

1 软件测试基础

1.1 软件测试概述

1.1.1 什么是软件测试

软件测试是伴随着软件的产生而产生的，有了软件生产和运行就必然有软件测试。早期的软件开发过程中，对测试的投入极少，测试介入得也晚，常常是等到形成代码，产品已经基本完成时才进行测试。

到 20 世纪 70 年代，尽管对“软件工程”的真正含义还缺乏共识，但这一词条已经频繁出现。当时对软件测试的定义是：“测试是为了发现错误而执行的一个程序或者系统的过程”。直到 20 世纪 80 年代早期，软件测试定义发生了改变，测试不单纯是一个发现错误的过程，而且包含软件质量评价的内容软件开发人员和测试人员开始坐在一起探讨软件工程和测试问题。制定了各类标准，包括 IEEE 标准、美国 ANSI 标准以及 ISO 国际标准。1983 年，Bill Hetzel “软件测试完全指南”一书中指出：“测试是以评价一个程序或者系统属性为目标的一种活动，测试是对软件质量的度量”。

20 世纪 90 年代，测试工具终于盛行起来。人们普遍意识到工具不仅是有用的，而且要对今天的软件系统进行充分的测试，工具是必不可少的。此时对软件测试的定义是：“测试是为了度量和提高被测软件的质量，对测试软件进行工程设计、实施和维护的整个生命周期过程”。

近 20 年来，随着计算机和软件技术的飞速发展，软件测试技术研究也取得了很大的突破，测试专家总结了很好的测试模型，比如著名的 V 模型、W 模型等，在测试过程改进方面提出了 TMM 的概念，在单元测试、自动化测试、负载压力测试以及测试管理等方面涌现了大量优秀的软件测试工具。

目前没有公认非常完整的定义形式。1983，IEEE 提出的软件工程标准术语，软件测试定义如下：“使用人工和自动手段来运行或测试某个系统的过程，其目的在于检验它是否满足规定的需求或是弄清预期结果与实际结果之间的差别”。

我们知道，软件是由文档、数据以及程序组成的，那么软件测试就应该是对软件形成过程的文档、数据以及程序进行的测试，而不仅仅是对程序进行的测试。

随着人们对软件工程化的重视以及软件规模的日益扩大，软件分析、设计的作用越来越突出，而且有资料表明，60%以上的软件错误并不是程序错误，而是分析和设计错误。因此，做好软件需求和设计阶段的测试工作就显得非常重要。这就是我们提倡的测试概念扩大化，提倡软件生命周期测试的理念。

1.1.3 软件测试的目的

早期的软件定义指出软件测试的目的是寻找错误，并且尽最大的可能找出最多的错误。Grenford J. Myers 就软件测试目的提出了以下观点。

- 1) 测试是程序的执行过程，目的在于发现错误；
- 2) 一个好的测试用例在于能发现至今未发现的错误；

3) 一个成功的测试是发现了至今未发现的错误的测试。

测试的目的,是想以最少的人力、物力和时间**找出**软件中潜在的各种**错误和缺陷**,通过修正各种错误和缺陷**提高软件质量**,**回避**软件发布后由于潜在的软件缺陷和错误造成的**隐患**所带来的**商业风险**。

同时,测试是以评价一个程序或者系统属性为目标的活动,测试是对软件质量的度量与评估,以验证软件的质量满足用户的需求的程度,为用户选择与接受软件提供有力的依据。

此外,通过分析错误产生的原因还可以帮助发现当前开发工作所采用的软件过程的缺陷,以便进行软件过程的改进。同时,通过对测试结果的分析整理,还可以修正软件开发规则,并为软件可靠性分析提供依据。

当然,通过最终的验收测试,也可以证明软件满足了用户的需求,树立人们使用软件的信心。

1.2 软件测试的职业发展

1.2.1 软件测试的主要工作

软件测试工程师一般会承担以下一些具体工作:

- **编写测试用例,进行测试设计、写作测试文档**(测试计划、测试方案、测试用例等);
- **执行测试,发现软件缺陷,提交缺陷报告,并确认缺陷最终得到了修正;**
- 检视代码、评审开发文档、评审测试文档
- 通过测试度量软件的质量
- 其他辅助工作,如:测试脚本开发、编写产品说明、客户培训

1.2.2 软件测试的职业发展方向

软件测试在职业发展上,概括来说可以分为“管理”和“技术”两大类。除此之外,软件测试还可以在质量领域发展。

1) 管理方向

目前不同的软件公司对软件测试管理方面的职位设置一般包括了**测试组长、测试经理、测试主管和测试总监**等等。

测试组长管理测试工程师,并不是完全的管理者,一般是进行测试计划的制订、编写、执行。在测试方面,他们会负责产品的测试重点和难点,还会负责带领新员工使得测试组内的测试工作进展顺利。

测试经理或者测试主管,要为产品负责,他们的重要工作是运作测试项目,制定和执行测试计划,测试结束后进行测试总结报告,对产品的质量进行评估,给出产品的发布建议。他们需要掌握更多项目管理知识,理解项目的**目标**,做好项目的**测试管理、质量管理、成本管理、时间管理、风险管理和人力的管理**,同时还要和开发部门,市场部门或者其他相关部门**密切配合、紧密合作**,所以对**沟通协调**的能力也有要求。同时在测试方面,他们也会负责产品的测试重点和难点。

测试总监要为产品线或者公司负责,他们也需要对**测试团队的发展**负责,进行**人员的招聘和培养**,留住关键人才,提高或者更新不合格的人员,提升团队的胜任力和职业能力,负

责项目的预算和控制；负责资源的计划和分配；持续改进测试的能力，提升效率和产品质量，从测试的角度对产品的成本、周期和质量负责。

当然，管理者并不是纯粹的管理者，他们依然需要保持对软件测试各种技术的领先性，这样管理工作才能顺利展开。

	组长	经理	总监
工作内容	带新人，重难点技术测试，具体测试执行	项目测试运作，计划，报告，质量；也做具体执行	全局性，方向性的管理；人员招聘，技术选型，策略，质量把控，不做具体执行；
项目多少	单个项目测试	多个项目	所有项目
负责人数	4-5 人	8-10 人	15+人
技术级别	中级	高级	资深
工作年限	3 年左右	7 年左右	10 年左右

2) 技术方向

技术方向有产品测试专家和专项测试工程师两方面。

产品测试专家，有的公司称为测试架构师，他们在业务需求向测试技术转换的过程中起了桥梁作用，负责产品测试的整体架构设计。他们负责对产品测试中的技术重点和难点进行研究和攻关；他们为测试组织提供最优的测试方法。他们负责协助测试经理制定测试项目计划和控制测试项目进度。他们负责测试团队内部的技术培训工作。他们需要有一定的战略规划能力、数据分析处理能力、面向生命周期的质量保证和持续改进能力。

专项测试工程师就是从事专业领域的测试工程师，比如功能测试、自动化测试、性能测试、安全测试等；早期可以从事全面的功能测试，后期特别需要在专业领域有技术深度的测试工程师，因此分化出了专门做自动化测试、性能测试、安全渗透测试方面的专职岗位。

从测试的体系角度看，软件测试体系中的任何一个环节，测试策略、测试分析设计、测试执行、测试评估、测试流程等每个领域的内涵都很丰富，包含了很多可以深入发展研究的技术，比如自动化测试技术、产品测试模拟工具开发、测试流程管理工具开发、缺陷。

测试工程师在自己的职业生涯的发展晋级路线：初级测试工程师->中级测试工程师->高级测试工程师->资深测试工程师->测试专家。

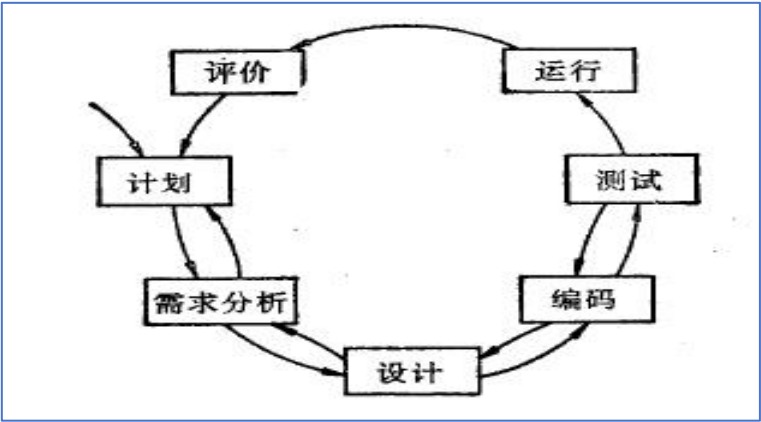
3) 质量方向

软件测试在质量领域的发展可以参考如下几个方向：

- 产品流程设计：负责企业在产品开发、市场、交付等全流程体系建设。
- 企业质量管理者：企业质量管理是企业管理中一个重要的组成部分。这里的质量不仅仅指的是产品质量，而是企业整体质量的提升。
- 客户满意度管理者：客户满意度管理，前提还是需要对用户的需求有很好的理解，这个需求不仅仅局限于产品，而是客户可感知的方方面面，如产品推广、投标达标、供货保障、工程交付、技术支持、备件支持和客户培训等。

1.4 软件生命周期

软件生命周期的各个阶段（基于瀑布模型的软件生命周期）



生命周期各阶段的工作内容及项目实例如下：

阶段	工作内容	实例
计划	<ul style="list-style-type: none">确定软件开发总目标；给出软件的功能、性能、可靠性以及接口等方面的设想；研究完成该项目的可行性，探讨问题解决方案；对可供开发使用的资源、成本、可取得的效益和开发进度做出估计；制定完成开发任务的实施计划。	<ul style="list-style-type: none">研发一个电商网站；支持用户网上购物，包括在线浏览商品、加入购物车、提交订单、在线支付等业务，且相关业务需要在一定时间之内完成，如支付订单需要在 2 秒内给出返回信息；该项目目前已经有比较成熟的竞品如淘宝、京东等电商平台，商业可行性和技术实现上不存在障碍；需要在 3 个月之内完成所有开发和测试工作，并推向市场；具体计划可参考项目计划。
需求分析	<ul style="list-style-type: none">对开发的软件进行详细的定义，由需求分析人员和用户共同讨论决定，哪些需求是可以满足的，并且给予确切的描述，写出软件需求说明书 SRS (Software Requirement Specification)。	<p>功能需求：</p> <ul style="list-style-type: none">商品：商品添加、更新、浏览、上下架订单：提交订单、结算、支付会员： 用户注册、登录、购物车 <p>性能需求：</p> <ul style="list-style-type: none">普通页面访问需在 2 秒内完成关键业务需要在 1 秒内完成如：注册支持 2000 人同时在线访问；
设计	<ul style="list-style-type: none">设计是软件工程的技术核心，这个阶段需要完成设计说明书。概要设计（HLD），在设计阶段把各项需求转换成相应的体系结构，每一部分是功能明确的模块。详细设计（LLD），对每个模块要完成的工作进行具体的描述。	<p>概要设计</p> <ul style="list-style-type: none">整个网站分为前台和后台两部分，前台包括：首页、会员中心、商品列表、购物车等模块。后台包括：商品管理、订单管理、会员管理、促销管理、系统设置等模块。各模块包含有详细的子模块，如商品管理模块包括子功能：商品添加、商品类目、商品修改、

		上下架等功能 详细设计 <ul style="list-style-type: none"> 具体的功能的业务流程或者伪码，包括若干个函数方法。
编码	<ul style="list-style-type: none"> 把软件设计转换成计算机可以接受的程序，即写成以某个程序设计语言表示的源程序清单，使用 RDBMS 工具建立数据库。 	<ul style="list-style-type: none"> 用 PHP 语言实现详细设计说明书中描述的所有函数。
测试	<ul style="list-style-type: none"> 测试是检验软件是否符合客户需求,达到质量要求,一般由独立的小组执行,测试工作分为:单元测试、集成测试、系统测试。 	<ul style="list-style-type: none"> 单元测试: 参照 LLD, 对每一个函数进行测试 集成测试: 参照 HLD, 对函数与函数的集成、模块与模块的集成进行测试 系统测试: 参照 SRS, 对每一个功能需求、性能需求等进行测试
运维	<ul style="list-style-type: none"> 这个阶段将软件交付用户投入正式使用,以后便进入维护阶段,可能有多种原因需要对它进行修改,如软件错误、系统软件升级、增强软件功能、提高性能等。 	<ul style="list-style-type: none"> 电商网站提供给用户使用,用户在使用过程中如发现问题可通过技术支持人员反映,问题解决后为用户进行软件升级……

1.5 软件研发组织和模型

1.5.1 软件研发相关要素

只有合适的**人员**借助合适的**工具**经过合适的**过程**才能研发出高质量的软件。

工具为人员和过程服务,起辅助作用,起关键作用的是人员和过程。

1.5.2 常见项目组织架构

项目组一般由项目经理领导并负责制定项目计划,分配任务。



项目组内设置开发组、测试组、产品设计组,各组由对应的经理或组长负责。各组包含若干初级、中级、高级工程师。项目组人员的规模视项目的规模大小而定。

项目组参与人员角色及职责如下:

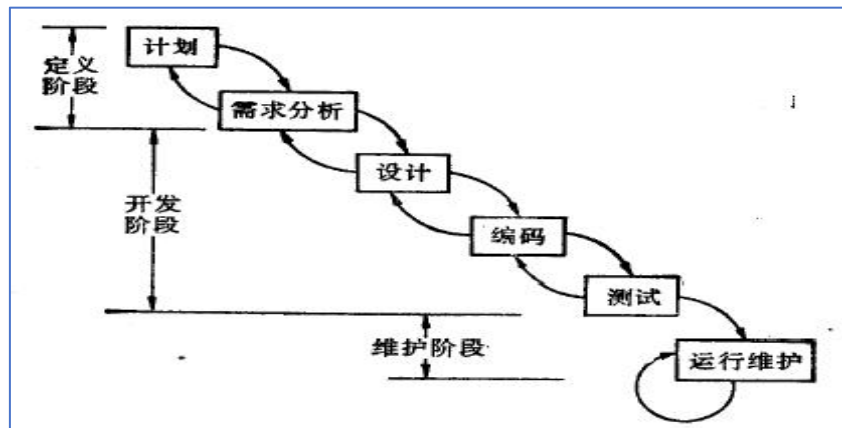
角色名	职责	是否必选
项目经理	负责整个项目的计划、执行、跟踪、发布。 是项目的最终决策人和负责人	Y

技术经理	负责项目的代码开发管理，包括：预估开发工作量、开发任务分配。开发日报发布、详细设计、代码 BUG 跟踪、发布前准备。是在技术开发方面的最终负责人。	Y
产品经理	负责项目的功能性需求与产品设计。 是产品功能设计与交互设计的最终负责人。也是项目 DEMO 演示时的主要参与者之一。	Y
运营经理	负责项目的非功能性需求，主要是运营、推广方面的设计与实施。也是项目 DEMO 演示时的主要参与者之一。	
测试经理	负责项目的测试计划、测试任务分配、BUG 跟踪。 是项目最终是否通过测试，是否可以上线的最终决策人。 也是项目 DEMO 演示时的主要参与者之一。	Y
客服经理	是项目的来自于客户需求的管理者。 也是项目 DEMO 演示时的主要参与者之一。	
开发组长	负责某一个开发小组的管理任务。受技术经理领导。	
架构师	负责整体架构，代码审核。	
开发工程师	负责具体的设计、编码、测试。受开发组长或技术经理领导。	Y
数据库管理员	负责项目的数据库设计的审核、变更，SQL 语句的审核，数据订正，测试或上线数据准备。	
系统管理员	负责环境部署、维护；网站流量或压力监控。	
界面设计师	负责用户界面设计与交互设计。	
产品运营师	负责产品的运营活动。	
产品设计师	负责需求的收集与分析，产品设计与交互设计。	
测试工程师	负责软件的集成测试，包括三个环境下的测试：测试环境、预发布环境、线上环境。	Y
配置管理员 (CMO)	负责本项目的分支管理，代码合并，版本控制。	Y
发布员	负责发布程序到：开发环境，测试环境，线上环境。	Y
软件过程专员 (SQA)	负责监督项目的过程计划与执行情况。协助项目的规范化以及风险告警。 目前测试专员仅协助技术经理，关注于开发过程的管理与执行。	Y

1.6 软件研发模型

常见的软件研发模型有：瀑布模型、螺旋模型、敏捷模型

1.6.1 瀑布模型



瀑布模型要求软件开发严格按照制定计划 → 需求分析 → 软件设计 → 程序编码 → 软件测试 → 运行维护的阶段进行, 每一个阶段都可以定义明确的产出物和验证准则。其过程是从上一项活动接收该项活动的工作对象作为输入, 利用这一输入实施该项活动应完成的内容给出该项活动的工作成果, 并作为输出传给下一项活动。同时评审该项活动的实施, 若确认, 则继续下一项活动; 否则返回前面, 甚至更前面的活动。瀑布模型在每一个阶段完成后都可以组织相关的评审和验证, 只有在评审通过后才能够进入到下一个阶段。由于需要对每一个阶段进行验证, 瀑布模型要求每一个阶段都有明确的文档产出, 对于严格的瀑布模型每一个阶段都不应该重叠, 而应该是在评审通过, 相关的产出物都已经基线后才能够进入到下一个阶段。

瀑布模型的优点:

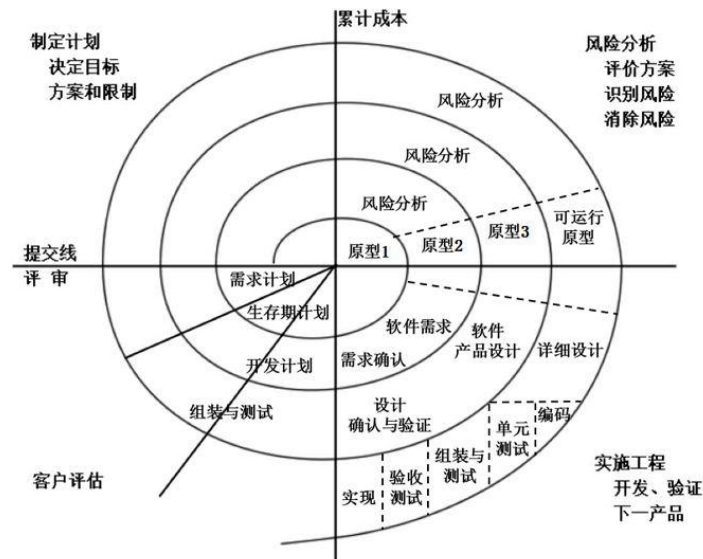
- 1) 项目每个阶段都有相应的检查点。
- 2) 前一个阶段结束就只需要关注后续的阶段。

瀑布模型的缺点:

- 1) 开发过程一般不能逆转, 否则代价非常大, 如果在前面的阶段有错误, 后面就会有很大的损失。
- 2) 软件的实际情况到项目开发的后期客户才能看到, 增加了开发的风险。
- 3) 客户往往很难在一开始就给出所有的确定的需求。所以这样就无法适应客户需求变化。

测试直到项目开发完后才介入。

1.6.2 螺旋模型



螺旋模型综合了基本的瀑布式模型和演化/渐增原型方法。螺旋模型有很多中间版本，这种模型的每一个周期中都包含有需求定义、风险分析、工程实现和评审这几个阶段，软件开发过程每迭代一次，软件开发又前进一个层次，在每一个开发过程中，都会提前进入风险识别、风险分析和风险控制，把软件项目分解成一个个小项目。每个小项目都标识一个或多个主要风险，直到所有主要风险因素都被确定。

螺旋模型的优点：

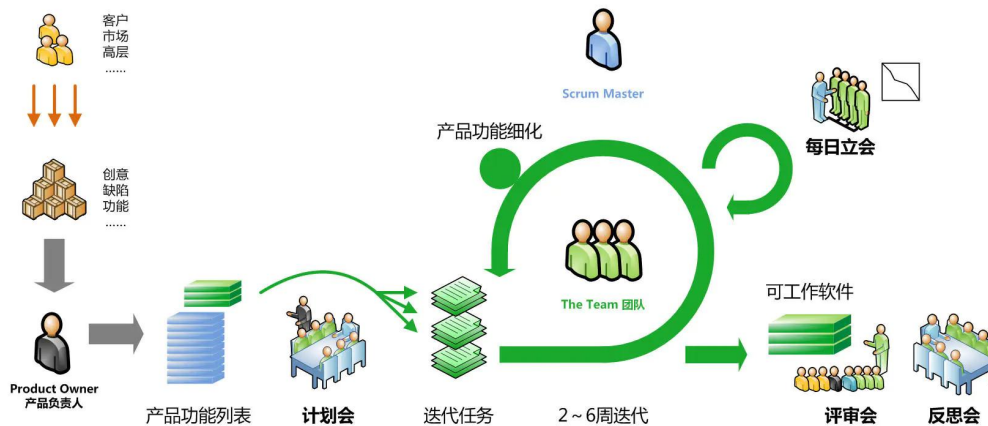
- 1) 可以在项目的每个开发过程的每个阶段变更。
- 2) 项目被分解成一个个小的项目，使得项目分阶段可见、可控。
- 3) 客户能够根据项目分阶段的最新情况了解项目，有效沟通。

螺旋模型的缺点：

- 1) 需要具有丰富的风险评估经验和专门知识，如果风险较大，又未能及时识别出风险，会有损失。
- 2) 过多增加迭代此处会增加开发成本，延迟交付时间。

1.6.3 敏捷模型

- 1) 以人为本
- 2) 及时沟通，快速反馈
- 3) 迭代开发，持续集成
- 4) 极限编程（XP Extreme Programming）



敏捷开发是一种以人为核心、**迭代、循序渐进**的开发方法。敏捷开发是针对传统的瀑布开发模式的弊端而产生的一种新的开发模式，目标是提高开发效率和响应能力。

迭代是指把**一个复杂且开发周期很长的开发任务，分解为很多小周期可完成的任务**，这样的**一个周期就是一次迭代的过程**；同时每一次迭代都可以生产或开发出一个**可以交付的软件产品**。

敏捷模型的特点：

- 1) 人和交互重于过程和工具
- 2) 客户协作重于合同谈判
- 3) 随时应对变化重于循规蹈矩

特点是项**目组内成员作为一个整体工作**，按短迭代周期工作，每次迭代交付一些成果，关注业务优先级，检查与调整。敏捷模型适用于较小的队伍，40、30、20、10 人或者更少。当项目规模增长，面对面的沟通就愈加困难。

1.7 软件测试的原则：

1. 尽早测试，研发中测试介入越早越好；
2. 测试不可能穷尽，只能尽可能覆盖；
3. 测试必须以产品需求为依据，不可主观臆断；
4. 测试过程中 bug 可追溯，可复现，且要闭环；
5. 测试要用尽可能少的用例覆盖尽可能多的需求；

1.8 缺陷产生的根源

- + 软件开发工具或环境问题导致；
- + 软件需求不明确或理解存在偏差；
- + 开发人员技术能力不足或代码不规范；
- + 沟通交流问题：【有效沟通：正确的理解，准确的表达】

