生命周期的认识根源和实践根源

麻锦涛 16281262 计科1602

同任何事物一样，一个软件产品或者软件系统也要经历孕育、诞生、成长、成熟、衰亡等阶段，一般称为软件生命周期。软件生命周期是由软件定义，软件开发和运维（也称为软件维护）3个时期组成的，每一个时期又可以划分为若干个阶段。

软件定义时期的任务是：确定软件开发工程必须完成的总体目标，确定工程的可行性，导出实现工程目标应该采用的策略及系统必须完成的功能；估计完成该项工程需要的资源和成本，并且制定工程进度表。这个时期的工作通常又称为系统分析，由系统分析员负责完成。软件定义时期又可以进一步划分为3个阶段，即问题定义，可行性研究和需求分析。

开发时期具体设计和实现在前一个时期定义的软件，它通常由下述4个阶段组成：总体设计，详细设计，编码和单元测试，综合测试。

其中前两个阶段又称为系统设计，后两个阶段又称为系统实现。维护时期的主要任务是使软件持久地满足用户的需求。具体来说，当软件在使用过程中发现错误时应该加以改正；当环境改变时应该修改软件以适应新的环境；当用户有新的要求时，应该及时改进软件以满足用户的新需求。通常对维护时期不再进一步划分阶段，但那时每一次维护活动本质上都是一次压缩和简化了的定义和开发过程。

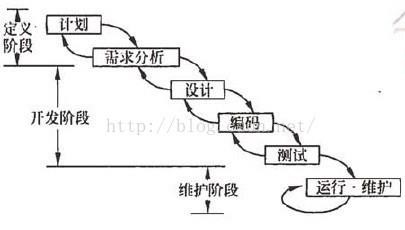
总的来说，如果按照进一步的划分，软件的生命周期可划分为较小的问题定义，可行性研究，需求分析，总体设计，详细设计，编码和单元测试，综合测试，软件维护8个阶段。

### 软件生命周期模型：

软件生命周期模型是指人们为了开发更好的软件而归纳总结的软件生命周期的典型实践参考。为了使规模大、结构复杂和管理复杂的软件开发变的容易控制和管理，人们把整个软件生命周期划分为若干阶段，使得每个阶段有明确的任务，整理出软件生命周期模型。软件生命周期模型主要包括了：瀑布模型、增量模型、原型模型、螺旋模型、喷泉模型等，详细信息如下：

### **(**一**)** 瀑布模型

瀑布模型是一个经典的软件生命周期模型，一般将软件开发分为可行性分析（计划）、需求分析、软件设计（概要设计、详细设计）、编码（含单元测试）、测试、运行维护等几个阶段，如图所示



瀑布模型是一种广为使用的生命周期模型，有时候软件生命周期模型指的就是瀑布模型，它主要有以下特点：

1. 阶段间具有顺序性和依赖性

必须等前一阶段的工作完成之后才能开始后裔阶段的工作；建议阶段的输出文档就是后一阶段的输出文档。只有前一阶段的输出文档正确，后一阶段的工作才能获得正确的结果。

1. 推迟实现的观点

在编码之前设置了系统分析与系统设计的各个阶段，分析与设计阶段的基本任务规定，在这两个阶段主要考虑目标系统的逻辑模型，不涉及软件的物理实现。清楚的区分逻辑与物理设计，尽可能地推迟程序的物理实现。

1. 质量保证的观点

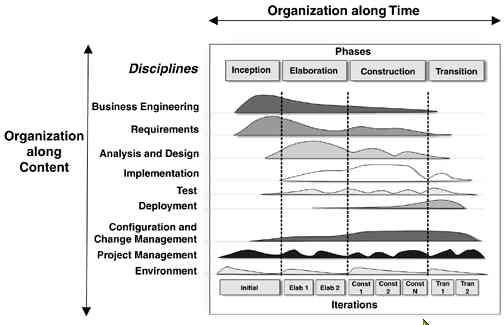
为了保证软件的优质以及高产，每个阶段都必须完成规定的文档，没有交出合格的文档就没有完成该阶段的任务，它有利于开发人员之间的交流，同时也是维护时期的重要依据。

每个阶段结束之前都要对所完成的文档进行评审，以便于尽早地发现问题，改正错误。及时审查时保证软件质量、降低软件成本的重要措施。

由于瀑布模型可以强迫开发人员使用规范的方法（例如，结构化技术）；严格规定了每个阶段必须提交的文档；要求每个阶段交出的所有产品都必须进过质量保证小组的仔细验证。

### **(**二**)** 增量和迭代模型

增量模型也称为渐增模型，使用增量模型进行开发时，把软件产品作为一系列的增量构建来设计，编码、集成和测试。每个构建由多个相互作用的模块构成，并且能够完成特定的功能。增量迭代是RUP 统一过程常采用的软件开发生命周期模型.增量和迭代有区别，但两者又经常一起使用.所以这里要先解释下增量和迭代的概念.假设现在要开发 A,B,C,D四个大的业务功能,每个功能都需要开发两周的时间.则对于增量方法而言可以将四个功能分为两次增量来完成,第一个增量完成A,B功能,第二 次增量完成C,D功能;而对于迭代开发来将则是分两次迭代来开发,第一次迭代完成A,B,C,D四个基本业务功能但不含复杂的业务逻辑,而第二个功能再逐 渐细化补充完整相关



的业务逻辑.在第一个月过去后采用增量开始时候A,B全部开发完成而 C,D 还一点都没有动;而采用迭代开发的时候 A,B,C,D 四个的 基础功能都已经完成.

RUP强调的每次迭代都包含了需求,设计和开发,测试等各个过程, 而且每次迭代完成后都是一 个可以交付的原型.迭代不是并行,在每次迭代过程中仍然要遵循需求->设计->开发的瀑布过程.迭代周期的长度跟项目的周期和规模有很大的关 系.小型项目可以一周一次迭代,而对于大型项目则可以 2-4 周一次迭代.如果项目没有一个很好的架构师,很难规划出每次迭代的内容和要达到的目标,验证相 关的交付和产出.因此迭代模型虽然能够很好的满足与用户的交付,需求的变化,但确是一个很难真正用好的模型. 就对风险的消除上,增量和迭代模型都能够很好的控制前期的风险并解决.但迭代模型在这方面更有优势.迭代模型更多的可以从总体方面去系统的思考问题,从最早就可以给出相对完善的框架或原型, 后期的每次迭代都是针对上次迭代的逐步精化.

业界比较标准的增量模型往往要求在软件需求规格说明书全部出来后后续的设计开发再进行增量.同时每个增量也可以是独立发布的小版本.由于系统的总体设计 往往对一个系统的架构和可扩展性有重大的影响,因此我们推荐的增量最好是在架构设计完成后再开始进行增量,这样可以更好的保证系统的健壮性和可扩展性.

### **(**三**)** 快速原型模型

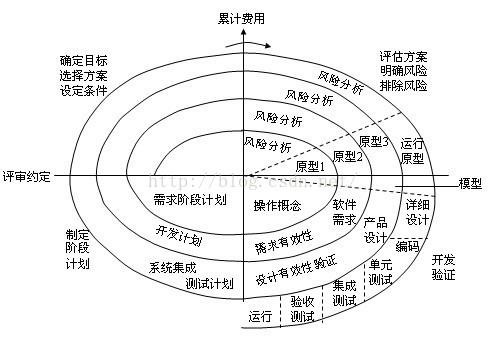
特点：快速建立起能够在计算机上运行的程序（最终产品功能的一个子集）。

优点： 软件产品的开发基本上是线性的

缺点： 必须迅速地构建原型然后根据用户意见循序的修改原型 适用范围： 用户需求不明确，需要通过构建原型来清楚的了解用户的真实需求。

### **(**四**)** 螺旋模型

螺旋模型是一个演化软件过程模型，将原型实现的迭代特征与线性顺序（瀑布）模型中控制的和系统化的方面结合起来。使得软件的增量版本的快速开发成为可能。在螺旋模型中，软件开发是一系列的增量发布。螺旋模型的整个开发过程如图所示



图中的螺旋线代表随着时间推进的工作进展；开发过程具有周期性重复的螺旋线形状。4个象限分别标志每个周期所划分的4 个阶段：制定计划、风险分析、实施工程和客户评估。螺旋模型要点：统一了瀑布模型与原型模型，与增量模型相似，更强调风险分析。

1.软件分多个版本开发，每个版本就是一次螺旋。

2.每个版本按照这样的顺序进行：

1. 确定软件目标，选取定实施方案，弄清项目开发的限制条件；

（图中左上象限）

1. 分析所选取方案，考虑如何识别和消除风险；（图中右上象限）
2. 实施软件开发；（图中右下象限）
3. 评价开发工作，提出修正建议，调整计划。（图中右下象限、左下象限）

3.需求不是一次获取和实现的，通过多个螺旋来完善。

4.计划也不是一次成型的，每次螺旋都需要调整。

该模型在实际工作中实用性还是相当高的，但可能是该模式很多资料都说得不太清楚，让很多人会有一些误解。

几种生命周期模型间的比较

软件生命周期模型规定了把生命周期划分成哪些阶段及各个阶段的执行顺序，因此，也称为过程模型。所以几种生命周期模型都规定了各个阶段应该执行的顺序，但是它们各自规定的生命周期中应该执行的顺序是不一样的。另外，不同的生命周期模型它们还有各自独特的要求，比如：瀑布模型的思想是通过完整的顺序性和前一阶段输出作为后一阶段的输入的思想，并尽可能推迟具体实现来达到软件开发的优质。快速原型分析的第一步便是快速建立一个能够反映用户主要需求的原型系统，让用户在计算机上使用它，通过实践来了解目标系统的概貌。增量模型是在完成了总体设计之后，一个构建一个构建逐个完成，并还强制要求了每一个构建的设计文档的说明，以及风险分析和测试。螺旋模型，实际上是为了降低软件风险，在每一个阶段都

设置了风险分析，然后通过逐个添加构建的方法来完成软件的开发。

可行性研究的地位和作用：

可行性研究阶段的主要作用是对在问题定义阶段所确定的问题是否有行得通的解决方案进行研究和分析。系统分析员需要进行一次大大压缩恶化简化了的系统分析和设计过程，也就是在较为抽象的较高层次进行分析和设计。

可行性研究的任务并不是去具体解决问题，而是研究问题的范围，探索这个问题是否值得去解，是否有可行的解决办法。可行性研究的结果是客户做出是否继续进行这项工程的决定的重要依据，通常只有那些投资可能缺德较大消息的那些工程才值得继续进行下去。可行性研究的意义在于对于那些不值得继续进行研究的项目或者工程应当及时停止，因为在再可行性研究后的阶段需要投入更多的人力和物力，及时终止不值得投资的工程项目，可以避免更大的浪费和损失。