删除水印 万兴PDF

软件测试教程 第二节 概念篇

从这节课开始，我们将开始正式进入测试课程，在开始第一次软件测试之前，我们需要先了解软件测试的一些基本概念。

这些基本概念将帮助我们更加明确工作的目标，以便于更快的融入到测试团队中去

在这里我们将回答以下问题：

> - 什么是需求
> - 什么是bug
> - 什么是测试用例
> - 开发模型和测试模型
> - 配置管理和软件测试

衡量软件测试结果的依据—需求

需求的概念

> 满足用户期望或正式规定文档（合同、标准、规范）所具有的条件和权能，包含用户需求和软件需求。

IEEE定义：软件需求是 (1)用户解决问题或达到目标所需条件或权能(Capability)。 (2)系统或系统部件要满足合同、标准、规范或其它正式规定文档所需具有的条件或权能。 一种反映上面(1)或(2)所述条件或权能的文档说明。它包括功能性需求及非功能性需求，非功能性需求对设计和实现提出了限制，比如性能要求，质量标准，或者设计限制。

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

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。

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。

大多数公司在进行软件开发的时候会把用户需求转化为软件需求，开发人员和测试人员工作的直接依据就是软件需求

软件需求是测试人员进行测试工作的基本依据。

```
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：
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。
```

删除水印 万兴PDF

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至15	字符型	
2	电子邮箱	必填，录入电子邮箱		字符型	
3	密码	必输，输入的密码隐藏*号显示，最短6位	6至15	字符型	
4	确认密码	必输，输入的密码隐藏*号显示，最短6位	6至15	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

1) 若用户不同意协议，系统禁止用户注册。

2) 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

1. 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

1. 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。

2. 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。

问题：

制作一个声控灯，不要从测试人员角度，说一下对这个灯的需求；

从软件测试人员角度看需求

需求是测试人员开展软件测试工作的依据。

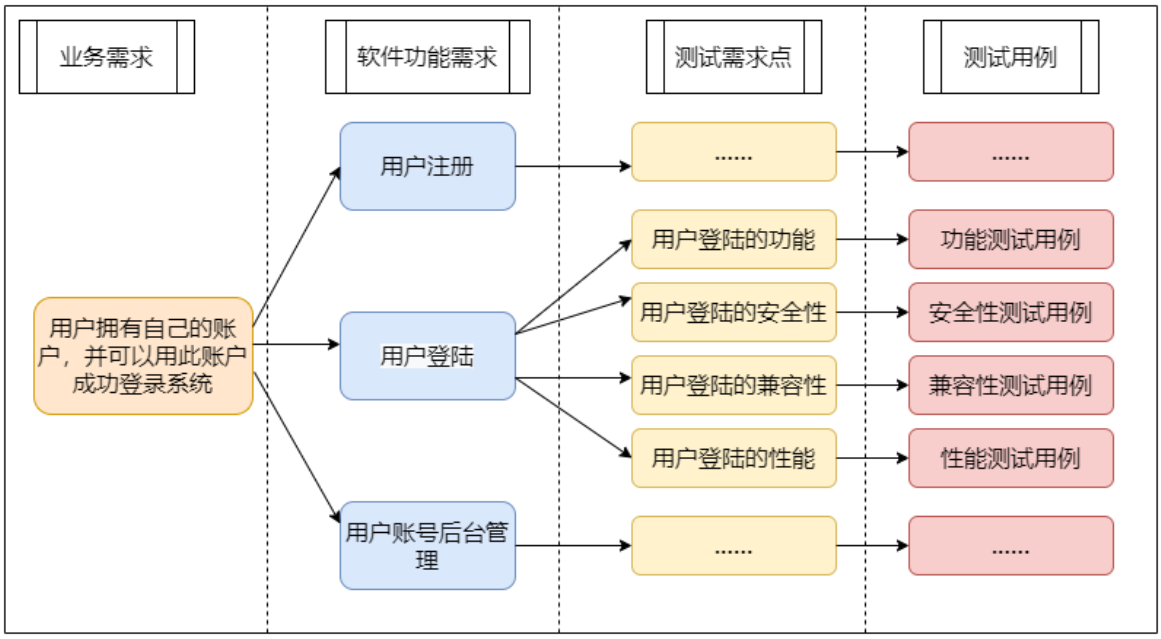
在具体设计测试用例的时候，首先需要搞清楚每一个业务需求对应的多个软件功能需求点，然后分析出每个软件功能需求点对应的多个测试需求点，然后针对每个测试需求点设计测试用例。

过程如下，业务需求—>软件功能需求点—>测试需求点—>测试用例

比特就业课

删除水印 万兴PDF

以“用户登陆”为例，来阐述下整个过程：



为什么需求对软件测试人员如此重要

- 从软件功能需求出发，无遗漏的识别出测试需求是至关重要的，这将直接关系到用例的**测试覆盖率**
- 对于识别出的每个测试需求点，需要采用**具体的设计测试用例的方法**来进行测试用例的设计

如何才可以深入理解被测试软件的需求

测试工程师在需求分析和设计阶段就开始介入，因为这个阶段是理解和掌握软件的原始业务需求的最好时机。

只有真正理解了原始业务需求之后，才有可能从业务需求的角度去设计针对性明确，从终端用户的使用场景到端到端的覆盖率较高的测试用例集。

测试用例的概念

测试用例（Test Case）是为了实施测试而向被测试的系统提供的一组集合，这组集合包含：测试环境、操作步骤、测试数据、预期结果等要素。

测试用例解决了两大问题：测什么，怎么测。

比特就业课

删除水印 万兴PDF

测试用例 ecsp-439: 用户注册成功	
步骤动作:	期望的结果:
进入注册页面，选择注册	系统展现注册页面
输入符合要求的单位名称、单位邮箱、密码、确认密码、组织机构代码、验证码，并确认同意《用户注册协议》，提交注册信息	系统进行注册操作，发送激活邮件。注册成功后，跳转到注册成功页面，并提示用户进行激活操作。
进入注册用的邮箱，进行激活操作	激活成功
用注册的邮箱和密码，进行登录操作	登录成功，系统展示欢迎页面
测试方式	手工
重要性	重要
测试环境	CHROME,IE10+
测试前提	系统运行正常，邮件服务器已开启
功能模块	注册登录

测试过程中可能会遇到以下问题： –不知道是否较全面的测试了所有功能 –测试的覆盖率无法衡量 –对新版本的重复测试很难实施 –存在大量冗余测试影响测试效率

测试用例的产生就是为了解决上述的问题。

> 巩固一下完成一个用例的小例子：
> 手机打电话？

比特就业课 万兴PDF

软件错误（BUG）的概念

第一个bug：

1945年9月的某天，在一间老式建筑里，从窗外飞进来一只飞蛾，此时Hopper正埋头工作在一台名为Mark II的计算机前，并没有注意到这只即将造就历史事件的飞蛾。这台计算机使用了大量的继电器（电子机械装置，那时还没有使用晶体管）。突然，Mark II死机了。Hopper试了很多次还是不能启动，他开始用各种方法查找问题，最后定位到了某个电路板的继电器上。Hopper观察这个继电器，惊奇地发现一只飞蛾已经被继电器打死。Hopper小心地用镊子将飞蛾夹出来，用透明胶布贴到“事件记录本”中，写上“第一个发现虫子的实例”。Hopper的事件记录本，连同那只飞蛾，现在都陈列在美国历史博物馆中。 软件错误的一般定义：`程序与规格说明之前不匹配`。

注意：以上说法是片面的，准确的来说：**当且仅当规格说明是存在的并且正确，程序与规格说明之间的不匹配才是错误。**

当需求规格说明书没有提到的功能，判断标准以最终用户为准：**当程序没有实现其最终用户合理预期的功能要求时，就是软件错误。**

删除水印 万兴PDF

开发模型和测试模型

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。

软件的生命周期

软件生命周期是指从软件产品的设想开始到软件不再使用而结束的时间。 如果把软件看成是有生命的事物，那么软件的生命周期可以分成6个阶段，即需求分析、计划、、设计、编码、测试、运行维护。

瀑布模型（Waterfall Model）

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。

优点： -强调开发的阶段性； -强调早期计划及需求调查； -强调产品测试。 缺点： -依赖于早期进行的唯一一次需求调查，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反馈应用于本产品的过程； **-风险往往迟至后期的测试阶段才显露，因而失去及早纠正的机会。**

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。

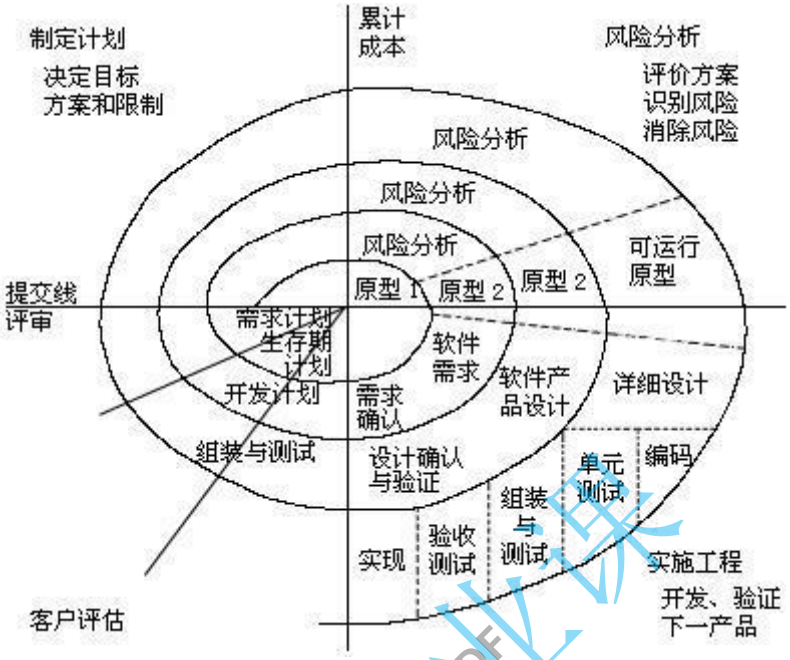
在瀑布模型中，测试阶段处于软件实现后，这意味着必须在代码完成后有足够的时间预留给测试活动，否则将导致测试不充分，从而把缺陷直接遗留给用户。

删除水印 万兴PDF

螺旋模型（Spiral Model）

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： –强调严格的全过程风险管理。 –强调各开发阶段的质量。 –提供机会检讨项目是否有价值继续下去。 •缺点： –引入非常严格的风险识别、风险分析和风险控制，这对风险管理的技能水平提出了很高的要求。这需要人员、资金和时间的投入。

增量、迭代

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。

增量通常和迭代混为一谈，但是其实两者是有区别的。增量是逐块建造的概念，例如画一幅人物画，我们可以先画人的头部，再画身体，再画手脚……而迭代是反复求精的概念，同样是画人物画，我们可以采用先画整体轮廓，再勾勒出基本雏形，再细化、着色。

敏捷

2001年，以Kent Beck、Alistair Cockburn、Ward Cunningham、Martin Fowler等人为首的“轻量”过程派聚集在犹他州的Snowbird，决定把“敏捷”(Agile)作为新的过程家族的名称。

在会议上，他们提出了《敏捷宣言》（http://agilemanifesto.org/）： 我们通过身体力行和帮助他人来揭示更好的软件开发方式。经由这项工作，我们形成了如下价值观。

> 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
> 在每对比对中，后者并非全无价值，但我们更看重前者。

删除水印 万兴PDF

由敏捷宣言可以看出，敏捷其实是有关软件开发的社会工程(Social Engineering)的。敏捷的主要贡献在于他更多地思考了如何去激发开发人员的工作热情，这是在软件工程几十年的发展过程中相对被忽略的领域。

敏捷开发有很多种方式，其中scrum是比较流行的一种。

scrum

scrum里面的角色

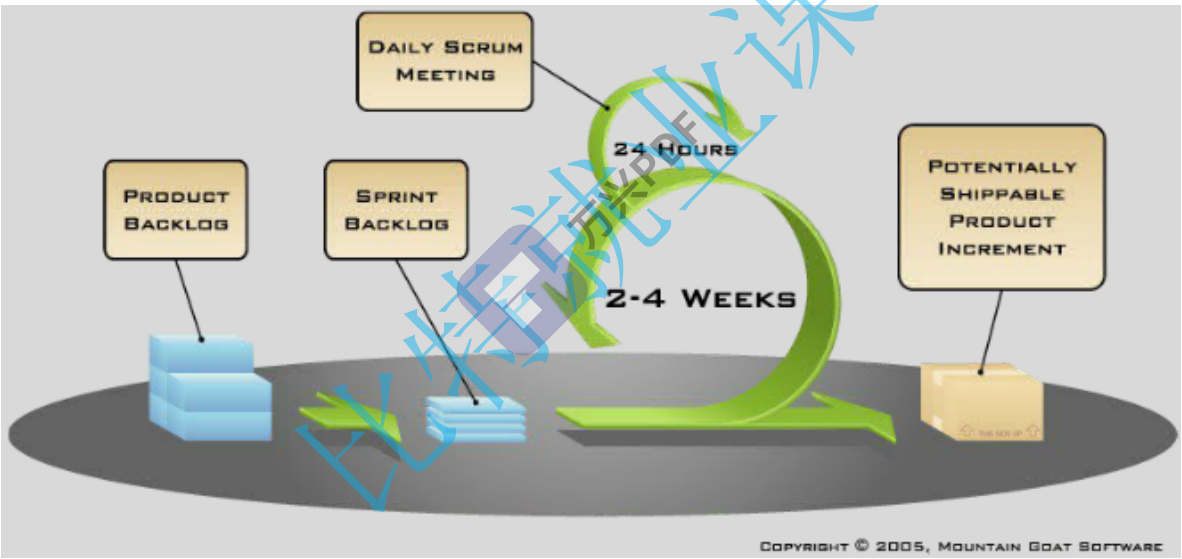
scrum由product owner(产品经理)、scrum master(项目经理)和team(研发团队)组成。

- 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



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，已达到持续改进的效果。

敏捷中的测试

挑战1：轻文档

删除水印 万兴PDF

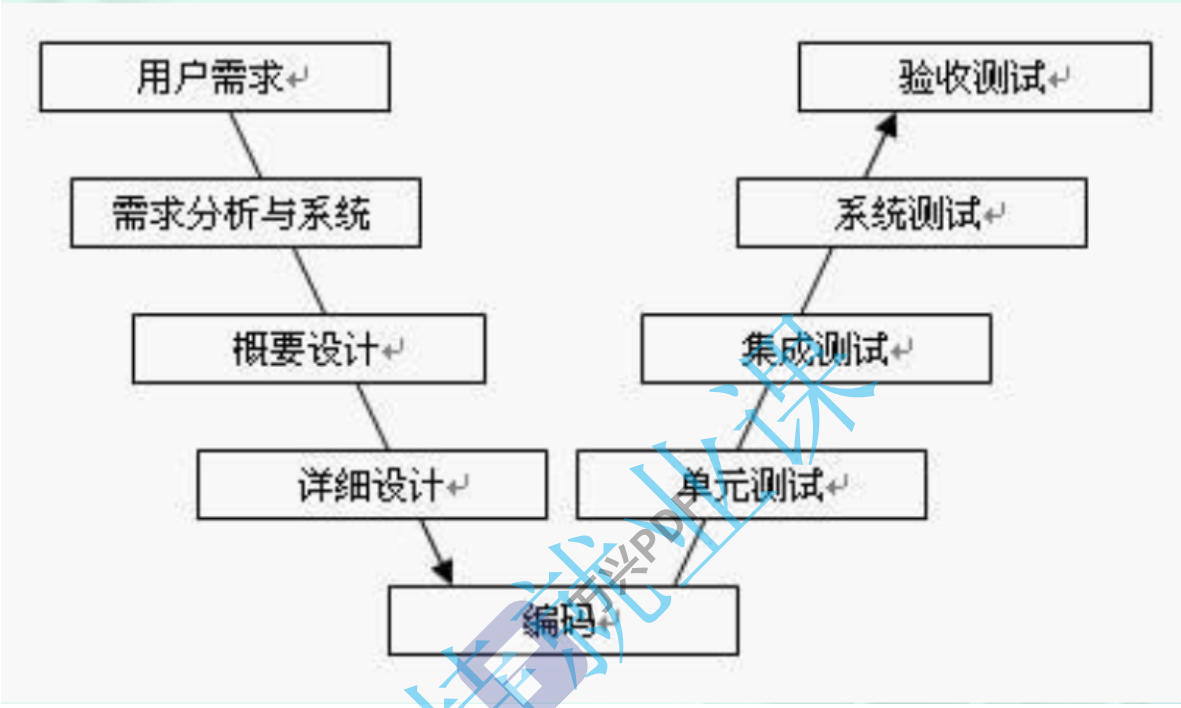
挑战2：快速迭代

1、测试工作的核心内客是没有变的，就是不断地找Bug，只是要调整好自己的心态，一切以敏捷的原则为主。

2、测试人员不能依赖文档，测试用例作用减弱，更多的采用思维导图、探索性测试（强调自由度，设计和执行同时执行，根据测试结果不断调整测试计划）、自动化测试

3、敏捷讲求合作，在敏捷项目组中，测试人员应该更主动点，多向开发人员了解需求、讨论设计、一起研究Bug出现的原因。

软件测试v模型

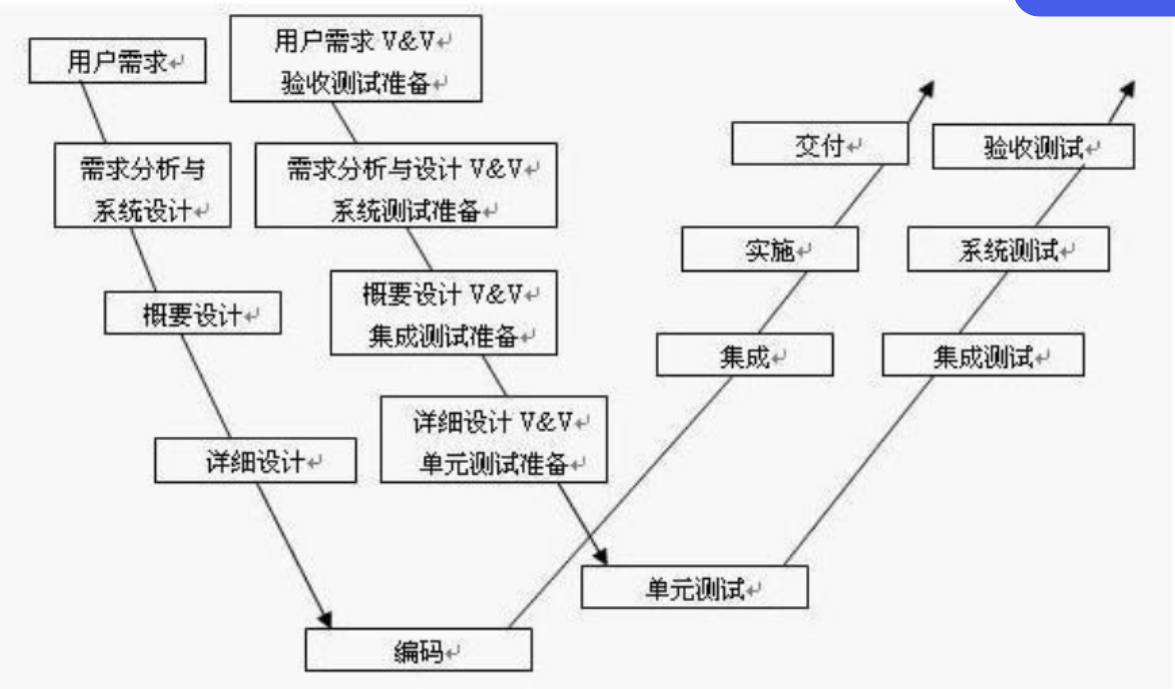


V模型最早是由Paul Rook在２０世纪８０年代后期提出的，目的是改进软件开发的效率和效果。是瀑布模型的变种

- 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
- 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
- 局限性:仅仅把测试作为在编码之后的一个阶段，未在需求阶段就进入测试

软件测试W模型

删除水印 万兴PDF



- 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。
- W模型特点：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
- W模型优点：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。
- 局限性：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。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。

比特测试学院