**课程编号：A0801050130**

**软件构造与测试**

**作业报告**



**东北大学软件学院**

**1、主要质量标准和依据**

1. 质量标准

“美丽手语”为“美丽人生”系列产品中的其中一个，具有手语的教学以及翻译功能，通过配套使用，听障用户可以轻松完成与普通人的双向沟通，使得用户教育、交友、婚恋、就业、医疗、安全、生活质量等方面的全面提升。同时，产品同其他“美丽人生”系统软件一样，也内置了健康咨询和健康服务功能，残障人群可以同时享受个性化的服务。

基于其本质为软件产品的特性，我们选用ISO/IEC 9126:1991质量标准。

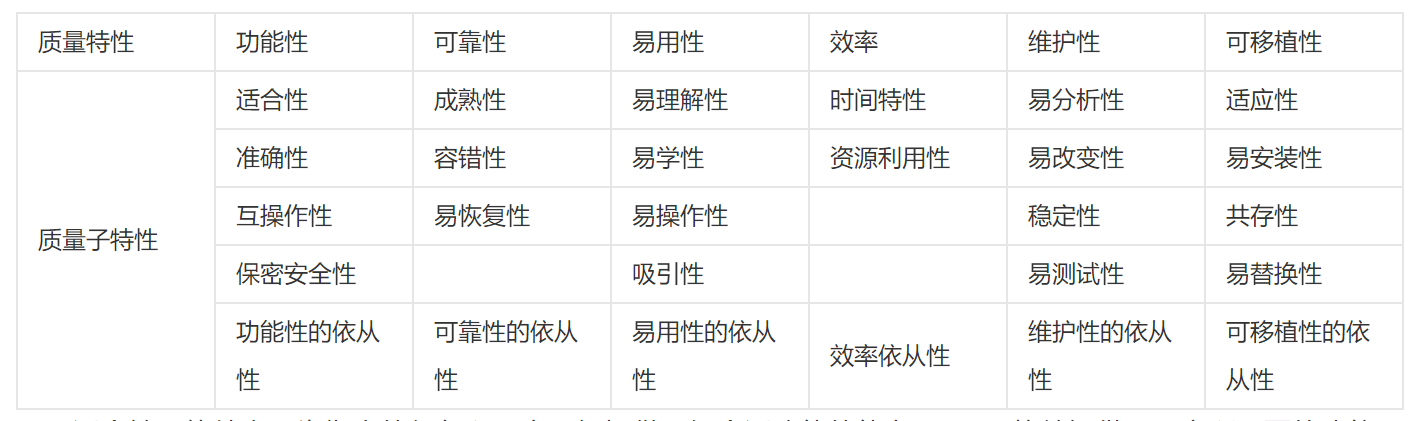
ISO/IEC9126软件质量模型是一种评价软件质量的通用模型，包括3个层次：

1、质量特性

2、质量子特性

3、度量指标

其中各六个质量特性与二十七个质量子特性的关系如下表：



产品应该满足的主要质量标准有：

功能性、可靠性、可使用性、效率、可维护性、可移植性。

功能性——产品满足需求文档和相关标准

可靠性——产品在规定条件下、规定时间内完成规定功能的能力。

易用性——用户理解、学习、使用所付出的时间和努力程度。

效率性——产品执行某项功能所需要的资源和时间。

软件维护性——软件发生错误时，规定条件下、时间内，使用规定的工具或方法修复规定功能的能力。

软件可移植性——从一种环境迁移到另一种环境的能力。

兼容性——与其他产品共享资源、环境、交换并使用信息的程度。

安全性——个人信息保护程度。

1. 剪裁结果及其依据

首先，由于该产品的理念是面向全世界的残疾人，产品就应符合国际标准，因此选用ISO标准。其次，该产品主要针对的用户群体为听障人士，同时需要满足“美丽人生”的功能定位：能够为残疾人提供个性化的健康资讯和健康服务；能够弥补因身体残疾造成的生活障碍，使残疾人的生活像正常人一样便利。因此我们需要ISO/IEC9126标准进行一定的裁剪。

剪裁结果及其依据如下：

1. 功能性：

包括以下几点：

适合性：软件产品为指定的任务和用户目标提供一组合适功能的能力。

准确性：软件提供给用户功能的精确度是否符合目标。

互操作性：软件与其它系统进行交互的能力。

保密安全性：软件保护信息和数据的安全能力。

功能性的依从性：遵循相关标准。

产品必须可以完成功能性的要求：这里主要采用适合性、准确性以及保密安全性：软件提供了用户所需要的功能，软件提供的功能是用户所需要的（比如手语学习翻译功能）；软件能准确地提供用户所需要的功能（比如个性化的健康资讯和健康服务）；软件可以保护用户的个人信息，保证信息不被泄露，软件具有账号密码安全防护。

1. 可靠性：

包括成熟性、容错性、易恢复性和可靠性的依从关系。

成熟性：软件产品为避免软件内部的错误扩散而导至系统失效的能力（主要是对内错误的隔离）

容错性：软件防止外部接口错误扩散而导致系统失效的能力（主要是对外错误的隔离）

易恢复性：系统失效后，重新恢复原有的功能和性能的能力。

可靠性的依从性：遵循相关标准。

主要是软件内部的质量标准，这里全部采用：

表现在当残障人员使用软件时，如果软件发生错误，app不会闪退，系统不会失效，当系统失效时，可以快速恢复，并且数据不会丢失。

1. 易用性：

易理解性：软件交互给用户的信息时，要清晰，准确，且要易懂，使用户能够快速理解软件。

易学性：软件使用户能学习其应用的能力。

易操作性：软件产品使用户能易于操作和控制它的能力。

易用性的依从性：遵循一定的标准。

这里全部采用，因为使用本类产品的人为全球的残障人群，故软件本身的操作要简单易懂，第一次使用的用户易于学习。考虑到产品的使用人群可能含有老年人和其他国家的人群，故软件功能设计不宜过于复杂，且界面尽可能简单易操作，界面字体大小应当可调，软件的语言也可以设置。

1. 效率：

时间特性：软件处理特定的业务请求所需要的响应时间。

资源利用性：软件处理特定的业务请求所消耗的系统资源。

效率依从性：遵循一定的标准。

效率也是评价软件质量的重要标准，这里全部采用：要求该应用响应时间不得超过5秒，超过5秒未响应则增加缓冲提示；该应用耗电应尽可能低，运行时占内存空间应尽可能少。响应时间效率最重要的是体现在手语互译和健康咨询上，用户在使用手语翻译功能时加载时间必须够快，可以提供本地缓存服务；用户在进行健康咨询时要保证低延迟，不影响正常的交流。

1. 可维护性：

易分析性：软件提供辅助手段帮助开发人员定位缺陷产生的原因，判断出修改的地方。

易改变性：软件产品使得指定的修改容易实现的能力。（降低修复问题的成本）

稳定性：软件产品避免由于软件修改而造成意外结果的能力。

易测试性：软件提供辅助性手段帮助测试人员实现其测试意图。

维护性的依从性：遵循相关标准。

当应用发生系统崩溃时，需要给出明显提示，让用户选择等待恢复或关闭软件重启，恢复时间或重启不应超过5秒；当发生系统崩溃时，会及时反馈给开发人员，方便系统漏洞的修复；当软件进行迭代更新时，旧版本的软件应当保证功能性。

1. 可移植性：

适应性：软件产品无需作相应变动就能适应不同环境的能力。

易安装性：尽可能少的提供选择，方便用户直接安装。

共存性：软件产品在公共环境中与其它软件分享公共资源共存的软件。

易替换性：软件产品在同样的环境下，替代另一个相同用途的软件产品的能力。

可移植性的依从性：遵循相关的标准。

产品需要能够在多种移动设备上运行，如手机、平板或笔记本。因为当前人们间交流的途径是多种多样的，既然要做到让患者像正常人一样便利，那么产品需要能够应对各种环境，如使用手机、平板或笔记本进行通话、视频聊天等等。

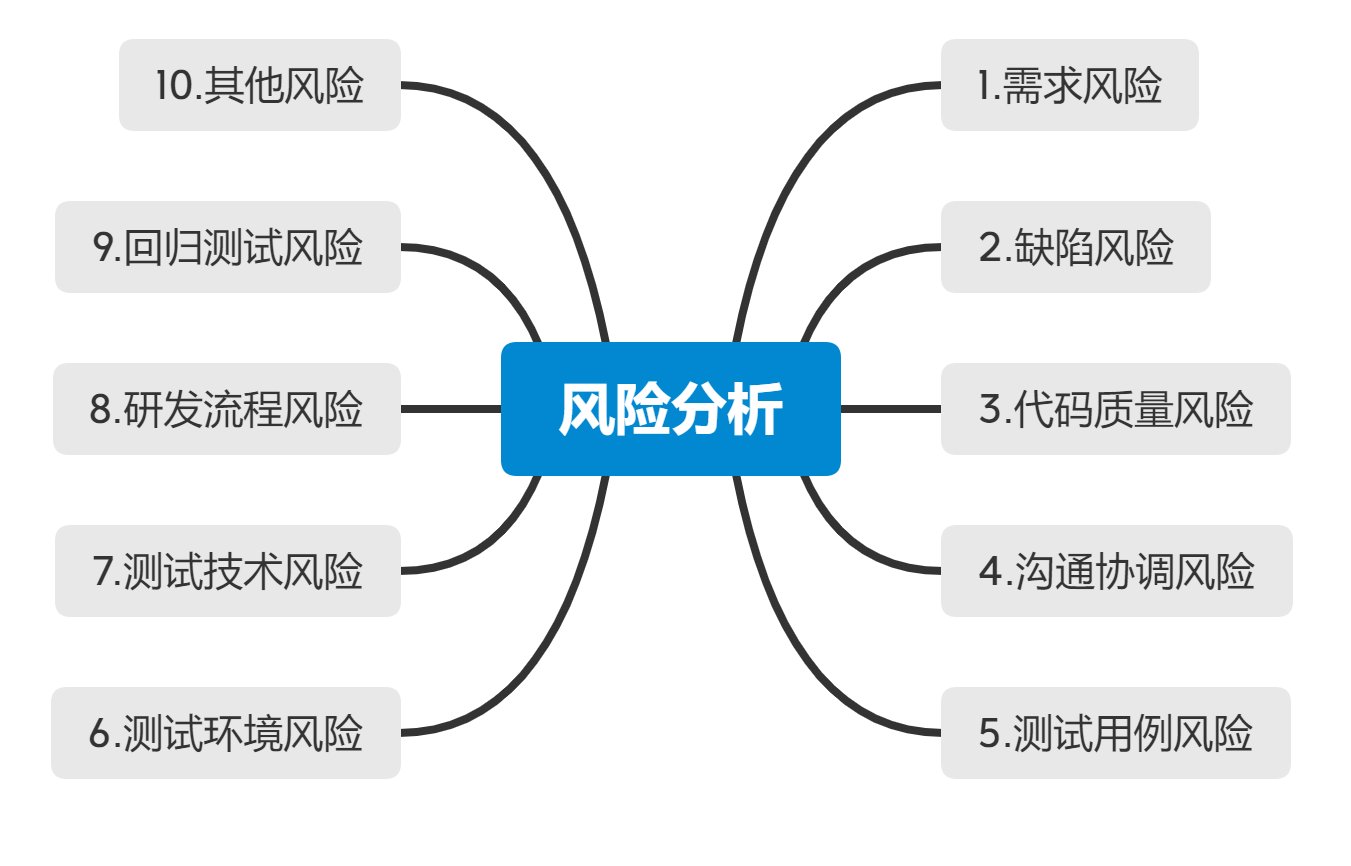
由于“美丽人生”系列也有硬件产品，故软件产品可以具有一定的嵌入性，嵌入到硬件，如智能轮椅，智能拐杖等产品中。

**2、测试风险分析**

1. 具体理由

软件测试的风险是指软件测试过程出现的或潜在的问题，造成的原因主要有测试计划不充分、测试方法有误或测试过程偏离造成测试的补充以及结果不准确。对该产品的测试风险分析列出以下风险：

1. 需求风险
2. 缺陷风险
3. 代码质量风险
4. 沟通协调风险
5. 测试用例风险
6. 测试环境风险
7. 测试技术风险
8. 研发流程风险
9. 回归测试风险
10. 其他不可预计风险



1. 具体理由
2. 需求风险

由于听障有时难以表达自己的想法，则会导致在测试工作中对软件需求的理解不准确，进而使测试范围出现误差，遗漏部分需求或执行错误的测试方法；此外，需求变更也会导致测试用例变更，同步时存在误差。

1. 缺陷风险

软件中一些缺陷偶然出现，难以再现，容易被忽略；

1. 代码质量风险

软体代码质量差，造成大量的缺陷，容易造成测试遗漏；

1. 沟通协调风险

角色之间的沟通、协作，难免会产生误解，沟通不畅，导致项目延期；

1. 测试用例风险

残障人群的具体使用场景十分复杂，这就可能导致测试用例设计不完整，忽略了边界条件、异常处理等情况，使得测试用例没有完全覆盖需求;测试用例没有得到全部执行，部分测试用例被有意或无意的遗漏;

1. 测试环境风险

有的测试环境与生产环境不能完全一致，导致测试结果存在误差；

1. 测试技术风险

部分项目存在技术难度、测试能力和水平等问题，导致项目进度缓慢；

1. 研发流程风险
2. 回归测试风险

通常回归测试没有运行所有测试用例，可能存在测试不完全；

1. 其他不可预计风险

某些紧急情况、不可抗力等也构成风险因素，并且很难预测和避免。

1. 引入新工具后的分析

若在测试过程中需要使用新的配置管理工具，这属于不可预计风险。

首先可能会有测试人员对新工具无法熟练运用的问题，这将会提高测试技术风险。

其次由于不同（版本）的操作系统、不同（版本）的数据库，不同（版本）的网络服务器、应用服务器，再加上不同的系统架构等组合，使得要构建的软件测试环境多种多样；而且现在随着软件运行环境的多样性、配置各种相关参数的“浩大工程”和测试软件的兼容性等方面的需要，使得构建软件测试环境的工作变得较为复杂和频繁，而软件测试环境的构建是否合理、稳定和具有代表性，将直接影响到测试结果的真实性、可靠性和正确性。因此引入新工具也将会提高测试环境风险和缺陷风险。

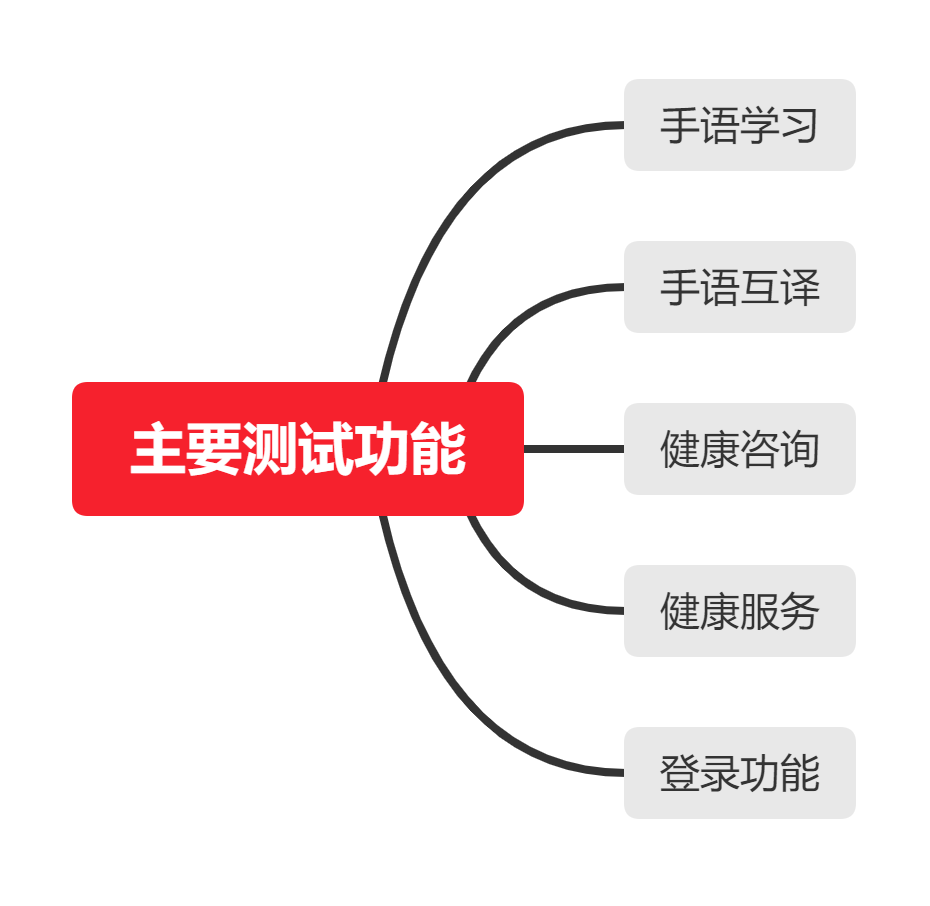
再者，若改变配置管理工具后，未于其他相关部门及时沟通，也会影响项目进度。这提高了沟通协调风险。

**3、总体测试策略**

（1）测试范围

由于该应用旨在为全世界的残疾人提供安全、方便、高效、易用的健康生活服务，根据该定位确定功能测试、界面测试以及非功能性测试：安全性测试、可靠性测试、性能测试等。

1. 功能性测试：
2. 手语学习功能
3. 手语互译功能
4. 个性化的健康资讯和健康咨询服务功能
5. 登录功能



1. 界面测试

由于本产品主要为移动端App，故界面测试也非常重要。

主要测试：

* 1. 界面美观性
  2. 界面合理性
  3. 界面易操作性
  4. 是否符合人机交互的特性

1. 性能测试：
2. 负载测试：测试同一时间访问该应用的用户数上线。
3. 压力测试：测试系统某些资源达到饱和或系统崩溃边缘时能承受的最大压力，最大用户数和最大数据量。
4. 并发测试：模拟用户并发访问，测试多用户同时访问该应用以上某一功能时是否存在死锁或其他问题。
5. 配置测试：测试各种环境（如不同机型、不同系统）对系统性能的影响。
6. 安全性测试：
7. 用户程序安全：系统是否会出现用户冲突；用户登陆密码是否可见可复制、系统是否会因用户权限的改变造成混乱等
8. 网络安全：模拟非授权攻击测试防护系统是否坚固；采用各种木马检查工具检查木马情况；被窃取账号后是否可以立刻修改账号密码以保证安全。
9. 数据库安全：数据是否机密、完整、可管理、独立、可备份、可恢复等。
10. 易用性测试：
11. 用户掌握该应用基本功能所需时间。
12. 是否有引导性操作，提升用户体验感。
13. 兼容性测试
14. 测试软件是否能在不同的操作系统平台上兼容，或测试软件是否能在同一操作平台的不同版本上兼容（和“美丽人生”的其他硬件能否兼容）；
15. 软件本身能否向前或向后兼容；
16. 测试软件能否与其他“美丽人生”相关的软件兼容；
17. 数据兼容性测试，主要是指数据能否共享等；
18. 测试目标

通过测试发现软件可能存在的缺陷并防止缺陷发生、获取对软件质量的信心并提供软件质量的信息、确保软件的功能符合用户的需求、确保软件满足了性能的要求、确保软件是健壮且适应用户环境的、为软件的质量评估和改进提供依据。

1. 测试重点

1.确定产品价值：

该产品致力于让高科技服务于残疾人，提高残疾人的生活质量，适用听障人士使用，且是面向全世界，它不仅要提供丰富的功能，更要安全且易于操作。

2.达到质量目标：

达到功能性、可靠性、安全性、效率性、可维护性、可移植性、兼容性、易用性的要求，提供安全、方便、高效、易用的健康生活服务

3.确定被测功能在系统中的地位：

客户最关注的功能显然应该得到测试的重视，需要投入较多的精力对其进行验证。除此之外，对那些非核心功能、但对核心功能产生影响的模块，我们也不应该忽视，至少要保证这些模块在一般情况下能够正常运转，在遇到异常情况时能够做出令人接受的合理反应。

4.确定被测功能的使用频率：

使用频率高的功能发生意外的可能性相对要高一些，而且，这些功能中一些小问题所造成的影响可能会随着使用次数的增多而被扩大。对于这样的模块，测试的标准不能设置得太低，特别是核心功能中使用率最高的模块，一般情况下应该重点测试。

在本系统中，应当重点测试手语相关功能与健康服务功能。

5.确定被测功能是否是一个错误易发的功能

在执行测试的过程中，我们会发现某些模块在几轮测试中总是比其他模块出现的错误多。这些模块应该引起测试的注意，因为它们在以后的测试中仍然可能会出现很多错误，我们不能减少对它们的测试，如果这些模块属于核心功能或者是用户常用的功能，可能还需要增加测试用例以更多地发现隐藏缺陷。

对于那些错误少且错误数量已呈收敛趋势的模块，如果其本身功能未发生改变或者其他模块的修改对其不造成影响，我们不妨适当减少对这些模块的测试次数，例如：在后面的几轮回归测试中不对其进行测试，直至最后的回归测试。

6.确定客户能够容忍哪些错误存在：

从另一个角度看，客户对不同模块中发生的不同错误的接受程度是不一样的。针对客户能够接受和不能接受的错误，我们可以相应地确立测试的侧重点。比如：对功能要求高于界面要求的模块，可以加强功能测试，减少界面测试用例个数或者减少界面测试用例的执行次数。

7.发生异常情况的可能性：

用户输入错误数据的可能性有多大，他们不太可能输入什么样的错误数据?用户操作时可能会改变通常的执行顺序吗?为当前功能提供输入数据的其他模块的出错机率高吗?当一个功能需要多个模块共同协作才能完成时，这些共同协作的模块是否都足够稳定?当前功能并发操作的可能性是否高?系统运行的相关软硬件是否安全和健壮?有很多的因素我们需要考虑。与不太可能出现的错误情况相关的测试用例，我们是否可以试着将其执行优先级设置低一些?例如：在多种错误输入数据中，优先尝试最有可能出现的错误数据;如果使用者一般不会对同一数据进行操作，那么对并发操作可以暂时不进行测试。

8.确定错误所造成的影响：

不同模块出现的错误造成的影响是不同的，这些影响可能是数据丢失、系统的异常退出等等。例如：某个操作数据库的模块出现异常后，有可能对数据库造成死锁，阻碍其他模块的正常运行。对这样的模块，需要进行仔细的验证。

在本软件中，用户在使用手语互译功能时发生的错误被定义为影响最大的错误，其次是健康服务功能；相对来说错误影响不大或者几乎没有影响的功能为手语学习，健康咨询，界面操作等等。

1. 测试难点

主要为测试技术，本产品在测试过程中的测试手段可以分为两大类：

白盒测试：通过逻辑覆盖、控制流分析等方法检测实现产品功能的代码块是否按照设计规格说明书进行，每一条通路是否能够正常工作，各个逻辑结构是否合理，主要针对单元测试。

黑盒测试：不考虑内部结构，根据需求规格说明书设计测试用例，检查程序功能是否正确无误，主要对界面和软件功能进行测试。

由于本产品面向残障人群，所以更注重实际使用过程中的功能与界面测试，因此在测试过程中应该以黑盒测试为优先测试手段。

1. 测试深度

主要为测试方式，测试方式的定义有多种模式，每一种测试方式都会采用某种测试技术。如黑盒测试或者白盒测试。

测试方式按测试阶段层次可分为：

底层测试：单元测试。

接口层次：集成测试，完成系统内单元之间接口和单元集成为一个完整系统的测试。

系统测试：负载测试、压力测试。

用户层次：验收测试，验收用户是否真正所需要的产品特性，验收测试关注用户环境、用户数据，用户也需要参与测试。比如Alpha测试和Beta测试。

实际上所有的测试都可以按以下顺序进行实行；

单元测试 -> 集成测试 -> (功能测试) -> 系统测试 -> 验收测试

这套模式同样适用于本产品的测试，如下图。



1. 测试广度

考虑系统的各种覆盖率。

需求覆盖率是指测试对需求的覆盖程度，通常的做法是将每一条分解后的软件需求和对应的测试建立一对多的映射关系，最终目标是保证测试可以覆盖每个需求，以保证软件产品的质量。

路径覆盖率=至少被执行过一次的路径数/总的路径数。

集成测试结束时，需求覆盖率不少于60%，路径覆盖率不少于40%；系统测试结束时需求覆盖率达到100%，路径覆盖率至少75%；发布时同系统测试阶段结束是要求相同。

同时，系统应当提高代码覆盖率，提高代码重用性。

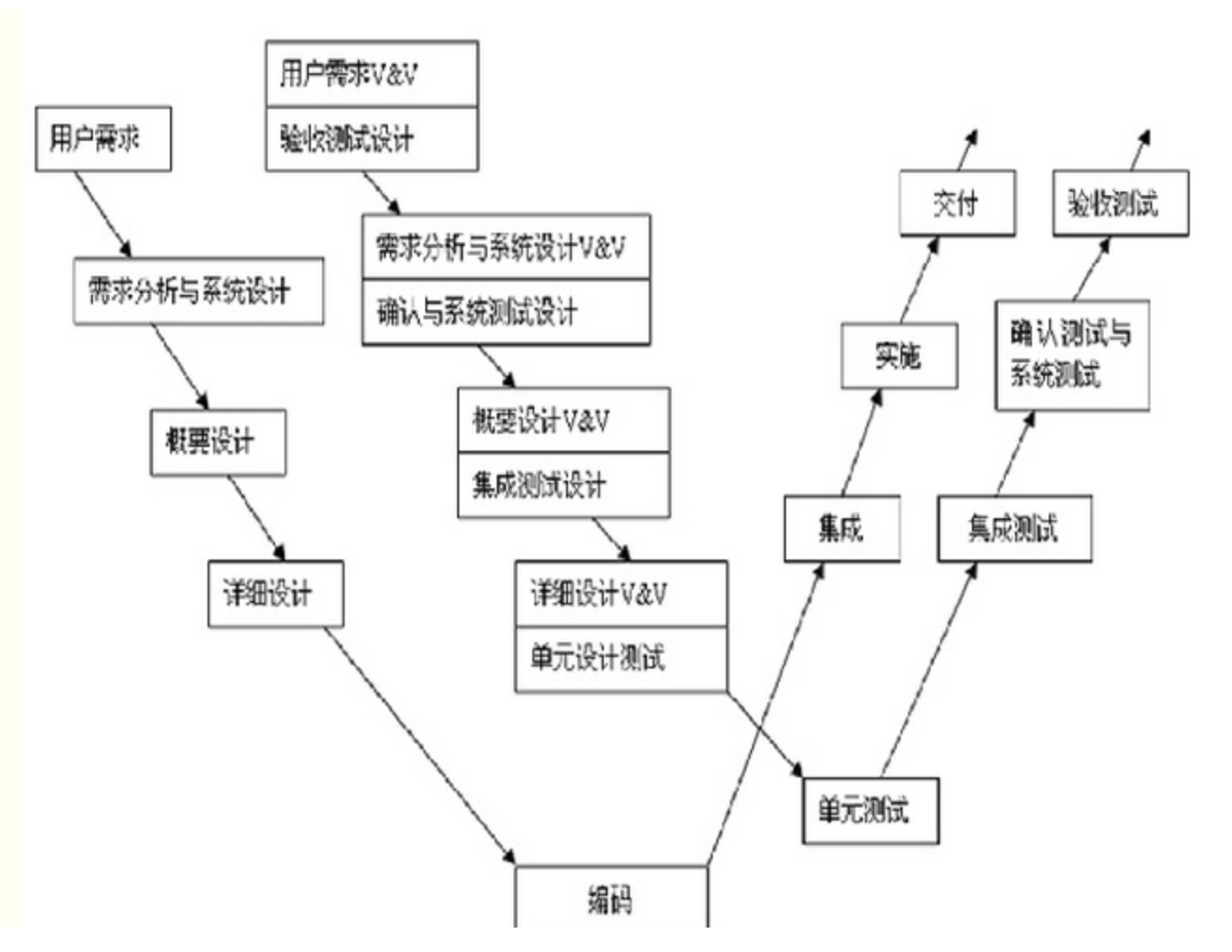
1. 测试活动安排

测试安排：W模型

W模型有两个V字模型组成，分别代表测试与开发过程。

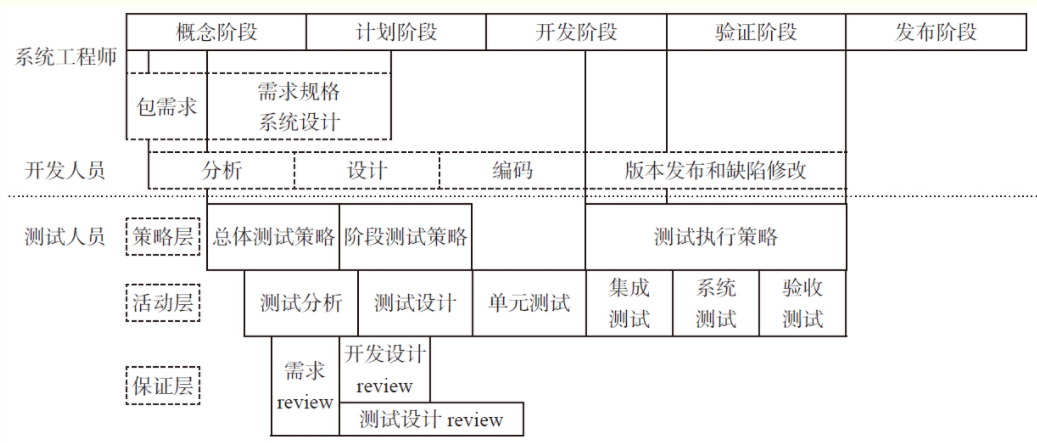
在这里，测试的对象就不仅仅是程序。需求和设等同样需要进行测试，测试和开发是一起进行的。

这有利于在早期发现问题，比如，需求分析完成以后，经过测试，我们就可以尽早的找出不合理或者错误的需求，对需求进行的测试，我们也可以在早期就了解项目情况，及早制定相应的应对计划，减少后期的测试工作时间，从而加快项目的整体进度。



测试计划与人员分工：

在软件开发的各个阶段都离不开测试活动，测试人员并非在代码实现之后开始测试，因此在各个阶段测试人员应与系统工程师和开发人员及时沟通配合，确保项目顺利进行。对于测试人员，将测试活动分为三个层次：策略层、活动层和保证层。策略层指挥测试该如何进行，确保测试做的是正确的事情；活动层负责具体的执行；保证层确保能够顺利地执行任务。



**4、测试充分性判断**

（1）功能性测试充分性判断标准

功能性测试标准：首先通过白盒测试测试实现产品功能的各个模块代码结构是否合理。

* 1. 手语学习功能

白盒测试：算法要求满足语句覆盖、条件组合覆盖、路径覆盖，通过测试用例。

* 1. 手语互译

白盒测试：算法要求满足语句覆盖、条件组合覆盖、路径覆盖，通过测试用例。

* 1. 健康资讯，健康服务

白盒测试：算法要求满足语句覆盖、条件组合覆盖、路径覆盖，通过测试用例。

其他功能略。

其次通过黑盒测试，根据需求规格说明书，利用等价类划分、边界值分析、因果图设计测试用例，检查程序功能是否正确无误，主要对界面和软件功能进行测试。

接近验收测试时，在测试过程中还需要考虑到用户的身份，以及不同身份用户可能会采取的动作，产品的反馈是否能达到用户的合理预期。例如针对老年人，不要采用过多的弹窗提示，因为老年人总是会把弹窗误认为是系统出故障。

要求所有测试用例通过率达到100%。

（2）性能测试充分性判断标准

系统性能指标：

响应时间：2s左右认为是最好的响应速度；5s左右认为是一般的响应速度；10s左右认为是差的响应速度。冷启动时间：不超过1.5s；热启动时间：不超过1s

错误率：错误率＝(失败交易数/交易总数)\*100%，事务成功率：高并发情况下，事务失败率小于5%则可以通过。

资源性能指标：

CPU占有率：空闲时间的消耗：CPU占有率应为0；在运行一些应用的情况下：CPU已占50%时，CPU占有率应小于30%；高负荷情况：下即CPU已占用80%以上时，CPU占有率应小于30%。

内存：空闲状态：切换至后台或启动后不操作；中强度状态：操作较长时间应用；强度状态：高强度使用应用。JVM内存使用率小于80%则通过。

磁盘吐吞量：一般情况下，磁盘繁忙率要低于70%。

网络吐吞量：一般情况下不能超过设备或链路最大传输能力的70%。

稳定性指标：通过72小时的压力测试。

要求非功能性测试用例通过率达到90％。

评价表格：

|  |  |
| --- | --- |
| **考核标准** | **得分** |
| （1）报告格式规范、语言通顺（10%）； |  |
| （2）质量标准分析合理（20%）； |  |
| （3）测试风险分析全面、合理（20%）； |  |
| （4）测试总体测试策略全面、合理（30%） |  |
| （5）测试充分性判断标准合理（20%） |  |