六、软件体系结构设计与评估

6.1 基于问卷调查的评估

6.1.1 基于问卷调查评估的定义

基于问卷调查的评估是通过设计与体系结构决策、文档内容、体系结构本身细节相关的调查问卷来对软件体系结构进行评估的一种方式。

6.1.2 本系统的评估调查问卷

本系统评估调查问卷由本寝室其他同学担任评估人员设计，同时成立调查小组，随机抽取本组4个组员对系统各项评估问题进行打分，具体结果如下表：

表4-1 本系统调查问卷

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 评估问题评估问题 | 组员1 | 组员2 | 组员3 | 平均分 |
| 软件功能实现情况？ | 10 | 9 | 10 | 9.67 |
| 软件界面情况？ | 10 | 9 | 10 | 9.67 |
| 软件可用性？ | 9 | 8 | 9 | 8.7 |
| 软件可测试性？ | 6 | 7 | 5 | 6 |
| 软件易用性？ | 9 | 9 | 8 | 8.7 |
| 软件可移植性？ | 2 | 4 | 3 | 3 |
| 软件可维护性？ | 8 | 9 | 9 | 8.7 |
| 代码可重用性？ | 9 | 9 | 8 | 8.7 |
| 文档完备性？ | 8 | 10 | 9 | 9 |
| 软件可管理性？ | 9 | 8 | 8 | 8.3 |

总结：由调查问卷可以得出，该项目存在一定的风险。本调查小组组员一致认为该项目中存在的风险是软件的可移植性较差，即只能在Windows平台使用该音乐管理系统，不能在其他平台正常访问，属于该软件设计过程中的缺陷。此外，软件的可测试性一项评分较低，原因是软件测试的简单性较差，需要特殊且复杂的软件测试软件，对软件测试人员水平要求较高。未来，本项目将努力提高软件可测试性的可操作性，降低软件在开发设计过程中的缺陷，提高测试效率。

6.1.3 基于问卷调查评估的优点

该评估方式自由灵活，可评估多种质量属性，也可以在软件体系结构的多个阶段进行。

6.1.4 基于问卷调查评估的缺点

由于不同评估人员对领域的熟悉程度、相关经验的认识程度不同，因此评估结果也会因为评估人员的不同而不同，评估结果受评估人员主观推断的影响程度较大，主观性较强。

6.2基于场景的评估

基于场景的方式由SEI首先提出并应用在体系结构权衡分析方法（ATAM）和软件体系结构分析方法（SAAM）中。

这种软件体系结构评估方式分析软件体系结构对场景也就是对系统的使用或修改活动的支持程度，从而判断该体系结构对这一场景所代表的质量需求的满足程度。这一评估方式考虑到了包括系统的开发人员、维护人员、最终用户、管理人员、测试人员等在内的所有与系统相关的人员对质量的要求。

6.2.1场景设计（8个）

通过评估小组的集中讨论，得出八种重要场景，并确定场景优先级为：①功能正确性②易用性③可重用性④可维护性⑤概念完整性⑥可定制性。

1. 可重用性

项目中许多功能需要被重复使用，譬如通过网络交换数据信息，读取和修改配置文件信息，控制音乐播放的各种操作。将这些功能集成封装在服务层的不同的类中，需要这些功能时引入某服务的依赖即可。在前端方面，设计出可重用的独立组件来展示歌曲信息，重用相同的组件和功能，但可以展示不同的歌曲信息。

|  |  |
| --- | --- |
| 场景元素 | 示例 |
| 刺激源 | 开发人员 |
| 刺激 | 使用系统中的歌曲播放模块 |
| 制品 | 代码 |
| 环境 | 设计时 |
| 响应 | 通过调用封装好的音乐播放服务实现对音乐播放的控制 |
| 响应度量 | 无需修改在系统各部分都可直接控制音乐的播放 |

1. 可维护性

项目开发维护过程中可能会遇到需要修改原有设计的情况，可维护性要求系统易于修改。在项目设计时使用开闭原则等设计方法来减少耦合，尽量避免代码修改。比如音乐播放器在设计网络服务类时先设计接口，然后用不同的实现类来拓展不同的功能。再比如音乐播放器的界面设计时可能会经常改动，但从界面中抽象出的数据和对数据的操作不变。于是就可以使用MVVM（Model-View-ViewModel）模式，将数据和对数据的操作写在ViewModel层中，界面写在View层中。这样使得界面的数据行为和具体样式解耦，对界面的修改不会导致业务逻辑代码的改变。

|  |  |
| --- | --- |
| 场景元素 | 示例 |
| 刺激源 | 开发人员 |
| 刺激 | 音乐播放器界面设计改变 |
| 制品 | 代码 |
| 环境 | 开发时 |
| 响应 | 界面的更改不影响逻辑控制代码 |
| 响应度量 | 开发人员修改的代码行数 |

1. 功能正确性

最终开发出的软件必须符合需求中要求的功能，在音乐播放器中就是使得音乐的播放控制、播放列表创建和管理、网络下载音乐和登录注册等功能被有效的实现。这是最重要和基础的质量属性。

|  |  |
| --- | --- |
| 场景元素 | 示例 |
| 刺激源 | 开发人员 |
| 刺激 | 开发音乐播放模块 |
| 制品 | 代码 |
| 环境 | 开发 |
| 响应 | 参照文档要求实现对应播放控制功能 |
| 响应度量 | 功能需求实现的比例 |

1. 概念完整性

在软件正常运行时，如果开发人员将该软件进行跨平台重构，或者更换不同歌曲源后需要按原方式播放时，如果和原设计冲突将导致难以重构的情况。所以歌曲数据的保存和处理应当遵循相同的数据结构，用统一的设计来进行数据库设计、前端界面设计和业务逻辑设计。当不同的开发人员设计相同的功能时就有了统一的参照，也便于沟通协作。

|  |  |
| --- | --- |
| 场景元素 | 示例 |
| 刺激源 | 开发人员 |
| 刺激 | 新成员加入 |
| 制品 | 代码 |
| 环境 | 在开发时 |
| 响应 | 新成员遵循统一的歌曲数据结构和编码规范进行开发 |
| 响应度量 | 新成员完成开发任务的时间 |

1. 易用性

易用性包含易理解性、易学习性和易操作性。当今市场上，存在许多音乐播放器，但大部分界面冗杂。本系统具有更加便捷的使用方式，让用户拥有更加简洁的界面，提高了易用性。当出现网络错误时有提示信息，文件异常时有提示信息并且把有问题的歌曲文件从播放列表中移除。在各个界面也增加了相应的文字说明，具有详细完善的平台用户使用说明手册，都提高了产品的易用性。

|  |  |
| --- | --- |
| 场景元素 | 示例 |
| 刺激源 | 用户 |
| 刺激 | 播放被删除的歌曲 |
| 制品 | 系统 |
| 环境 | 运行时 |
| 响应 | 将被删除的歌曲从播放列表中删除并及时给用户弹出提示信息 |
| 响应度量 | 提示信息在1秒钟内弹出 |

1. 可定制性

用户使用音乐播放器时，界面可以按照用户的喜好进行个性化修改。如更改背景颜色，背景图片，切换日间和夜间模式等。

|  |  |
| --- | --- |
| 场景元素 | 示例 |
| 刺激源 | 用户 |
| 刺激 | 按照喜好修改界面样式 |
| 制品 | 系统 |
| 环境 | 运行时 |
| 响应 | 界面样式按照用户选择的选项改变 |
| 响应度量 | 用户可修改样式的数量 |

6.2.2ATAM方法

ATAM是评价软件构架的一种综合全面的方法,它考虑了可修改性、可靠性和安全性等多种质量属性。这种方法不仅可以揭示出构架满足特定质量目标的情况,而且(因为它认识到了构架决策会影响多个质量属性)可以使我们更清楚地认识到质量目标之间的联系——即如何权衡诸多质量目标。

本文选择ATAM作为基于JavaWeb的校园活动购票平台软件构架的评估方法。之所以选择方法进行评估分析是因为校园活动购票系统既受到中央系统的约束,又有新的系统功能部分的要求,同时还要考虑降低维护成本等需求,可能会涉及到多方面的折中。

阶段一：体系结构表述及收集评估有关信息

步骤1：介绍ATAM方法

评估小组向参加会议的项目代表介绍了ATAM评估方法,说明了每个人将参与的过程,并回答了关于该方法及其结果和目标的问题。评估小组用一个标准的演示来简要描述的评估步骤和结果。

步骤2:介绍商业动机

在这一步中，项目的决策者(项目经理或校园活动购票平台的客户)从商业的角度介绍了系统的概况。首先介绍了校园活动购票平台所需要完成的任务，之后对这些任务的具体需求进行了介绍。部分学生要求对于该校园活动购票平台可能需要增加活动预约、在线选位以及在金明校区和明伦校区多增设一些线下取票机等，部分教师提出了本平台应增设教师活动专栏功能，方便活动购票。此外，他们还同时提出一些质量属性方面的要求，例如要尽量减少成本和维护所需的IT技术，以及对系统的反应时间要求等。

步骤3：介绍体系结构

软件构架师在这一步对音乐管理系统构架进行介绍，即对之前我们在其他章节所描述的构架进行了介绍。

另外，软件构架师还对其中所使用的构架风格和设计模式进行了介绍，其中包括面向对象方法、隐式调用、状态模式等设计思想。

步骤4：确定体系结构方法

然后,软件构架师对音乐管理系统构架的设计进一步地介绍。这一步基于前面对软件构架的整体介绍的基础上,构架师介绍了一些主要用于实现可修改性、可用性和性能等质量目标的构架方法,并对一些重要的设计和决策进行解释。

包括以下几个:

1)选择Windows 10作为系统的开发环境，Windows 10操作简单，提供出色的图形界面，有丰富的库函数可供选择。另外Windows 10支持组件技术，方便模块化编程.在Windows平台上有丰富的开发工具可供选择，这样在开发时，可以节省大量的开发时间。

2)软件开发工具应快速高效的开发出图形界面。应选用如WPF这个可重用性和可维护性高，功能完善的框架。

3)采用MySql数据库对服务端音乐和用户数据进行存储管理，由于MySql对数据暂存时间要求不高，因此使用MySql数据库就能满足要求，且MySql数据库基于Windows系统下的集成开发环境，开发速度快。

4)采用MVVM模式， 即将系统按不同的职责分成若干抽象程度不同的层次，使用MVVM Light框架作为MVVM框架，能够将数据和业务逻辑和页面的实现分离，提高可重用性和可维护性。也通过固定的分层模式规范了开发过程，增强了概念完整性。

阶段二：以体系结构为中心进行分析

步骤5：生成质量属性效用树

评估小组和项目决策者(包括软件构架师、项目经理和客户)针对该系统最重要的质量属性目标进行讨论.风险承担者开始强调安全性和可用性等，后来通过对以后可能遇到的新需求的探讨,风险承担者认为音乐管理系统的可维护性在以后的维护工作中也会很重要。经过这个讨论，得到一个系统质量属性效应树。

随后，通过会议的继续讨论，参加会议的代表们又对质量属性进行进一步求精，得出了可以分析的质量属性场景。然后对每个场景的重要性和实现难度用数值的方式来衡量，最后以累加和的大小来划分每个场景的讨论优先级。

步骤6：分析体系结构方法

至此，通过前面两步，已经得到了划分好优先级的具体质量需求(第5步)以及已经确定出来的构架方法(第3. 4步)。在这一步，与会者提出若干问题以澄清对这些构架方法的理解,并深入分析软件构架中的有风险决策、无风险决策、敏感点和权衡点。从优先级高的场景开始逐个进行讨论分析。

阶段三：以风险承担者为中心进行分析

步骤7：分析体系结构方法

在该步中,全体风险承担者按照类似于第一阶段第步的方式进行讨论。基于前面所得的场景,每个风险成员从自己的角度提出新得到场景。在这么个螺旋式的讨论过程中,可能会对原来产生的质量属性效应树进行增减。由于效应树已经比较全,这次集体讨论增加的场景只有几个。

步骤8：分析体系结构方法

重复步骤6即可

阶段四：提交最终评估报告

步骤9：给出评估结果

经过上面对质量属性效应树和场景的分析,评估小组将所得评估结果进行汇总后,对所有的风险承担者进行一个简单报告。其中得出了系统构架存在的敏感点和权衡点,很多部分都是和商业动机权衡的。因为在音乐管理系统构架的设计和决策中,选用了比较稳定和成熟的技术,因此在评估过程中没有发现较大的风险性的决策。

6.3小结

软件构架的分析与评估是一种控制项目风险的低成本手段,本编档针对基于音乐管理系统的特点,运用基于问卷调查的评估方式及基于ATAM的评估方式对音乐管理系统构架进行了分析和评估。