

深圳龙视传媒有限公司	文 档 编 号	LVM_ORG_P_003
	版本号	V1.0
文档名称	生命周期模型及选择指南	

生命周期模型及选择指南

拟制	王慧敏	日期	2012-08-01
审核	何萍	日期	2012-08-15
批准	易睿	日期	2012-08-30

修订记录

日期	修订版本	修改描述	作者	评审成员
2012-08-29	V1.0	新建	王慧敏	EPG

目 录

1	目的.....	1
2	范围.....	1
3	关键词.....	1
4	缩略语.....	1
5	过程概述.....	1
6	生命周期模型描述.....	2
6.1	瀑布模型.....	2
6.1.1	概述.....	2
6.1.2	阶段定义.....	2
6.1.3	适用情况.....	3
6.1.4	优点.....	3
6.1.5	缺点.....	3
6.1.6	本企业适合项目类型.....	4
6.2	迭代模型.....	4
6.2.1	概述.....	4
6.2.2	阶段定义.....	4
6.2.3	适用情况.....	5
6.2.4	优点.....	5
6.2.5	缺点.....	5
6.2.6	本企业适合项目类型.....	5
6.3	原型模型.....	5
6.3.1	概述.....	5
6.3.2	阶段定义.....	6
6.3.3	适用情况.....	6
6.3.4	优点.....	6
6.3.5	缺点.....	6
6.3.6	本企业适合的项目类型.....	7
7	参考资料/引用文档.....	7

1 目的

软件生命周期由制定计划、需求开发、设计、编码、测试、维护等各项活动组成，而如何将这些活动合理、有效地衔接组织起来，就需要在软件项目策划阶段选择合适的软件生命周期模型。正如每个项目的目的是唯一的，每个项目的软件生命周期模型也将是唯一的，定义软件生命周期是项目计划的一个重要步骤，它将直接影响到 WBS 及软件开发计划的制定。本文的目的是为了指导软件项目策划人员如何选用软件生命周期模型。

2 范围

本文档适用于公司中的所有软件项目。

3 关键词

关键词	解释
软件生命周期 (Software life cycle)	从软件产品的设想开始到软件不再使用而结束的时间周期。软件生命周期一般包括需求阶段、设计阶段、实现阶段、测试阶段、运行和维护阶段。
软件过程	有关开发和维护软件及其相关产品（例如：项目计划、设计文档、代码、测试用例、用户手册等）的活动、方法、实践和变更的集合。

4 缩略语

无

5 过程概述

为了使项目在定义软件过程时能够依据其特性选择适用的软件生命周期，使得项目开发过程流程化、易于管理、提高开发速度和产品质量，以达到更好的满足客户的要求，组织规定了以下几种适于本组织使用的生命周期模型：

- 瀑布模型
- 迭代模型

- 原型模型

6 生命周期模型描述

6.1 瀑布模型

6.1.1 概述

瀑布模型，它是一种线型顺序模型，是项目自始至终按照一定顺序的步骤从需求分析进展到系统测试直到提交用户使用，它提供了一种结构化的、自顶向下的软件开发方法，每阶段主要工作成果从一个阶段传递到下一个阶段，必须经过严格的评审或测试，以判定是否可以开始下一阶段工作，各阶段相互独立、不重叠。瀑布模型是所有软件生命周期模型的基础。

瀑布模型的开发流程如下图：

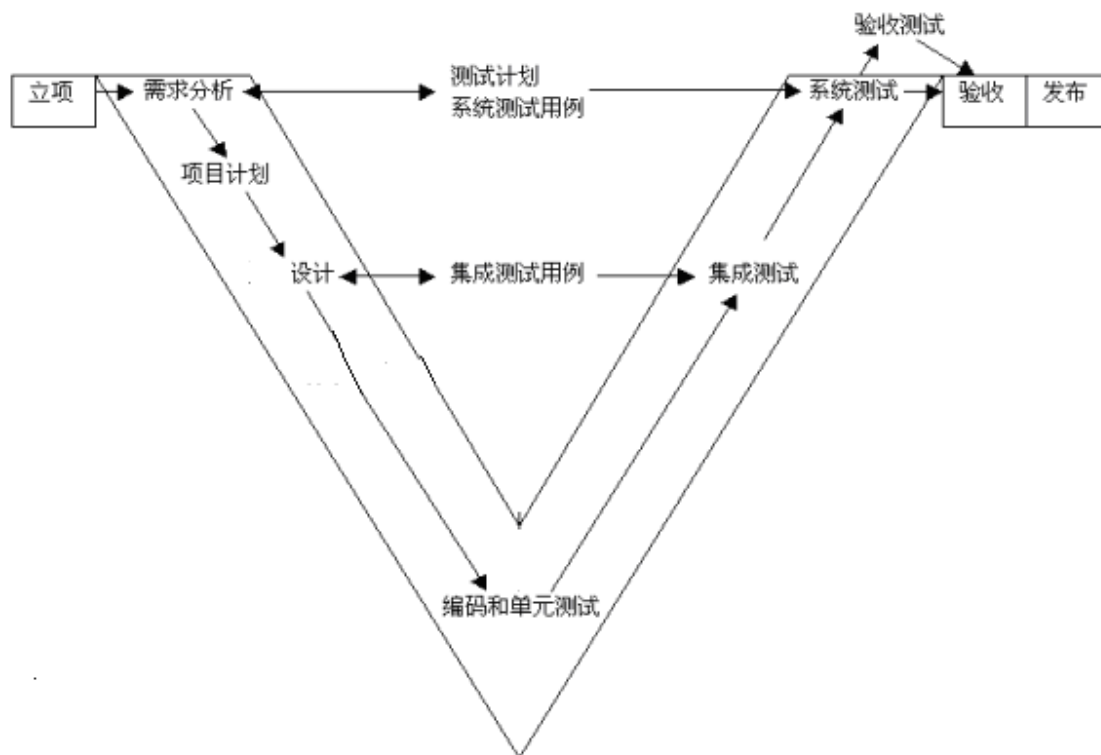


图 1 V 字模型示意图

6.1.2 阶段定义

No	阶段	入口标准	任务	出口标准
1	需求分析	项目立项报告已经由高层经理签字，项目开始启动。	需求访谈及分析 测试方案设计	包需求、系统设计需求，需求分解分配、及测试方案完成并形成基线
2	设计	包需求、系统设计需求，需求分解分配、及测试方案完成并形成基线	进行数据库设计 各模块的软件设计说明书 集成测试设计	软件设计说明书及集成测试方案与计划完成并形成基线。
3	实现	软件设计说明书及集成测试方案与计划完成并形成基线。	进行编码及单元测试	编码及、单元测试、用户文档编写完成并形成基线。
4	测试	系统测试设计完成 集成测试设计完成 编码及单元测试完成 用户文档完成（安装、操作、维护）	进行集成、系统测试	集成、系统测试完成并形成基线
5	运行维护	测试已经完成	系统安装、运行、维护	组织不再对产品进行维护

6.1.3 适用情况

- 充分理解用户需求，且需求是确定不变的；
- 用户有一定的能力，对需求的表述是确切的；
- 充分理解该解决方案的技术和体系；
- 需要一个可维护性和可支持性较高的解决方案；
- 所有过程工作产品的控制基线，需要有可见度和可靠性；
- 适用于新的有较多用户的产品、平台/中间件开发项目，或者是用户对开发过程有严格要求的工程定制项目；
- 项目经理有一定的项目管理经验；
- 要求开发周期时间较充分。

6.1.4 优点

- 强调开发的阶段性；
- 强调早期的计划及需求调查与分析；
- 强调产品测试的完备性；
- 过程文档齐全，便于追溯和重用；
- 过程的可见性强，便于过程质量控制；
- 只要需求是稳定的，则进度也是稳定的。

6.1.5 缺点

- 无法解决软件需求不明确或不准确的问题；
- 灵活性差，依赖于早期进行的需求调查，不能适应需求的变化；
- 由于是单一流程，开发中的经验教训不能及时反馈并应用于本产品的过程改进。

6.1.6 本企业适合项目类型

组织所熟悉领域的应用系统开发；

6.2 迭代模型

6.2.1 概述

在项目做计划的过程中，选用迭代模型时，有如下要求：

- 进行第一次项目计划时，确定所选择的生命周期模型为迭代模型时，要求在计划中明确进行迭代流程阶段、迭代的次数、每次迭代所选的生命周期模型以及每次迭代的起止日期。
- 每次迭代选择适当的生命周期模型。
- 对项目 WBS 的要求：

以下表格可以与 WBS 结合，用于明确各流程阶段的工作任务、该任务在本次迭代中的重要程度（强、中、弱）、该流程阶段的控制点及控制手段（如重要程度为“强”的任务须进行评审，“中”的任务可以通过变更过程进行控制，“弱”的任务可以通过批准直接在文档的修订页中注明）。

迭代次数	流程阶段	工作任务	重要程度 (强、中、弱)	工作产品	控制点及控制手段

- 根据每次迭代的 WBS 任务和各 WBS 任务在本次迭代中的重要程度（强、中、弱），参照迭代模型样例图，绘制本项目的迭代模型图。
- 从第二次到第 N 次的迭代，在不与第一次计划冲突的基础上，制定本次迭代的小计划，也可以直接在项目的 PROJECT 图上进行本次迭代计划的细化。
- 如果后几次迭代对第一次计划的内容有变动，如进度的调整，控制点的变化等，则须进行变更及批准。

6.2.2 阶段定义

参见瀑布模型。

6.2.3 适用情况

- 规模较大的项目或产品
- 需求的清晰度低，且需要进一步的调查
- 技术或体系结构方面的知识匮乏

6.2.4 优点

- 允许变更需求，中途的修改是容易的，如果在项目组内部和外部之间有良好的沟通渠道
- 有助于项目组的学习和提高，团队成员有机会在整个生命周期中边做边学，各显其能
- 迭代流程自身可在进行过程中得到改进和精炼
- 生成性能更强壮的产品
- 风险管理比较容易，可及早降低风险，前提是存在良好的信息传递渠道
- 与其他生命周期模型相比，它在开发周期内具有更好的性能

6.2.5 缺点

- 因本模型较为灵活，对管理的要求较高，项目经理需要有丰富的项目管理经验
- 迭代的次数和任务规划难把握，对项目策划要求较高

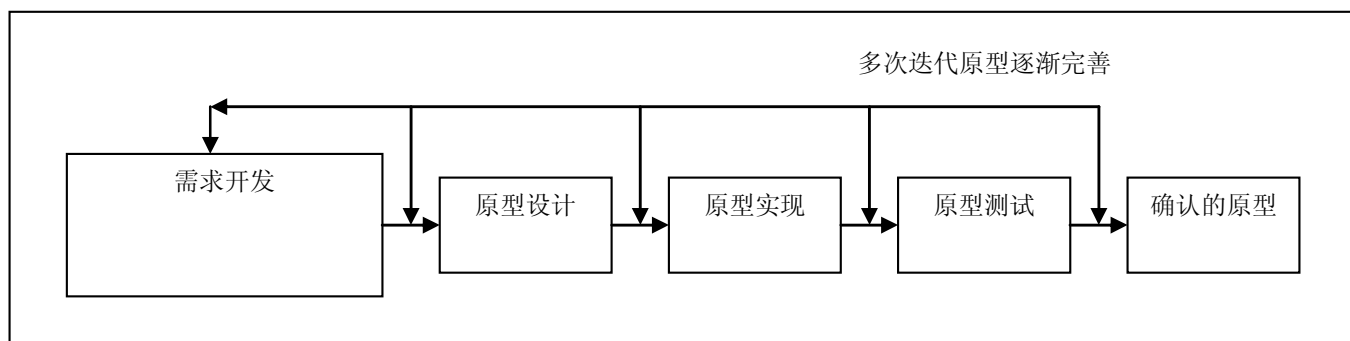
6.2.6 本企业适合项目类型

新领域、新技术的研发项目。

6.3 原型模型

6.3.1 概述

原型模型本身是一个迭代的模型，是为了解决在产品开发的早期阶段存在的不确定性、二义性和不完整性等问题，通过建立原型使开发者进一步确定其应开发的产品，使开发者的想象更具体化，也更易于被客户所理解。原型只是真实系统的一部分或一个模型，完全可能不完成任何有用的事情，通常包括抛弃型和进化型两种，抛弃型指原型建立、分析之后要扔掉，整个系统重新分析和设计；进化型则是对需求的定义较清楚的情形，原型建立之后要保留，作为系逐渐增加的基础，采用进化型一定要重视软件设计的系统性和完整性，并且在质量要求方面没有捷径，因此，对于描述相同的功能，建立进化型原型比建立抛弃型原型所花的时间要多。



图表 2 原型+瀑布模型开发流程图

6.3.2 阶段定义

No	阶段	入口标准	任务	出口标准
1	需求开发	项目立项报告已经由高层经理签字，项目开始启动。	需求访谈及分析	包需求、系统设计需求，需求分解分配、及测试方案完成
2	原型设计	软件需求说明	快速设计系统原型	软件设计说明书已完成
3	原型实现	原型设计说明	快速进行原型的制作	系统原型
4	原型测试	系统原型	用户测试评估原型，进一步精化需求，开发人员制作新的软件需求	改进后的软件需求，并作为下个原型的设计入口，循环至原型得到确认

6.3.3 适用情况

- 项目包含一种新技术，例：新硬件、新的开发语言、新的系统架构等；
- 需求不很清楚；
- 存在关于性能、可靠性和可行性方面的主要的、未解决的问题；
- 用户界面对系统成功是很关键的，但不很清楚。

6.3.4 优点

- 开发者可以很快的构建系统，客户也可以尽快的感受到实际的系统。
- 客户和开发者对系统有更好的理解。
- 客户充分参与，可以减少部分培训的工作。

6.3.5 缺点

- 由于原型并非最终产品，如果原型不能利用，可能导致成本的增加；同时会引起客户的误解，以为产品即将完成。
- 开发者为了使原型能够尽早工作，常常会采取一些临时性的做法，而不管这些做法是否合理，如：使用一个效率低的算法仅仅是为了演示功能。在经过一段时间后，开发者对这些做法已经习惯了，忘记了它们不合理的原因。于是这些不合理的做法就成了系统的组成部分。

6.3.6 本企业适合的项目类型

新领域的应用项目的开发

7 参考资料/引用文档

- 《软件工程实践者的研究方法》，Roger S. Pressman 著，梅宏等译，机械工业出版社，2002 年 9 月
- 《实用软件工程》郑人杰、殷人昆、陶永雷著，清华大学出版社，1997 年 4 月
- 《软件需求》，Karl E. Wiegers 著，陆丽娜、王忠民、王志敏等译，机械工业出版社，2000 年 7 月
- 《统一软件开发过程》，Ivar Jacobson、Grady Booch、James Rumbaugh 著，周伯生、冯学民、樊东平等译，机械工业出版社，2002 年 1 月