第二章节 概念篇

本节课课程目标

> 需求的概念
>
> 开发模型
>
> 测试模型

1. 什么是需求

在多数软件公司，会有两部分需求，一部分是用户需求，一部分是软件需求。

1.1 用户需求

用户需求：可以简单理解为甲方提出的需求，如果没有甲方，那么就是终端用户使用产品时必须要完成的任务。该需求一般比较简略，通常是一句话。

> 用户的需求是五花八门，往往只是一句话
>
> 比如：实现一个声控灯，实现一个软件的登录功能

1.2 软件需求

或者叫功能需求，该需求会详细描述开发人员必须实现的软件功能。软件需求是测试人员进行测试工作的基本依据。

用户需求和软件需求有什么不同呢？看看下面的案例

> 女朋友饿了的例子
>
> 用户需求:
>
> 女朋友说, 我饿了, 这是一个用户需求. 很简略.
>
> 软件需求:
>
> 需要你和她反复的沟通了解更加详细具体的需求, 来制定解决方案.
>
> 比如你问她, "想吃啥?", 她说, "随便"
>
> "吃米饭炒菜?", "不想吃"; "那你想吃啥?", "随便"
>
> "吃油泼面?", "不想吃"; "那你想吃啥?", "随便"
>
> ...
>
> 最终理解清楚用户需求之后, 知道女朋友想吃的是你做的红烧肉, 那么再去研究肉怎么买, 怎么做等等的具体步骤, 是软件需求.

在工作中我们实际见到的软件需求文档类似于下面的表述:

> 软件需求规格说明书 一、用户需求:
>
> 平台支持邮箱注册
>
> 二、软件需求:

比特就业课

1.1.1.1　　注册账号

1.1.1.1.1　　功能概述

用户可以通过填写邮箱信息在平台注册个人用户。

1.1.1.1.2　　用户角色

匿名用户。

1.1.1.1.3　　前置条件

无。

1.1.1.1.4　　输入

序号	栏位名称	栏位说明	长度	类型	备注
1	姓名	必填，录入个人姓名	6~15 位	字符型	
2	电子邮箱	必填，录入电子邮箱		字符型	
3	密码	必填，输入的密码隐藏*号显示	6~15 位	字符型	
4	确认密码	必填，输入的密码隐藏*号显示	6~15 位	字符型	
5	验证码	必填，录入验证码		字符型	
6	注册	注册操作		操作型	

1.1.1.1.5　　处理

1.1.1.1.5.1　　基本事件流

1、 用户选择注册；

2、 系统展现用户协议界面，并请用户确认是否同意用户协议

若用户不同意协议，系统禁止用户注册。

若用户同意协议，用户进行注册信息填写。

3、 用户填写注册信息。

注册个人，填写：姓名，电子邮箱，密码，确认密码，验证码。

4、 用户提交注册信息；

5、 系统提示用户并向用户注册的电子邮件地址发送一封含有激活信息的电子邮件。系统并提示用户，若未收到激活邮件，可使用注册的邮箱和密码登录系统后再次发送激活邮件。

6、 用户可执行激活操作，直接跳转至注册邮箱门户页面。

7、 用户通过接收到的电子邮件中的激活信息激活账号，用户注册完成，流程结束。

1.1.1.1.5.2　　扩展事件流

用户注册并激活成功后，第一次登录平台时，提示用户完善信息；

1.1.1.1.5.3　　异常事件流

若用户未收到激活邮件，可在登录界面录入电子邮件及密码后，再次发送激活邮件。

每次发送的激活邮件，仅在发送邮件后起 24 小时之内有效，超过 24 小时后需重新发送激活邮件。

1.1.1.1.6　　输出

用户注册成功

1.1.1.1.7　　后置条件

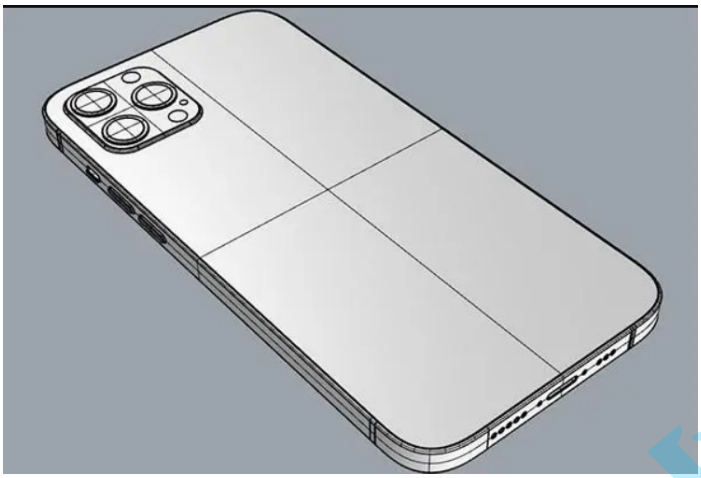
该模块为用户登陆等的前置模块。

注意：用户的需求不能直接作为开发和测试的依据。针对用户的需求，产品经理需要进行需求分析（技术可行性、市场可行性、成本投入和收益占比等）后才可转变为软件需求。

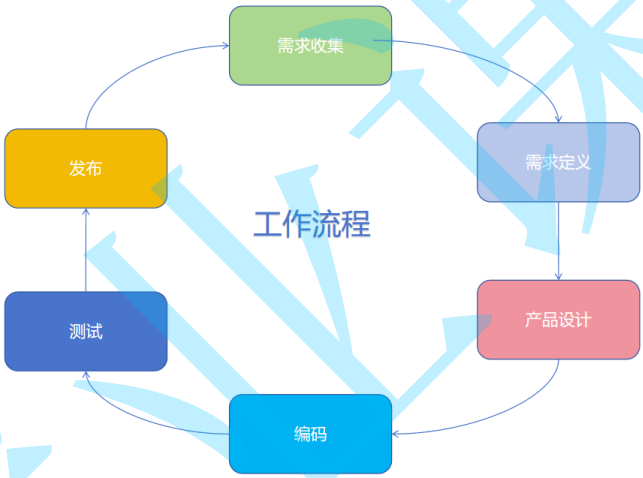
2. 开发模型

2.1 什么是“模型”

你以为的模型



实际的模型：



随着软件工程学科的发展，人们对计算机软件的认识逐渐深入。软件工作的范围不仅仅局限在程序编写，而是扩展到了整个软件生命周期，如软件基本概念的形成、需求分析、设计、实现、测试、安装部署、运行维护，直到软件被更新和替换新的版本。软件工程还包括很多技术性的管理工作，例如过程管理、产品管理、资源管理和质量管理，在这些方面也逐步地建立起了标准或规范。

2.2 软件的生命周期

> 认识具体的开发模型之前先了解软件的生命周期。

什么是生命周期?

> 生命周期指的是从生命的开始到生命结束的一段时间。以人为例，人类的生命周期是从生命孕育的开始，中间会经历幼年，童年，少年，青年，老年，最终直至死亡。

> 而软件/产品的生命周期也是如此，需求的开始是软件生命的起点，中间会经历需求的计划、设计，程序开发，程序测试等阶段，直至软件不再进行维护便到了生命的重点。

案例：

> 假如我想要建造一套房子（别问，问就是一个人造房子），房子的生命周期（流程）是什么样的？

步骤	总结	映射软件流程
为什么要建房子？商品房还是普通住宅？建造100层技术上是否可行？	明确合理的建房目标	需求分析
什么时候开发建房子？计划竣工时间？多久可以交房？	计划好时间	计划
建房前明确流程：先打地基，做基础框架，砌墙、粉刷、水电工程.....	设计好具体的建房流程	设计
按照前面的流程和时间实施建房中....	施工中	编码
房屋建造完成，开发商验收成果、买家验收房子品质（房子是否牢固，是否漏水及其他偷工减料的地方，是否按照规定来建造的）	检查房屋建造结果	测试
检查结束开始逐步入住，使用中出现了各种情况如房屋漏水、墙面掉皮、下水道堵塞等问题，一边使用一边找物业修理	使用并及时维护	运行维护

因此，我们就得到了软件（开发）的生命周期：

需求分析——计划——设计——编码——测试——运行维护

对于软件的生命周期中，每个阶段都在做什么呢？

阶段	具体内容	产出
需求分析	分析用户需求是否合理，分别从市场需求、技术等方面进行分析。	该阶段会输出需求等文档。
计划	对成立的需求执行需求执行计划，多长时间内完成该需求，每段时间具体完成哪些功能。	该阶段会输出计划等文档。
设计	将需求细化成一个个任务，团队成员各司其职领取任务并进行技术设计（如何进行架构设计，设计哪些接口、采用什么技术）	该阶段会输出技术等文档。
编码	开发人员参考需求文档、设计文档、交互图等等文件进行代码的编写。	代码文件等文档。

测试	测试人员需要介入到软件的测试中来，参考测试用例对软件进行测试。	测试用例、测试设计与计划、测试报告等文档
运行维护	项目测试结束之后，项目需要进行上线，并对产品进行线上的维护。线上的维护主要分为三个方面。分别为修复性维护、完善性维护和预防性维护。 > 修复性维护：对项目中未发现的问题进行修复。 > 完善性维护：对功能进行完善。 > 预防性维护：居安思危，为了避免产品在线上出现一些其他不可预料的问题，进行一些防护的手段。	

2.3 常见开发模型

2.3.1 瀑布模型

瀑布模型在软件工程中占有重要地位，是所有其他模型的基础框架。瀑布模型的每一个阶段都只执行一次，因此是线性顺序进行的软件开发模式。

瀑布模型的一个最大缺陷在于，可以运行的产品很迟才能被看到。这会给项目带来很大的风险，尤其是集成的风险。因为如果在需求引入的一个缺陷要到测试阶段甚至更后的阶段才发现，通常会导致前面阶段的工作大面积返工，业界流行的说法是：“集成之日就是爆炸之日”。尽管瀑布模型存在很大的缺陷，例如，在前期阶段未发现的错误会传递并扩散到后面的阶段，而在后面阶段发现这些错误时，可能已经很难回头再修正，从而导致项目的失败。但是目前很多软件企业还是沿用了瀑布模型的线性思想，在这个基础上做出自己的修改。例如细化了各个阶段，在某些重点关注的阶段之间掺入迭代的思想。在瀑布模型中，测试阶段处于软件实现后，这意味着必须在代码完成后有足够的时间预留给测试活动，否则将导致测试不充分，从而把缺陷直接遗留给用户

瀑布模型优缺点总结：

优点/特点	缺点

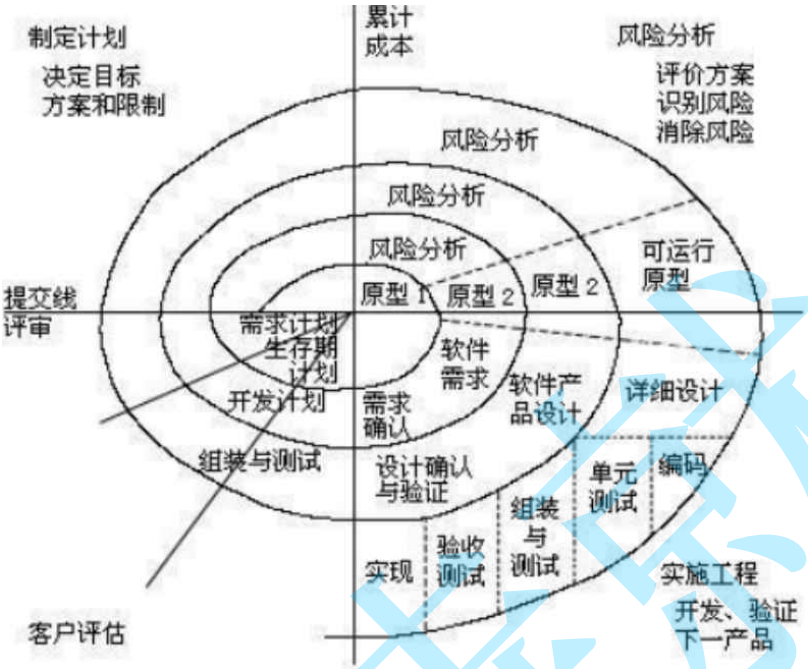
• 强调开发的阶段性；
• 线性结构，每个阶段只执行一次
• 是其他模型的基础框架
	• 测试后置
	  ◦ 前面各阶段遗留的风险推迟到测试阶段才被发现，导致项目大面积返工，失去了及早修复的机会
	  ◦ 必须留有足够的时间给测试活动，否则导致测试不充分，将缺陷直接暴露给用户（产品质量差）
	• 周期太长，产品很迟才能被看到和使用，可能会导致需求/功能过时

> 瀑布模型存在很严重的项目风险，那瀑布模型就不能够被采用了吗？
>
> **瀑布模型的适用场景：需求固定的小项目**
>
> 然而企业中存在许多些规模庞大、复杂度高、风险大的项目，这种情况下可以哪种模型呢？

2.3.2 螺旋模型

一般在软件开发初期阶段需求不是很明确时，采用渐进式的开发模式。螺旋模型是渐进式开发模型的代表之一。

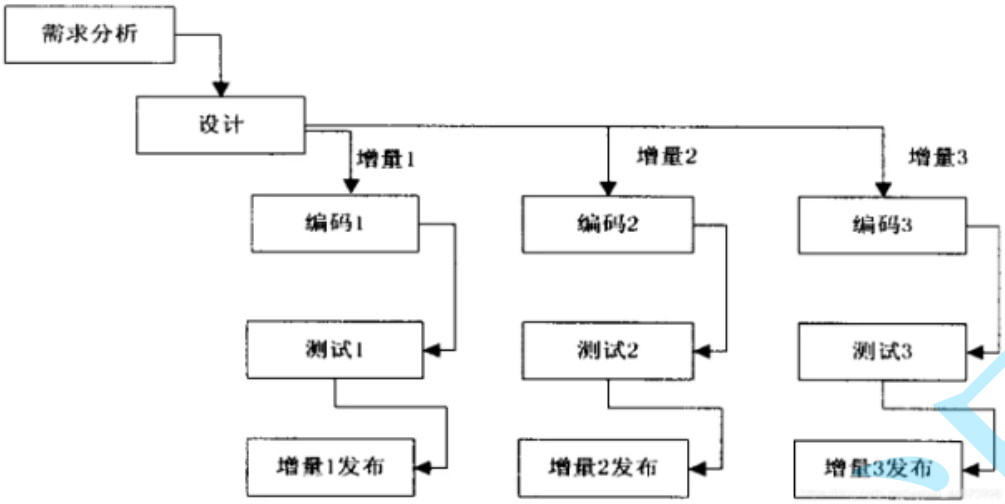
这对于那些规模庞大、复杂度高、风险大的项目尤其适合。这种迭代开发的模式给软件测试带来了新的要求，它不允许有一段独立的测试时间和阶段，测试必须跟随开发的迭代而迭代。因此，回归测试的重要性就不言而喻了。



优点	缺点
• 强调严格的全过程风险管理。 • 强调各开发阶段的质量。 • 增加风险分析和原型	• 项目中可能存在的风险性与风险管理人员的技能水平有直接关系 • 需求人员、资金、时间的增加和投入，可能会导致项目的成本太高

适用场景：规模庞大、复杂度高、风险大的项目。

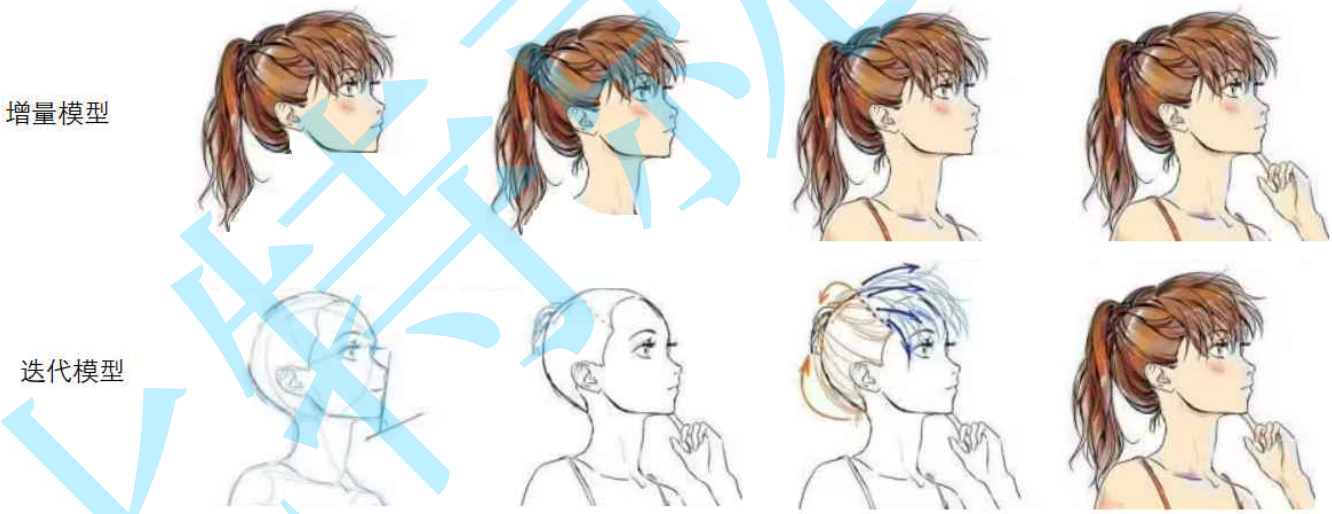
2.3.3 增量模型、迭代模型



增量开发

增量开发能显著降低项目风险，结合软件持续构建机制，构成了当今流行的软件工程最佳实践之一。增量开发模型，鼓励用户反馈，在每个迭代过程中，促使开发小组以一种循环的、可预测的方式驱动产品 的开发。因此，在这种开发模式下，每一次的迭代都意味着可能有需求的更改、构建出新的可执行软件 版本，意味着测试需要频繁进行，测试人员需要与开发人员更加紧密地协作。

与此类似的有一个迭代开发，增量开发和迭代开发往往容易被人，但是其实两者是有区别的。增量是逐块建造的概念，迭代是反复求精的概念。

> 例如画一幅人物画
>
> 
>
>
> 增量模型是先画人的头部，再画身体，再画手脚……
>
> 迭代模型是先画整体轮廓，再勾勒出基本雏形，再细化、着色...

适用场景：大型项目，需求不明确

2.3.4 敏捷模型

在早期，迭代瀑布模型非常流行来完成一个项目。但是现在开发人员在使用它开发软件时面临着各种各样的问题。主要困难包括在项目开发期间处理来自客户的变更请求以及合并这些变更所需的高成本和时间。为了克服瀑布模型的这些缺点，在1990年代中期提出了敏捷软件开发模型。

敏捷模型主要旨在帮助项目快速适应变更请求。因此，敏捷模型的主要目的是促进项目的快速完成。要完成这项任务，需要敏捷。敏捷性是通过使过程适应项目，删除对特定项目可能不是必需的活动来实现的。此外，避免任何浪费时间和精力的事情。

在敏捷模型中，需求被分解成许多可以增量开发的小部分。敏捷模型采用迭代开发。每个增量部分都是在迭代中开发的。每次迭代都旨在小而易于管理，并且只能在几周内完成。一次为客户计划、开发和部署一个迭代。没有制定长期计划。

敏捷模型中有一个非常重要的《敏捷宣言》，宣言内容：

> 个体与交互重于过程和工具
>
> 可用的软件重于完备的文档
>
> 客户协作重于合同谈判
>
> 响应变化重于遵循计划

宣言中主要运用了对比的手法，然而，在每对比对中，后者并非全无价值，但我们更看重前者。

通过敏捷宣言可以总结出敏捷模型的四个特点：**轻文档，轻流程，重目标，重产出**。敏捷开发有很多种方式，其中scrum是比较流行的一种。

Scrum是敏捷模型中的一种，又称为迭代式增量软件开发模型。

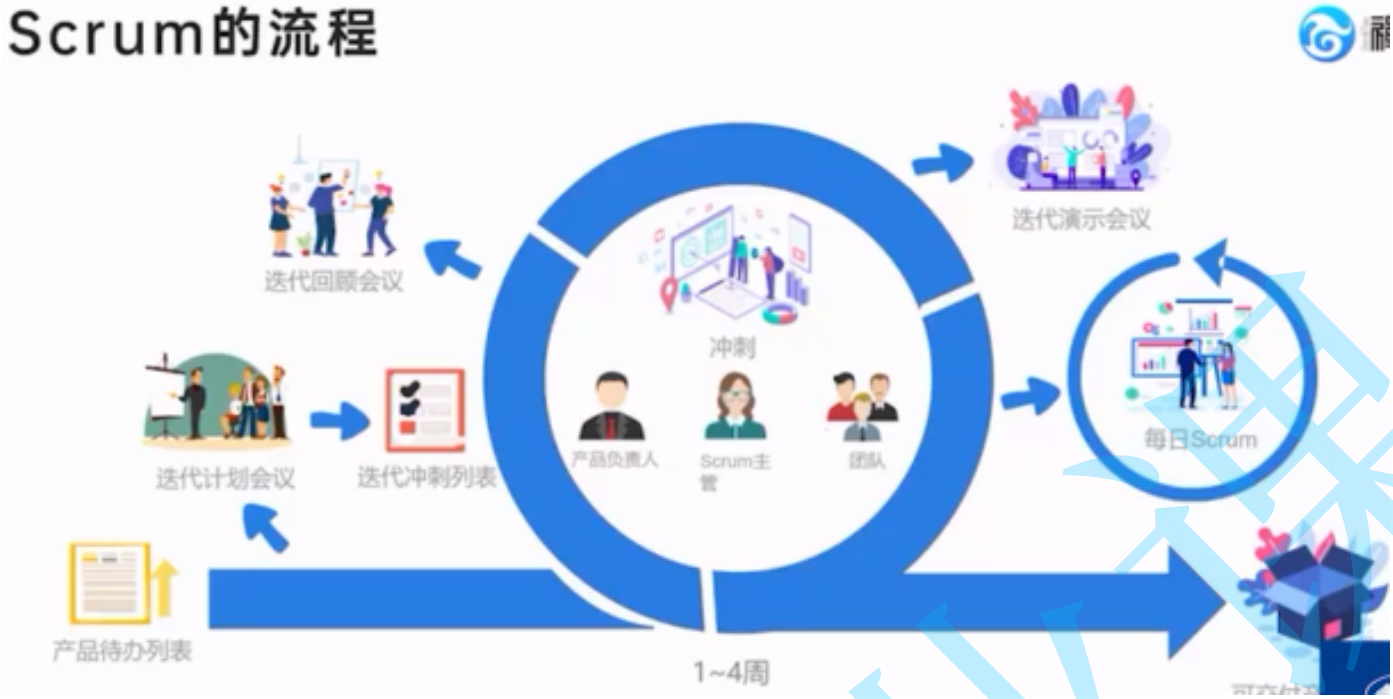
在scrum模型中，主要有**三个角色**和**五个重要会议**。

三个角色：

> scrum由product owner(产品经理)、scrum master(项目经理)和team(研发团队)组成。
>
> - 其中product owner负责整理user story(用户故事)，定义其商业价值，对其进行排序，制定发布计划，对产品负责。
> - scrum master负责召开各种会议，协调项目，为研发团队服务。
> - 研发团队则由不同技能的成员组成，通过紧密协同，完成每一次迭代的目标，交付产品。

迭代开发

与瀑布不同，scrum将产品的开发分解为若干个小sprint(迭代)，其周期从1周到4周不等，但不会超过4周。参与的团队成员一般是5到9人。每期迭代要完成的user story是固定的。每次迭代会产生一定的交付。



scrum的基本流程如上图所示：

- 产品负责人负责整理user story，形成左侧的product backlog。
- 发布计划会议：product owner负责讲解user story，对其进行估算和排序，发布计划会议的产出就是制定出这一期迭代要完成的story列表，sprint backlog。
- 迭代计划会议：项目团队对每一个story进行任务分解，分解的标准是完成该story的所有任务，每个任务都有明确的负责人，并完成工时的初估计。
- 每日例会：每天scrum master召集站立会议，团队成员回答昨天做了什么今天计划做什么，有什么问题。
- 演示会议：迭代结束之后，召开演示会议，相关人员都受邀参加，团队负责向大家展示本次迭代取得的成果。期间大家的反馈记录下来，由po整理，形成新的story。
- 回顾会议：项目团队对本期迭代进行总结，发现不足，制定改进计划，下一次迭代继续改进，以达到持续改进的效果。

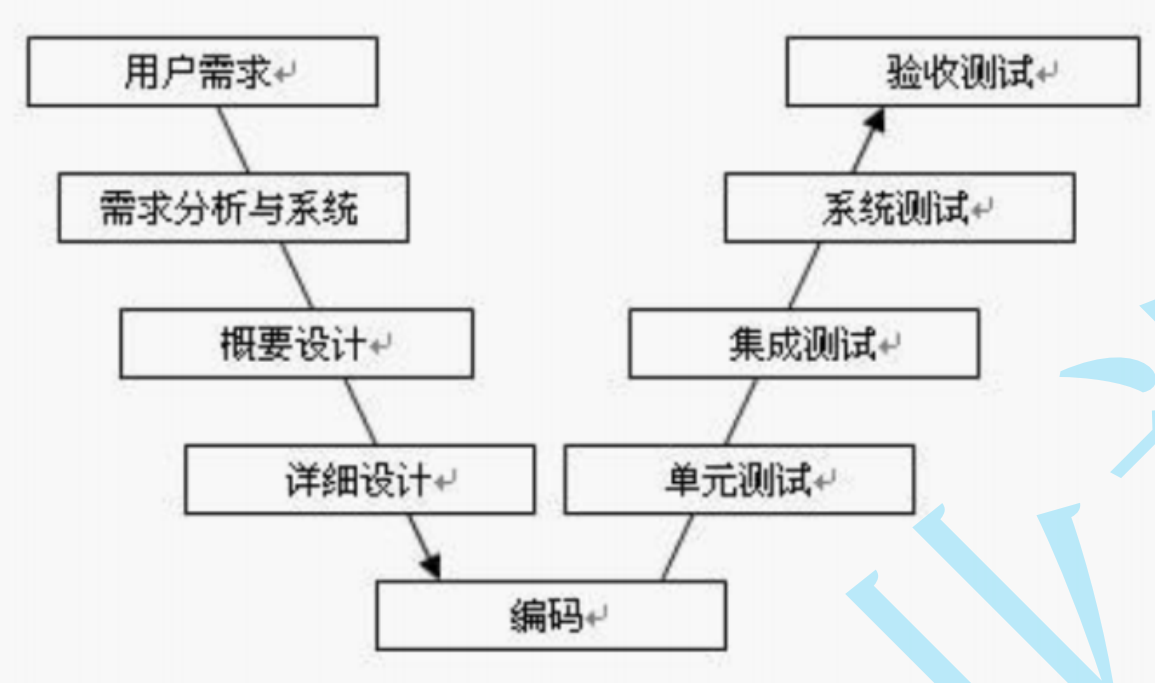
敏捷中的测试

> - 轻文档和快速迭代
>   - 敏捷模型中强调轻文档，所以测试人员不应使用传统的Excel编写测试用例的方法，更多的是使用思维导图、探索性测试（强调自由度，设计和执行同时进行，根据测试结果不断调整测试计划）、自动化测试等
>   - 敏捷讲求合作，在敏捷项目组中，测试人员应多主动跟开发人员了解需求、讨论设计、一起研究bug出现的原因。

2.4 测试模型

测试模型中有两个非常重要且具有标志性的测试模型：V模型和W模型

2.4.1 V模型



V模型最早是由Paul Rook在20世纪80年代后期提出的，目的是改进软件开发的效率和效果。是瀑布模型的变种 。

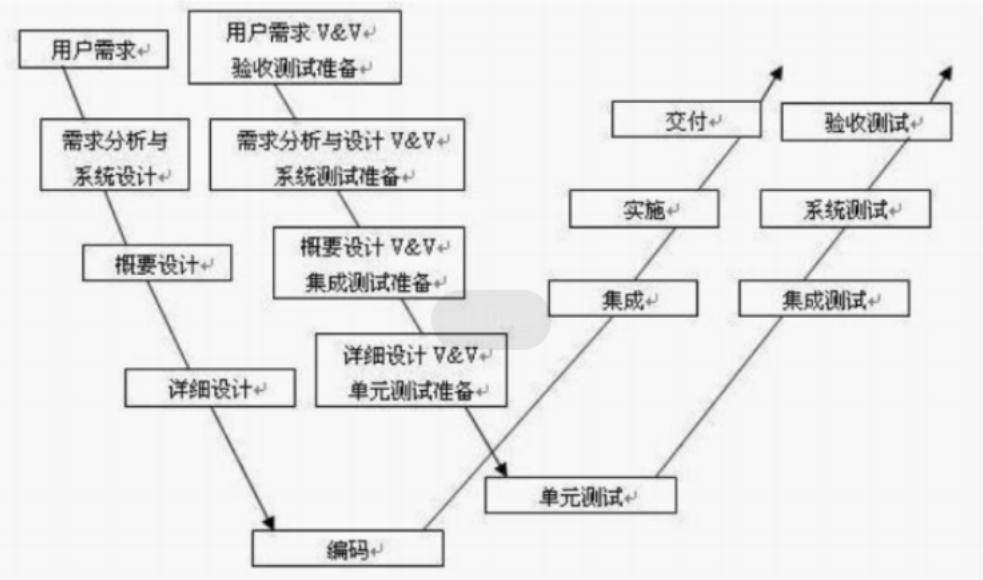
优点：

- **明确的标注了测试过程中存在的不同类型的测试**，并且清楚的描述了这些测试阶段和开发过程期间各阶段的对应关系，有效提升测试的质量和效率。
- V模型指出：
  - 单元和集成测试应检测程序的执行是否满足软件设计的要求；
  - 系统测试应检测系统功能、性能的质量特性是否达到系统要求的指标；
  - 验收测试确定软件的实现是否满足用户需要或合同的要求

缺点：仅仅把测试作为在编码之后的一个阶段，未在需求阶段就介入测试。缺点同瀑布模型。

2.4.2 W模型(双V模型)

> V模型中未将测试前置的问题在W模型中得以解决。



W模型增加了软件各开发阶段中应同步进行的验证和确认活动。W模型由两个V字型模型组成，分别代表测试与开发过程，图中明确表示出了测试与开发的并行关系。

特点：测试的对象不仅是程序，需求、设计等同样要测试，测试与开发是同步进行的

优点：

- 有利于尽早地全面的发现问题。例如，需求分析完成后，测试人员就应该参与到对需求的验证和确认活动中，以尽早地找出缺陷所在。同时，对需求的测试也有利于及时了解项目难度和测试风险，及早制定应对措施，显著减少总体测试时间，加快项目进度。

缺点：

- 需求、设计、编码等活动被视为串行的；
- 测试和开发活动也保持着一种线性的前后关系，上一阶段完全结束，才可正式开始下一个阶段工作。
- 重流程，无法支持敏捷开发模式。对于当前软件开发复杂多变的情况，W模型并不能解除测试管理面临着困惑。