发工作被组织为一系列小的项目，被称为一系列的迭代。
- 每个迭代周期结束时都应该得到一个经过测试的，集成起来的基本可用的软件产品，某些少量程序缺陷的修复或用户培训可以放在最后一个迭代周期后进行。

(1) 易于应对需求变更

(2) 提升团队士气

- PSP能够提供：

- 1、个体软件过程原则；
- 2、软件开发工程师如何制定准确的计划；
- 3、软件工程师为改善产品质量需要采取的措施；
- 4、度量个体软件过程的改进；
- 5、流程的改变对软件工程师个人能力的影响。

- PSP是包括了数据记录表格、过程操作指南和规程在内的结构化框架。

PSP基本原则

- 软件系统的整体质量由该系统中质量最差的某些组件所决定；
 - 软件组件的质量取决于开发这些组件的软件工程师，更加确切的说，是由这些工程师所使用的开发过程所决定；
 - 作为合格的软件工程师，应当自己度量、跟踪自己的工作，自己管理软件组件的质量；
 - 作为合格的软件工程师，应当从自己开发过程的偏差中学习、总结，并将这些经验教训整合到自己的开发实践中，也就是说，应当建立持续地自我改进机制。
- 1、将系统、规范、可度量的方法应用于软件的开发、运行和维护，即将工程应用于软件；
 - 2、对1中所述方法的研究。

发布方式：

- ①本机软件。
- ②本机和远程相结合软件。
 - C/S架构
 - AppStore, Google Play
- ③远程软件。B/S架构

项目评审

- 对项目进行评估审核的过程，评审会议。
- 开发人员演示软件功能，其他人员就自己关心的问题提问，双方进行详细讨论。
- 以功能为核心。

- 瀑布模型可以帮助开发人员理解软件开发中的活动和任务，界定清晰的软件开发检查点。
- 问题1：在实践中很少有项目能够以纯线性的方式进行，通过回到前面的阶段或修改前一个阶段的结果是非常常见的现象，而这时带来的成本增加和系统开发的混乱是很难避免的。
- 问题2：开发人员很难判断其前期的工作是否正确，导致后期更改。
 - Boehm's Law: 在开发过程中越晚修正缺陷，代价就会越高。
- 问题3：需求误解