# 软件工程管理

## 1. 开源软件开发

* 高质量：可用性不理想：开发者缺乏在可用性，HCI方面的设计经验，常常会模仿商用软件的设计，缺乏自主创新，导致可用性和用户满意度不高；发布的频率过高造成缺乏成熟的设计：大部分开源的项目都没有得到详细并有计划性的设计；参与的人员太多使得设计的质量不高；文档质量不高：开源软件的文档质量并没有商业软件那么好，商业软件有完善的技术文档和用户手册，但很多时候看文档比看源码更有效率；软件越来越复杂而难以控制：参与开发的人员众多，并且开发者们都往往按照自己的需求以及意愿来进行开发；集成和版本控制的成本增加；过多冗余的功能。
* 高开发效率：需求获取及分析：不存在需求不明或与用户难以沟通的问题；设计：不一定要经历从概要设计到详细设计的过程；实现：开发者分散开发，并行工作；可进行快速的bug修复。
* 高强度测试：可以有效地利用开源社区来查找和改进软件的缺陷，测试团队庞大，如果有一个足够大的beta测试人员和协作开发人员的基础，几乎所有的问题都可以被快速的找出并被一些人纠正\*\*（如果有足够多的眼睛，所有的错误都是浅显的）

## 2. 敏捷

* **敏捷的本质：**敏捷的本质，是承认软件开发的复杂性。而且承认，这种复杂性，达到了这样的一种程度：“无法通过足够充分的前期准备，而消除后续的风险。甚至于，前期准备得越是充分，后续的风险越大”；软件开发具有复杂性和可变性；软件开发需求可变性“实际上具有其合理性”。
* 高效率：敏捷开发是轻量型的开发方法，它反对传统、庞大、重型的过程；强调与人交流的重要性，提倡用高质量的软件代替文档，具有能够适应需求变化，进行快速开发的能力。这类方法以快捷、轻便的思维方式，迅速解决了一些传统软件开发中存的问题，提高了软件企业的生产效率，得到了迅速的推广
* 高适应性：产品角度：适用于需求不明确并且快速改变的情况，如系统有比较高的关键性、可靠性、安全性方面的要求，不完全适合；组织结构的角度：组织结构的文化必须支持谈判、人员少而精并且彼此信任、开发人员所作决定得到认可，环境设施满足成员间快速沟通之需要；适用于30人及以下较小队伍、较小规模。
* **敏捷VS计划驱动：**适应性,短计划VS可预期(严格遵循预先计划的需求、分析、设计、编码、测试的步骤顺序进行，变更代价高昂导致难以应对变更)；短周期VS长周期；可运行软件进行度量VS可交付产物或步骤成果进行度量；敏捷仍有严格的规范,与最终产物有关。
* TDD：在开发功能代码之前，先编写测试代码，然后只编写使测试通过的功能代码，迭代该过程，直到所有的测试通过哟，然后重构代码优化。（不可运行-可运行-重构）。测试用例用来表达需求。TDD**实践问题**：测试范围的确定；TDD导致大量的Mock和Stub；Test Case并没有想像中的那么简单：测试用例的完备与否，测试代码本身逻辑的正确与否都依赖于程序员。
* SCRUM：是一个框架，包括Scrum团队及其相关的角色、事件、工件和规则。框架中的每个模块都有一个特定的目的，对Srum的成功和使用都至关重要。Scrum框架包括三个角色（产品负责人、Scrum Master、团队）四个仪式（Sprint计划会议、每日站会、Sprint评审会议、Sprint回顾会议）三个物件（产品BackLog，Sprint BackLog，燃尽图）。具有可操作的过程管理，而xp具有较强的操作性。
* **中国实施敏捷可能会遇到的困难：**1.文化冲击，敏捷导致原来项目管理模式的改变，在企业里推广敏捷很大程度上会触及到企业文化的变革，企业可能无法快速适应2.当遇到困难的时候，他们会选择性的回到其习惯的开发方式，或者根据自己的情况进行裁减，选择性地实施，可能就会变成半敏捷的方式3.学习成本

## 3. 测试

* **软件测试**的经典定义是：在规定的条件下对[程序](https://zh.wikipedia.org/wiki/%E7%A8%8B%E5%BA%8F)进行操作，以发现[程序错误](https://zh.wikipedia.org/wiki/%E7%A8%8B%E5%BA%8F%E9%94%99%E8%AF%AF)，衡量[软件质量](https://zh.wikipedia.org/wiki/%E8%BB%9F%E9%AB%94%E5%93%81%E8%B3%AA)，并对其是否能满足设计要求进行[评估](https://zh.wikipedia.org/w/index.php?title=%E8%AF%84%E4%BC%B0&action=edit&redlink=1)的过程。
* 自动化测试工具：
* 随机测试
* 众测

## 4. 估算

**概念：**根据软件的开发内容、开发工具、开发人员等因素对需求分析、软件设计、编码、测试与整个开发过程所花费的时间、费用及工作量的预测。确定开发时间和开发成本的过程。

估算内容：规模估算、工作量估算、成本估算、进度估算、风险估算、缺陷估算、资源估算等。

明确一点：估算的偏差是必然会存在的，估算不在于精确而在于有用。

**为什么要估算？**估算是一种手段，为了计划、项目管理。

**估算方法及优缺点：代码行**:直观，可以作为衡量工作量的标准; 在项目结束之前不能得到精确的数据。**类比**：估算较为准确；要依赖实际经验和类似历史项目。**专家**：不需要历史数据，适合新项目；主观，专家技术带来误判。**自顶向下**：估算工作量小，速度快；估算不全面，盲目性大。**自底向上**：准确性高；会遗漏系统级工作量，估算值往往偏低。**参数模型**：客观，结果是可重复的；没有采取校准结果往往不准确。**功能点分析**：基于用例，可保持与需求变化的同步；加权调整需要依赖个人经验。

**专家vs模型:**1.专家有天然的优势，可以掌握更多的信息，并且可以更加灵活的运用信息2.构建准确的估算模型很困难，缺少相关性的分析以及学习数据集的数量和质量都会带来严重的影响。

**影响估算准确度的因素**：1.对项目信息的掌握不完全2.太过乐观3.人的主观性4.缺乏优秀的估算工具5.缺乏历史数据的支持

## 4. 软件过程

**软件过程管理基本定律**： 如果开发过程可以在统计控制下，改进过程就可以导致更好的结果。将整个软件开发任务作为一个可控制，可度量，可改进的过程。

统计控制背后的基本原理是度量。度量是昂贵的有破坏性的,度量的越少越好,只度量必须。

**软件过程提高软件能力:**

1知道开发过程所处的状态

2开发一个期望的过程的原型

3建立提高过程质量的举措的列表并按优先级排序

4为实施这些举措定计划

5提供资源来实施计划

如何开发一个CMM框架：1. 明白当前过程状态；2.开发一个期望过程的版本3. 按优先级建立要求的活动4. 制定完成活动的计划5. 提交资源完成计划。

软件过程成熟等级：1初始级2可重复级(提供承诺控制)3定义级(过程定性，没有定量评价过程；文档化过程)4管理级(数据收集过程，昂贵)5优化级( 收集数据，对过程本身进行优化)

初始级：检测是否为初始级：如何面对危机。遵循过程还是单纯的编码和测试。

improve：1.加强项目管理 2.保证遵守过程，质量保证小组 3.变更控制。

可重复级：组织面对新的挑战（新的方法、工具，新的项目）时将存在危机

improve：1.建立过程小组，提高软件开发过程。2.建立软件开发过程的架构。3.引入软件工程的方法和技术。

定义级：过程定性，很少数据说明过程的有效性

improve：1.建立过程可度量的内容，来确定过程的质量和成本。2.建立过程数据库 3.收集维护数据并分析 4.评估软件质量,通知管理层。

管理级：最大的问题是收集数据的成本。过程数据用于描述软件的开发过程，用于过程改进，而不能用于横向比较项目和个人。

improve：1.支持自动化的收集过程数据 2.使用数据分析和改进过程

优化级：优势：1.帮助管理者发现哪里需要帮助，怎样提供支持 2.使专家可以用精准、量化的术语进行交流。 3.提供框架，使专家了解工作表现以及如何改进.

注：CMM五个级别定义：

过程成熟度级别定义：

1. Initial. Until the process is under statistical control, no orderly progress in process improvement is possible.

项目管理；管理监督（review & approval）；QA保证过程质量；变更管理

1. Repeatable. The organization has achieved a stable process with a repeatable level of statistical control by initiating rigorous project management of commitments, cost, schedule, and changes.

建立一个过程小组关注如何提高过程、定义过程；建立一个软件开发过程架构以描述执行开发过程需要的技术管理活动；如果没有则引入一组软件工程方法和技术

1. Defined. The organization has defined the process, to ensure consistent implementation and provide a basis for better understanding of the process. At this point, advanced technology can usefully be introduced.

建立少量的过程度量来标识过程每一步的质量和花费；建立一个过程数据库和管理、维护的资源；提供足够的过程资源以收集和维护数据并使用它给项目成员建议；评估每个产品的相关质量，通知管理成员哪部分没有达到质量目标。

1. Managed. The organization has initiated comprehensive process measurements, beyond those of cost and schedule performance. This is when the most significant quality improvements begins.

支持自动收集过程数据；使用这些数据分析改进过程以防止问题和提高效率

Optimizing. The organization now has a foundation for continued improvement and optimization of the process.第4级关注产品，第5级关注过程自身