

## 软件工程的定义

- 软件工程是指导计算机软件开发和维护的一门工程学科。
- 采用工程化的方法来开发和维护软件，把经过实践考验而证明正确的工程管理技术和优秀的技术方法结合起来，开发出高质量的软件并能有效的维护学科领域。

## 解释导致 software crisis 本质原因、表现，述说克服软件危机的方法

原因：

- 大容量、高速度计算机的出现，使计算机的应用范围迅速扩大
- 软件开发规模越来越大
- 软件开发复杂度越来越高
- 用户需求不明确
- 个人的设计以及应用场景实例的不丰富不能符合当前需求。

表现：

- 软件开发进度难以预测
- 软件开发成本难以控制
- 用户对产品功能难以满足
- 软件产品质量无法保证
- 软件产品难以维护
- 软件缺少适当的文档资料

克服软件危机的方法：

- 通过吸收和借鉴人类长期以来从事各种工程项目中积累的经验，以此建立与系统化软件生产有关的概念、原则、方法、技术和工具，规范行业。
- 推广在实践中总结出来的开发软件的成功的技术和方法，并研究更好、更有效的技术和方法，尽快克服在计算机系统早期发展阶段形成的一些错误概念和作法。
- 改进软件系统的生产活动，以期达到降低软件生产成本、改进软件产品质量、提高软件生产率水平的目标。

- 研制和使用软件工具，用以辅助进行软件项目管理与技术生产，将软件生命周期各阶段使用的软件工具有机地集合成为一个整体，形成能够连续支持软件开发与维护全过程的集成化软件支援环境，以期从管理和技术两方面解决软件危机问题。

## 软件生命周期

软件生命周期又称为软件生存周期或系统开发生命周期，是软件的产生直到报废的生命周期。

- 问题的定义以及规划  
明确用户需求的方向范围
- 需求分析  
明确用户对软件系统详尽的需求，编修需求规格说明书和用户手册并提交评审
- 可行性研究
- 软件设计
  - 1) 概要设计（基本框架）
  - 2) 详细设计（细分模块）
- 程序编码
- 软件测试
- 运行维护

## SWEBoK 的 15 个知识域（An Overview of the SWEBoK Guide 请中文翻译其名称与简短说明）

- Software Requirements 软件需求：用来描述解决现实世界某个问题的软件产品及对软件产品的约束，设计需求获取、需求分析、建立需求规格说明和确认，领域问题建模，软件开发技术、经济和时间可行性分析。软件需求的好坏直接影响软件开发全过程。
- Software Design 软件设计：软件工程最核心的内容。软件设计由软件体系结构设计、软件详细设计两种活动组成。它涉及软

件体系结构、构件、接口，还设计软件设计质量分析和评估、软件设计的表示、软件设计策略和方法。

- **Software Construction 软件构造**：通过编码、单元测试、集成测试、调试、确认等活动，生成可用的、有意义的软件。软件构造除要求软件设计功能外，还要求控制和降低程序复杂性、预计变更、进行程序验证和制定软件构造标准。软件构造与软件配置管理、工具和方法。
- **Software Testing 软件测试**：软件生存周期的重要部分，涉及测试标准、技术、度量和测试过程。测试的目的是标识缺陷和问题，改善产品质量。软件测试覆盖整个软件开发过程。正确的软件工程质量观是预防、避免缺陷和问题。测试的终点是建立一个有限的测试用例集，动态地验证程序是否达到预期行为。
- **Software Maintenance 软件维护**：软件生存周期的组成部分。软件维护要支持系统快速地、便捷地满足新的需求。基于服务的软件维护越来越受到重视。软件组织力图是软件运营时间更长，软件维护成为令人关注的焦点。
- **Software Configuration Management 软件配置管理**：是一种标识、组织和控制修改的技术，维护整个系统生命周期中软件配置的一致性和可追踪性。内容包括配置管理过程的管理、软件配置鉴别、配置管理控制管理、配置管理状态记录、配置管理审计、软件发布和交付管理。
- **Software Engineering Management 软件工程管理**：运用管理活动，确保软件开发和维护是系统的、规范的、可度量的。它设计基础设施管理、项目管理、度量和控制计划三个层次。度量是软件管理决策的基础。
- **Software Engineering Process 软件工程过程**：生产一个最终能满足用户需求且达到工程目标的软件产品所需要的步骤。软件工程过程主要包括开发过程、运作过程、维护过程。它们覆盖了需求、设计、实现、确认以及维护等活动。
- **Software Engineering Models and Methods 软件工程模型与方法**：建模、模型类型、分析、和软件开发方法。

- **Software Quality 软件质量：**涉及软件质量需求、软件质量度量、软件属性检测、软件质量管理技术和过程等。
- **Software Engineering Professional Practice 软件工程专业实践：**关注软件工程师必须具备的专业，负责和符合伦理的软件工程知识，技能和态度。
- **Software Engineering Economics 软件工程经济学：**关注在业务环境中做出决策，以使技术决策与组织的业务目标保持一致。
- **Computing Foundations 计算基础：**提供软件工程实践所需的计算背景。
- **Mathematical Foundations 数学基础：**提供软件工程实践所需的数学背景。
- **Engineering Foundations 工程基础：**提供软件工程实践所需的工程背景。

## **简单解释 CMMI 的五个级别。例如：Level 1 - Initial：无序，自发生产模式。**

- **L1：CMMI一级，完成级。**在完成级水平上，企业对项目的目标与要做的努力很清晰。项目的目标得以实现。因此，任务是完成了。但是由于任务的完成带有很大的偶然性，企业无法保证在实施同类项目的时候仍然能够完成任务。企业在一级上的项目实施对实施人员有很大的依赖性。
- **L2：CMMI二级，管理级。**在管理级水平上，企业在项目实施上能够遵守既定的计划与流程，有资源准备，权责到人，对相关的项目实施人员有相应的培训，对整个流程有监测与控制，并与上级单位对项目与流程进行审查。企业在二级水平上体现了对项目的一系列的管理程序。这一系列的管理手段排除了企业在一级时完成任务的随机性，保证了企业的所有项目实施都会得到成功。
- **L3：CMMI三级，定义级。**在定义级水平上，企业不仅仅能够对项目的实施有一整套的管理措施，并保障项目的完成；而且，企业能够根据自身的特殊情况以及自己的标准流程，将这套管

理体系与流程予以制度化。这样，企业不仅能够在同类的项目上得到成功的实施，在不同类的项目上一样能够得到成功的实施。科学的管理成为企业的一种文化，企业的组织财富。

- L4: CMMI四级，量化管理级。在量化管理级水平上，企业的项目管理不仅仅形成了一种制度，而且要实现数字化的管理。对管理流程要做到量化与数字化。通过量化技术来实现流程的稳定性，实现管理的精度，降低项目实施再质量上的波动。
- L5: CMMI五级，优化级。在优化级水平上，企业的项目管理达到了最高的境界。企业仅仅能够通过信息手段与数字手段来实现对项目的管理，而且能够充分利用信息资料，对企业在项目实施的过程中可能出现的次品予以预防。能够主动地改善流程，运用新技术，实现流程的优化。

### **用自己语言简述 SWEBoK 或 CMMI （约200字）**

- CMMI (Capability Maturity Model Integration) 中文翻译：能力成熟度模型集成。是一种管理流程的标准化。遵循该模型的标准，就能够在管理上迈出一大步。其目的是帮助软件企业对软件工程过程进行管理和改进，增强开发与改进能力，从而能按时地、不超预算地开发出高质量的软件。
- CMMI主要关注点就是成本效益、明确重点、过程集中和灵活性四个方面，相对应的有五个不同的标准。而每一个标准对企业的管理力度都有着不同的要求。企业可以改进管理模式，不断地提高自己的CMMI等级，从而达到提升管理水平的目的。