# 软件工程名词解释高频题一览

说明前面加\*表示只是理论上会考，有时一个名词会多次出现，有些英文不一样（如过程），则是两个概念。如果英文也一样，看书上定义，如果定义差不多（例如原型和抽象）答哪个都行，如果定义不一样则看情况或者都答   
适用于《软件工程 第4版》人民邮电出版社

第一章：   
软件工程（SE）   
分析（analysis）   
合成（synthesis）   
工具（tool）   
过程（procedure）   
泛型（paradigm）   
错误（error）   
故障（fault）   
失效（failure）   
\*客户（custom）   
\*开发者（developer）   
\*用户（user）   
系统（system）   
活动（activity）   
\*实体（entity）   
\*关系（relationship）   
系统边界（system boundary）   
抽象（abstraction）   
原型（prototype）   
复用、重用（reuse）

第二章   
过程（process）   
阶段化开发（Phased development）   
循环周期（cycle time）   
运行系统（operation system）   
开发系统（development system）   
增量开发（incremental development）   
迭代开发（iterative development）   
极限编程（XP）   
水晶法（crystal）   
并列争球法（scrum）   
统一过程（UP/RUP）

第三章   
项目进度（project schedule）   
可交付产品（deliverable）   
活动（activity）   
里程碑（milestone）   
风险（risk）   
风险影响（risk impact）   
风险概率（risk probability）   
风险暴露（risk exposure）

第四章   
需求（requirement）   
软件规格说明（SRS，software requirement specification）   
功能性需求（functional requirement）   
非功能性需求（nonfunctional requirement）   
设计约束（design constraint）   
过程约束（process constraint）   
需求定义（requirement definition）   
需求规格说明（requirement specification）（实际上跟SRS一样）   
用例（use case）百度百科：对一组动作序列的抽象描述，系统执行这些动作序列，产生相应的结果。

第五章   
设计（design）   
体系结构（architecture）   
体系结构风格（architecture style）   
例程设计（routine design）   
克隆（clone）   
参考模型（reference model）   
设计公约（design convention）   
设计模式（design pattern）   
设计原则（design principle）   
\*创新设计（innovation design）

第六章   
\*重构（refactoring）   
设计原则（design principle）区别于系统设计中设计原则   
模块化（Modular）   
模块（Module）

耦合（coupled）   
非耦合（uncoupled）   
数据耦合（data coupling）   
标记耦合/特征耦合（stamp coupling）   
控制耦合（control coupling）   
公共耦合（common coupling）   
内容耦合（content coupling）

内聚（cohesion）   
偶然内聚（coincidental cohesion）   
逻辑内聚（logical cohesion）   
时间内聚/时态内聚（temporal cohesion）   
过程内聚（procedural cohesion）   
通信内聚（communicational cohesion）   
顺序内聚（squential cohesion）   
功能内聚（functional cohesion）   
信息内聚（information cohesion）

接口（interface）   
面向对象一系列名词

第七章   
结对编程/派对编程（pair programming）

第八章   
故障（fault）   
失效（failure）和第一章一样但是定义更准确   
算法故障（algorithm fault）   
计算故障（computation fault）/精度故障（precision fault）   
文档故障（documentation fault）   
压力故障（stress fault）/过载故障（overload fault）   
能力故障（capacity fault）/边界故障（boundary fault）   
计时故障（timing fault）   
吞吐量故障（throughput fault）/性能故障（performance fault）   
硬件和系统软件故障（hardware and system software fault）

标准过程故障（standards and procedure fault）   
正交缺陷分类（orthogonal defect classification）

模块测试（module testing）/构建测试（component testing）/单元测试（unit testing）   
集成测试（integration testing）

\*忘我编程（egoless programming）   
黑盒测试（black box test）/闭盒测试（close box test）   
白盒测试（white box test）/开盒测试（open box test）   
代码评审（code review）   
代码走查（code walkthrough）   
代码审查（code inspection）

语句覆盖（Statement coverage）   
分支覆盖/判定覆盖（Branch coverage）   
条件覆盖（condition coverage）   
路径覆盖（path coverage）   
以上四条区分理解即可

自底向上测试（bottom-up）   
自顶向下测试（top-bottom）   
驱动程序（component driver）   
桩（stub）

第九章   
功能测试（function test）   
性能测试（performance test）   
验收测试（acceptance test）   
安装测试（installation test）   
系统配置（system configuration）   
配置管理（configuration management）   
版本（version）   
发布（release）   
基线（base line）稳定可用的一个版本（包括程序和文档），是后续开发的基础   
回归测试（regressive test）

压力测试（stress test）   
容量测试（volume test）   
配置测试（configuration test）   
兼容性测试（compatible test）   
安全性测试（security test）   
\*环境测试（environment test）   
\*计时测试（timing test）   
质量测试（quality test）   
恢复测试（recovery test）   
\*维护测试（maintenance test）   
\*文档测试（documentation test）

基准测试（benchmark test）   
试验性测试（pilot test）   
α测试（alpha test）   
β测试（beta test）   
并行测试（parallel test）