Review-软件过程模型

by:2022211928 gwc

软件过程

软件生命周期 SDLC

Software Development Life Cycle

软件过程定义了： 1.人员与分工　2.所执行的活动　3.活动的细节和步骤

软件过程**通过**： 1.定义软件生产过程中的活动 2.定义这些活动的顺序及其关系 **来组织和管理软件生命周期**

软件过程的**目的**是：1. 标准化(可模仿)　2.可预见性(降低风险) 3.提高开发效率 4.获得高质量产品 5.提升制定时间和预算计划的能力

> 软件开发需要**过程**吗？能不能写了再**改**?

黑盒过程--存在的问题：1.要求开发之前需求被充分理解 2.与客户的交互只在开始(需求)和最后(发布)——类似于产品制造过程

白盒过程--优点：1.通过改进可见性来减少风险 2.在开发过程中，通过不断地获得顾客的回馈允许变更——类似于服 务过程

软件过程的典型阶段

⬇️Dream(提出设想)

⬇️Investigation(深入调研)

⬇️Software Specification(需求规格说明)

⬇️Software Design(软件设计)

⬇️Software Implementation(软件实现)

⬇️Software Deployment (软件部署)

⬇️Software Validation(软件验证)

🔄Software Evolution(软件演化)

软件过程：在软件开发生命周期内采取特定 的方式和策略进行过程管理控制

↓

软件过程模型：是一种**开发策略**，这种策略针对软件工程的各个阶段提供了一套范形，**使工程的进展达到预期的目的**

软件过程模型

软件过程模型分类:

- **预测**型(Predictive): **提前**进行**大量的计划**工作，然后**一次性执行。**执行是一个连续的过程。

  瀑布模型 (Waterfall)、V模型(V Model)、W模型（W Model）形式化过程 (Formal model)

- **迭代**型(Iterative): 允许对未完成的工作**进行反馈**，从而**改进和修改**该工作。

  螺旋模型 (Spiral)、原型模型 (Prototype)

- **增量**型(Incremental): **分次**向客户提供已完成的，可以立即使用的**可交付成果 。**

  增量模型 (Incremental)、快速应用开发(RAD)

- **敏捷**型(Agile): 既有**迭代**，也有**增量** ，便于完善工作，**频繁交付 。**

  XP、Scrum、Kanban、DevOps等

(对比图)

Characteristics				
Approach	**Requirements**	**Activities**	**Delivery**	**Goal**
Predictive	Fixed	Performed once for the entire project	Single delivery	Manage cost
Iterative	Dynamic	Repeated until correct	Single delivery	Correctness of solution
Incremental	Dynamic	Performed once for a given increment	Frequent smaller deliveries	Speed
Agile	Dynamic	Repeated until correct	Frequent small deliveries	Customer value via frequent deliveries and feedback

模型举例&各自特点（不包含敏捷）

【预测型】瀑布模型

- 简介:

  将SDLC各项活动规定为按固定顺序而连接的若干阶段工作，形如瀑布流水，最终得到软件产品

- 特点:

  1.上一个阶段结束，**下一个阶段才能开始**　2.每个阶段均有里程碑和提交物

  3.上一阶段输出是下一阶段输入　　4.每个阶段**均需要进行V&V**　　5.**侧重于文档与产出物**

  （又称“鲑鱼模型”，难以向前一阶段回溯）

- 优点：**追求效率**
- 缺点：**过于理想化**
- 适用场合:

  1.软件**项目较小**，各模块间接口定义非常清晰　2.需求在项目开始之前已经被**全面的了解**，产品的**定义非常稳定**

  3.**需求**在开发中**不太可能发生重大改变**　　4.使用的**技术非常成熟**，**团队成员都很熟悉**这些技术

  5.负责各个步骤的子团队分属不同的机构或不同的地理位置，**不可能做到频繁的交流**　6.外部环境的不可控因素很

【预测型】V模型 & W模型

- 简介:

  V模型是瀑布模型的一种变形

  相对于V模型，W模型增加了软件开发各阶段中**同步进行的验证和确认**活动

- 特点:

  V模型、W模型均**强调测试的重要性**，将**开发活动**与**测试活动**紧密联系在一起

W模型由两个V字型模型组成，分别代表测试与开发过程，明确表示出了测试与开发的并行关系

- 优点：**开发过程重视测试/验证**

  1.简单易用，只要按照规定的步骤执行即可　　2.**强调测试或验证**与开发过程的对应性和并行性

  3.测试方案在编码之前就已经制定了　　4.与瀑布模型相比，项目开发成功的机会更高　　5.避免缺陷向下游流动

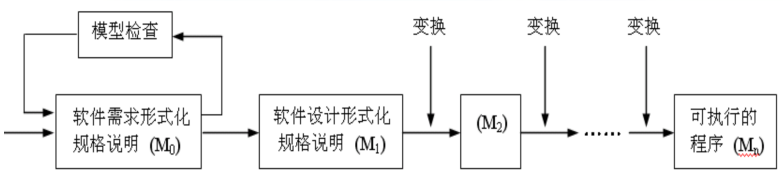
- 缺点：比瀑布模型繁琐

【预测型】形式化过程模型

- 简介&特点：

  使用严格的数学形式来刻画每一阶段的产物(需求、设计、程序、测试)

  应用一系列形式化方法在各阶段的产物之间进行自动/半自动的转换



——**太过于理想化**，因此仅停留在理论研究中，**实践中很少使用**

- 优点：

  应用数学分析方法，歧义性、不完整性、不一致性等问题更容易被发现和改正，**目的**是“**提供无缺陷的软件**”

- 缺陷：

  形式化数学方法**难以理解，可视性太差，对开发人员技能要求较高**

  构造形式化模型是一件**非常耗时**的工作，**成本也很高**

  软件系统中的某些方面**难以用形式化模型表达出来**（如用户界面）

- 应用场合：

  对可靠性和安全性要求较高的一些关键系统，在真正被投入使用之前，需要严格保证100%的正确；传统的方法靠人去验证，难以奏效

【迭代型】螺旋模型

- 简介：

  （图片见文末）

  螺旋模型**沿着螺线旋转**，在**四个象限**内表达四个方面的活动：

  1.**制定计划**：确定软件目标，选定实施方案，弄清项目开发的限制　　2.**风险分析**：分析所选方案，考虑如何识别和消除风险

  3.**实施工程**：实施软件开发　　4.**客户评估**：评价开发工作，提出修正建议

- 特点：

  开发过程中及时**识别和分析风险**，并采取适当措施以**消除或减少风险带来的危害**

  与增量、RAD等的最大区别在于**重视风险评估**

- 优点：结合了原型的迭代性质与瀑布模型的系统性和可控性，是一种 风险驱动型的过程模型

  1.采用循环的方式**逐步加深系统定义和实现的深度**，同时**更好的理解、 应对和降低风险**

  2.确定一系列里程碑，**确保各方都得到可行的系统解决方案**

3.始终**保持可操作性，直到软件生命周期的结束**

4.由**风险驱动**，支持现有软件的**复用**

- 缺点：

1.适用于**大规模软件项目**，特别是内部项目，**周期长、成本高**

2.软件**开发人员应该擅长寻找可能的风险，准确的分析风险**，否则将会带来更大的风险

【迭代型】快速原型法

- 简介：

“原型”的类型：

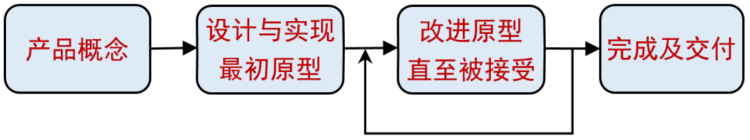
- Throwaway prototyping(**抛弃式原型**)

最初的原型在完成并得到用户认可之后，将**不会作为交付给用户的最终系统的一部分**，而是被抛弃，其目的只是为了收集与验证需求 。该类原型**可能是不可执行的**(例如，只包含用户界面)

- Evolutionary prototyping(**演化式原型**)

最初构造的原型将具备较高的质量，**包含了系统的核心功能**，然后通过收集需求对其进行不断的**改善和精化** 。该类**原型是可执行的**，将成为最终系统的一部分

- 特点：



- 优点：

提高和改善客户/用户的参与程度，最大程度的**响应用户需求的变化**

克服预测型模型的缺点，**减少由于需求不够明确带来的开发风险**

- 缺点：

为了尽快完成原型，开发者**没有考虑**整体软件的**质量**和长期的**可维护性**，**系统结构通常较差**

可能混淆原型系统与最终系统，原型系统在完全满足用户需求之后可能会被直接交付给客户使用

额外的开发费用

PS：迭代过程模型总结

迭代过程模型的目的：需求的变更频繁，要求在非常短的期限内实现，以充分满足客户/用户要求、及时投入市场

存在的问题： 1.由于构建产品所需的周期数不确定，给项目管理带来困难 2.迭代速度太快，项目陷入混乱；迭代速度太慢，影响生产率 3.为追求软件的高质量而牺牲了开发速度、灵活性和可扩展性

【增量型】增量模型

- 简介：

软件被**作为一系列的增量**来设计、实现、集成和测试，**每一个增量**是由多个相互作用的模块所形成的**特定功能模块**或**功能模块组。**

- 特点：

（本质）以**迭代的方式运用瀑布模型**

- 优点：

1.在**时间要求较高**的情况下交付产品（对客户起到“镇定剂”的作用）

2.**人员分配灵活。**如果**找不到足够的开发人员，可采用增量**模型（早期的增量由少量人员实现，如果客户反馈较好，则在下一个增量中 投入更多的人力）

3.逐步增加产品功能可以**使用户有较充裕的时间来学习和适应**新产品

4.具有**较高优先权的模块被首先交付**，使得**最重要的功能**肯定**接受了最多的测试**，使得**项目总体性失败的风险比较低**

- 缺点/困难：

1.每个附加的增量并入现有软件时，必须**不破坏原来已构造好的部分**

2.同时，加入新增量时**应简单、方便** ——该类软件的**体系结构应当是开放**的

3.仍然**无法处理需求发生变更**的情况

4.管理人员必须有足够的技术能力来**协调**好**各增量**之间的关系

【增量型】RAD模型

- 简介：

快速应用开发RAD（**R**apid **A**pplication **D**evelopment)

- 特点：

多个团队并行进行开发，但**启动时间有先后**，先启动团队的**提交物将作为**后启动团队的**输入**

- 优点：

侧重于**短开发周期**（一般为60~90天）的**增量过程模型**，是**瀑布模型的高速变体**，通过**基于构件的构建方法**实现**快速开发**

- 缺点：

需要**大量的人力资源**来创建多个相对独立的RAD团队

如果没有在短时间内为急速完成整个系统**做好准备**，RAD项目将会失败

如果系统不能被**合理的模块化**，RAD将会带来很多问题

技术风险很高的情况下（采用很多新技术、软件需与其他已有软件建立集成等等），**不宜**采用RAD

各类模型一句话归纳

- 瀑布模型：【基本模型】将全部需求以整体方式向前推进，无迭代
- 原型模型：【基本模型】始终结果可见，不断迭代修正原型，直到开发完成
- 增量模型：【串行的瀑布】将需求分成多份，串行推进，无迭代
- RAD模型：【并行的瀑布】将需求分成多份，可并行推进，无迭代
- 螺旋模型：【原型+瀑布】按瀑布阶段划分，各阶段分别迭代(原型+风险分析)
- 敏捷模型：【增量+迭代+原型】将需求分成尽量小的碎片，以碎片为单位进行高速迭代

总结

适用方式 客观因素	敏捷 (Agile)	计划驱动 (Plan-driven)	形式化的开发方法 (Formal Method)
产品可靠性要求	不高，容忍经常出错	必须有较高可靠性	有极高的可靠性和质量要求
需求变化	经常变化	不经常变化	固定的需求，需求可以建模
团队人员数量	不多	较多	不多
人员经验	有资深程序员带队	以中层技术人员为主	资深专家
公司文化	鼓励变化，行业充满变数	崇尚秩序，按时交付	精益求精
实际例子	写一个微博网站	开发下一版本办公软件；给商业用户开发软件	开发底层正则表达式解析模块；科学计算；复杂系统核心组件
用错方式的后果	用敏捷的方法开发登月火箭控制程序，前N批宇航员都挂了	用敏捷方法，商业用户未必受得了2-4周一次更新的频率	敏捷方法的大部分招数都和这类用户无关，用户关心的是：把可靠性提高到99.99%，不要让微小的错误把系统搞崩溃！

（图：螺旋模型示意图）