目录

1. 项目介绍.....2
1.1 项目规划.....2
1.2 软件主要功能.....2
1.3 主要限制.....2
1.4 编写目的.....2
1.5 项目可交付成果.....2
1.6 术语表.....3
1.7 参考文献.....4
2. 项目组织.....5
2.1 项目日程.....5
2.2 发布和重要时间节点.....5
2.3 项目预估图表.....6
2.4 项目成员/人员组织.....9
2.5 管理报告和沟通.....10
3. 跟踪与控制机制.....11
3.1 质量保证与控制.....11
3.2 改动管理与控制.....11
4. 方法与工具.....13
5. 风险管理.....15

1. 项目介绍

1.1 项目规划

本软件系统是为某地区历史学会刊物的编辑开发的在线出版系统。通过为文章的审查和出版流程提供自动化的工具，取代原有的人工方式，以最大化编辑的生产力和效率. 在最大化编辑的工作效率和产量的同时，这个系统会满足编辑的各种需要而且保持良好的易用性和可理解性.

1.2 软件主要功能

最终，编辑可以使用这个产品来管理由审核人员及作者组成的小组，并且同他们更高效地交流。同时，这个软件可以通过电子邮件促进作者，审核人员和编辑的交流。

1.3 主要限制

- 预算: ￥8000
- 时间: 两个月
- 人员: 5 名小组成员.

1.4 编写目的

本文档描述了网上出版系统的项目计划. 它同时也提供了对于项目的开销，工作和时间的估计。这些内容会用于控制和监督指导项目的执行. 这整个流程的目标就是保证项目能够在规定的预算和时间下完成.

1.5 项目可交付成果

所有下列的内容即为网上出版系统所要求的可交付的项目产品.

软件文档:

- 安装文档
- 终端用户文档

项目文档:

- 软件需求规格说明书(SRS)
- 软件设计说明书(SDS)
- 软件开发计划书(SPP)
- 软件测试计划书
- 软件测试报告

1.6 术语表

本系统所用术语如表 10-3 所示

术语	定义
作者	提交待审核文章的人. 针对多个作者的情况，这个概念特指主要作者, 即主要负责沟通的联系人。
数据库	这个系统下维护的所有信息。
编辑	收到文章，将文章送审，并最终决定文章是否出版的人。
历史学会数据库	已经存在的成员数据库。
成员	在历史学会数据库中列出的成员。
读者	任何访问网页阅读文章的人。
评论	关于文章是否适合出版的文本建议，可能包括改进意见。
审稿人	查验文章且拥有推荐文章出版或要求文章修改权力的人。

表 10-3 术语表

1.7 参考文献

IEEE Standard for Software Project Management Plans (IEEE 1058-1998).

2. 项目组织

2.1 项目日程

本节给出了项目任务和时间表的概述。这个项目将会采用瀑布式的开发方式来完成不同时期的任务。

2.2 发布和重要时间节点

发布和重要时间节点如表 10-4 所示

开发阶段	阶段完成时间	重要事件	发布时间
计划	04/07/16	完成项目开发计划书	04/07/2016
需求定义	04/12/16	完成软件需求规格说明书的拟定 完成软件设计说明书的拟定 完成需求规格说明书的终稿	04/10/16 04/11/16 04/12/16
设计(功能和系统)	04/22/16	拟定测试计划书 完成程序和数据库说明 设计说明书终稿完成	04/15/16 04/18/16 04/22/16

开发	05/12/16	完成前端和后台的开发 完成系统测试	05/10/16 05/12/16
集成和测试	05/23/16	完成最终版本的系统测试 完成测试报告 完成用户指南	05/14/16 05/17/16 05/23/16
安装和验收	05/31/16	完成维护计划 项目完成	05/25/16 06/02/16

2.3 项目预估图表

项目估算计划如表 10-5 所示，项目估算表的细节如图 10-1 所示，项目的 Gantt 图如图 10-2 所示

任务名称	时长	人员
分配项目资源	2 天	1
搭建项目环境	2 天	2
建立项目计划	2 天	1
定义和完善软件需求	2 天	1
编写需求规格说明书	1 天	1
进行结构测试	3 天	2
设计数据库	2 天	1
设计接口	3 天	1
编写软件设计说明书	1 天	3

编程	14 天	3
准备测试	2 天	2
执行测试	5 天	2
编写用户手册	1 天	2
重新进入另一个软件开发周期	8 天	4

10-5 项目预估图表

	任务模式	任务名称	工期	开始时间	完成时间	前置任务	资源名称	添加
1		**项目初始准备**	**4 个工作日**	**2016年4月1日**	**2016年4月6日**			
2		分配项目资源	2 个工作日	2016年4月1日	2016年4月4日		邓博洋	
3		搭建项目环境	2 个工作日	2016年4月5日	2016年4月6日		邓博洋, 寇宇增	
4		**项目计划**	**1 个工作**	**2016年4月7日**	**2016年4月7日**			
5		建立项目计划	1 个工作日	2016年4月7日	2016年4月7日		邓博洋	
6		**需求分析**	**3 个工作**	**2016年4月8日**	**2016年4月12日**			
7		定义和完善软件需求	2 个工作日	2016年4月8日	2016年4月11日		左宗源	
8		编写需求规格说明书	1 个工作日	2016年4月12日	2016年4月12日	7	左宗源	
9		**项目设计**	**8 个工作**	**2016年4月13日**	**2016年4月22日**			
10		进行结构测试	3 个工作日	2016年4月13日	2016年4月15日	8	蔡哲源, 邓博洋	
11		设计接口	2 个工作日	2016年4月18日	2016年4月19日	8	杨晨	
12		设计数据库	2 个工作日	2016年4月20日	2016年4月21日	8	邓博洋	
13		编写软件设计说明书	1 个工作日	2016年4月22日	2016年4月22日		蔡哲源, 杨晨, 寇宇增	
14		**项目实现**	**14 个工作日**	**2016年4月25日**	**2016年5月12日**	**10, 11, 12**		
15		编程	14 个工作日	2016年4月25日	2016年5月12日		蔡哲源, 杨晨, 邓博洋	
16		**软件测试**	**5 个工作**	**2016年5月13日**	**2016年5月19日**	**5, 8, 13**		
17		准备测试	2 个工作日	2016年5月13日	2016年5月16日		蔡哲源, 邓博洋	
18		执行测试	3 个工作日	2016年5月17日	2016年5月19日	17	蔡哲源, 寇宇增	
19		**完善安装说明**	**2 个工作日**	**2016年5月20日**	**2016年5月23日**			
20		编写用户手册	2 个工作日	2016年5月20日	2016年5月23日		邓博洋, 左宗源	
21		**软件维护**	**8 个工作**	**2016年5月24日**	**2016年6月2日**			
22		重新进入另一个软件开发周期	8 个工作日	2016年5月24日	2016年6月2日		蔡哲源, 邓博洋, 寇宇增, 杨晨, 左宗源	

图 10-1 项目预估图表细节

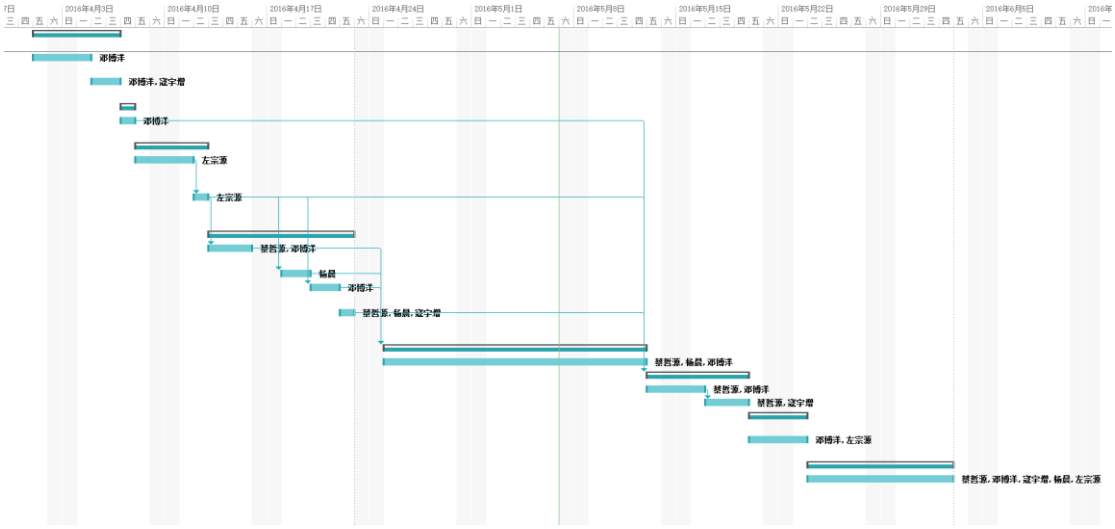

图 10-2 Gantt 图

2.4 项目成员/人员组织

项目经理

- 邓博洋

电子邮件：Joan@gmail.com

团队：

- 左宗源

电子邮件：Paul@hotmail.com

- 杨晨

电子邮件：BBB@gmail.com

- 蔡哲源

电子邮件：CCCC@gmail.com

- 寇宇增
- 电子邮件：KKKKK@gmail.com

2.5 管理报告和沟通

进展汇报和内部 / 外部的团队使用同样的交流机制。

会议

为了监控项目进展, 项目组会组织每周例会和一些根据需要组织的非正式的会议。 在会议期间, 团队成员会面对面讨论遇到的问题以及取得的进展，然后所有的结论和决定都会被记录到会议笔记中.非正式会议在整个项目的进程中都有可能召开，具体会根据团队成员及用户的需求来决定。

电子邮件

电子邮件将会作为日常的用来解决问题和疑惑的交流通道. 电子邮件还会被用于组织每周例会和传送文档。

状态报告

每周, 每个成员必须要填写一个当周的工作时长表. 工作时长表将整个项目分成了六个阶段。

在工作时长表的帮助下，项目经理和团队成员将能够实时获知项目的时间花费情况。

3.跟踪与控制机制

确认了用于项目跟踪和控制的技术。为了确保项目按照说明书的要求顺利进行，项目团队将要设置一些监督的策略。

3.1 质量保证与控制

- 进行严格的改动管理
- 以会议的形式进行质量审核
- 在实现设计前广泛地使用快速原型技术
- 与客户进行紧密的交流，每两周举行会议并保持邮件联络

3.2 改动管理与控制

- 对于会影响用户体验的改动，必须通知所有的客户。
- 对于不会影响用户体验的改动，仅需通知一名客户代表
- 根据团队大小不同，可能需要成立内部控制小组。当一名团队成员提出了改动，必须要得到其他两位成员的同意。
- 使用正式的版本编号。在一个新版本发布之前，版本的所有改动都必须在全体团队成员可读的公用文档中写明。版本号的构成如下：

<主版本发布号>.<次版本发布号><漏洞修补>

版本的改动需要审核。前一个版本和所有旧的文档或代码版本都应该保存下来。这可以确保当需要恢复之前版本时，所需的版本时可用的。

改动管理策略

当完成并核准一个产品时，所有的改动都会通过 Git 上传。产品经理将根据将进行的改动的风险预测和感知收益来审核这些改动是否被核准。由于项目采用了瀑布式

方法，对于改动的答复将被保守地处理，因为它们有可能对改动的下游活动产生极其巨大的破坏。

4. 方法与工具

开发方法

此项目使用瀑布式的软件开发方法来发布软件产品，并根据 IEEE 软件开发生命周期标准（IEEE 1074-1997）中一个特定版本的标准来组织工作。软件开发计划书（SPP）需要遵守 IEEE 软件项目管理方案规范（IEEE 1058-1998）。

开发工具

下面的工作分类将使用制定的开发工具来满足产品的需求：

文档发布：

- Microsoft Word 2012

项目管理：

- Microsoft Project 2012

具体实现：

- Microsoft Visual Studio 2012
- Microsoft SQL Server 2012
- Web Standards Update for Visual Studio & Jscript Editor Extensions

版本控制：

- Git 和 GitLab

5. 风险管理

下面这部分是对于这个项目团队可能遇到的所有风险的说明。

团队成员的离开：

团队成员可能因为个人原因而离开团队。

如何避免：确保每个成员了解项目的重要性并拥有同样的完成项目的决心。

减小风险：做好备份计划，以便如果真的有团队成员离开了，剩余团队仍然能够完成项目。在团队成员离开前，必须将所有的工作项目写成文档，以便让其他人来接手。

所开发的程序不可用：

最终的程序可能与客户的需求不相符，或者不能在指定的平台上运行。

如何避免：在写程序前仔细地进行系统的分析和设计。仔细地设计检测点。

减小风险：与讲师、团队成员和客户代表讨论，看看是否能通过改变设计或具体实现来解决。

不称职的项目经理：

不称职的项目经理可能导致整个项目的失败。

如何避免：项目经理需要保持对最新技术的了解和对项目开发的跟进。

减小风险：与讲师和团队成员讨论，尝试重新组织团队。

知识欠缺：

欠缺对项目的技术性和领域性的知识。

如何避免：所有团队成员事先都应当努力学习，客户应当提供学习资料或交流渠道给开发团队。

减小风险：向专家请求帮助。

开发时间问题：

剩余时间不足以完成项目。

如何避免：保持良好的时间管理与监督。

减小风险：与讲师和客户代表讨论，是否可能减小项目规模或延长开发时间。

过多的需求：

客户可能想要太多的项目功能。

如何避免：在写程序前仔细地进行系统的分析和设计。如果存在过多的需求，与讲师和客户代表讨论。

减小风险：与讲师和客户代表讨论。在下一个版本前暂时推迟这些需求的具体实现，或者减少需求量。

糟糕的设计：

包括糟糕的系统设计、用户界面设计以及数据处理流程设计。

如何避免：在最开始的时候正确地分析系统和设计；征询专家、讲师和客户代表的意见。保持对系统的审核与整合。

减小风险：征询专家、讲师和客户代表的意见。如果时间充裕，重新设计并重新进行代码实现。

缺乏与客户的互动：

项目团队可能因为缺乏与客户的交流与互动而得不到实时的反馈。这可能使得程序不可用或不合格。

如何避免：定期举行审核会议和状态报告，确保项目团队和客户之间的沟通渠道。

减小风险：定期拜访客户。

缺乏硬件资源：

项目团队可能没有足够开发项目的硬件资源，例如测试的机器。

如何避免：根据系统分析准备硬件。

减小风险：与客户和讲师讨论这个问题。

代码失窃：

开发过程中，代码可能被盗。

如何避免：对写好的代码实施权限管理，使用版本控制系统来管理代码。所有的代码都应该至少有一份备份。

减小风险：定期更新备份，并用防火墙保护好系统。