

# 测试计划

烫烫烫烫烫小组

文档名称：测试计划\_烫烫烫烫烫

## 文档描述

文件状态	文档类型	测试文档
<input type="checkbox"/> 草稿	文件标识	TestDoc
<input type="checkbox"/> 讨论稿	版 本	0.5
<input checked="" type="checkbox"/> 正式发布	作 者	王韬懿 安剑锋 张庭瑞 何林 张栋
	完成日期	2016. 4. 27

## 文档修订

版本	日期	更改人	描述(注明修改的条款或页)
0.1	2016.4.27	王韬懿	加入测试计划部分
0.2	2016.4.27	张栋	加入引言部分
0.3	2016.4.27	安剑锋	加入项目描述
0.4	2016.4.27	张庭瑞	加入测试内容以及方法
0.5	2016.4.27	何林	补充测试计划部分

# 目 录

一 前言.....	5
1.1 编写目的.....	5
1.2 项目背景.....	5
1.2.1 项目名称 .....	5
1.2.2 技术体系 .....	5
1.2.3 环境配置 .....	5
1.2.4 目标用户 .....	5
1.3 项目定义.....	5
1.4 参考资料.....	6
二 产品描述.....	6
2.1 系统描述.....	6
2.2 系统功能.....	7
2.3 系统性能.....	7
2.3.1 时间特性.....	7
2.3.2 适应性.....	7
三 测试计划.....	7
3.1 测试目标.....	7
3.2 人员分工.....	8
3.3 测试计划.....	9
3.4 测试阶段 .....	10
3.4.1 搭建测试环境.....	10
3.4.2 单元测试 .....	10
3.4.3 集成测试.....	11
3.4.4 系统测试.....	12
3.4.5 文档验证.....	12
3.4.6 测试评审 .....	13
3.5 测试规范性 .....	13
3.5.1 测试流程规范性 .....	13
3.5.2 测试计划规范性 .....	13
3.5.3 测试用例规范性 .....	13
3.5.4 缺陷管理规范 .....	14
3.6 缺陷定义 .....	14
3.7 风险管理 .....	14
3.7.1 风险管理步骤 .....	14
3.7.2 风险及应对策略 .....	15
四 测试内容及方法.....	15
4.1 测试方法 .....	15
4.2 测试内容 .....	15
4.2.1 个人中心模块 .....	16

4.2.2 数据分类模块 .....	16
4.2.3 回收站模块 .....	16
4.2.4 权限模块 .....	16
4.3 安全可靠测试 .....	16
4.4 易用性测试 .....	17
4.5 速度测试 .....	17
4.6 可扩展性测试 .....	17
4.7 回归测试 .....	17
4.8 兼容性测试 .....	17
4.9 数据库测试 .....	17
4.10 文档检查 .....	18
4.11 代码评审 .....	18

# 一 前言

## 1.1 编写目的

测试是软件工程中必不可少的一项，软件质量的保证必须通过测试实现，在实际工程中，测试本身就是一项耗时耗力的大工程，测试本身必须要有计划才能保证在最低的成本支出下实现最优配置，保证项目的质量。制定本项目计划主要出于以下几个方面的考虑：

1	为测试各项活动制定一个现实可行的、综合的计划，包括每个测试活动的对象、范围、方法、进度和预期结果
2	为项目实施建立一个组织模型，并定义测试项目中每个角色的责任和工作内容
3	开发有效的测试模型，尽量降低测试的难度和测试的风险
4	确定测试所需要的时间和资源，以保证其可获得性，有效性
5	确立每个测试阶段测试完成以及测试成功的标准、要实现的目标
6	识别出测试活动中各种风险，并消除可能存在的风险，降低由不可能存在的风险所带来的损失

## 1.2 项目背景

### 1.2.1 项目名称

内容管理系统

### 1.2.2 技术体系

Thinkphp 等。

### 1.2.3 环境配置

Phpstudy 环境。

### 1.2.4 目标用户

- (1)内容管理系统成员
- (2)任课教师单老师及其助教
- (3)其他人员

## 1.3 项目定义

序号	项目定义	含义
1	成本效率	效率产出与成本的比值

## 1.4 参考资料

序号	资料名称	作者	来源
1	内容管理系统说明文档	-----	项目内部资产
2	测试计划 5	Running Man 小组	项目内部资产
3	测试计划 6	易泳	项目内部资产
4	测试计划有关内容	-----	百度百科

# 二 产品描述

## 2.1 系统描述

本系统是由使用 thinkphp 框架开发的 onethink 管理系统的二次开发项目的一个分支小版本。内容是用来管理所有文档的地方，包含一些基本的操作：新增、编辑、禁用（启用）、删除、查询、排序等。

(1)内容分为三类：个人中心，分类数据，回收站。

(2)个人中心有我的文档、草稿箱和待审核。我的文档：当前登录用户发布的文档。草稿箱：查看和编辑草稿。待审核：列出所有未审核的文档，方便审核。

(3)分类数据（左侧分类的菜单，需要在系统->分类管理里添加分类才能显示）：所有的文档都会属于某个分类，所以我们用分类将文档进行归类、整理。文档有子文档的概念，用来区分文档之间的层级关系，就像树干与枝叶。合理地使用子文档，能够完成复杂的文档系统。点击文档标题后，会进入其子文档列表。在这里新增，则表示新增其子文档。且子文档还可以继续向下延伸，理论上无限的。这里有三个特别的按钮：移动、复制、粘贴。移动是将选中的一篇或多篇文档从一个分类（或父文档）移动到另一个分类（或父文档）下。复制的操作同上，但是原文档会保留。操作步骤：选中若干文档->点击移动（复制）->选择目标分类（父文档）->点击粘贴

(4)回收站：删除一篇文档，并不是真正的删除，而是将其放入回收站。清空回收站才是真删除。当然，也可以将文档还原。

## 2.2 系统功能

用户角色	功能名称	主要功能描述
管理员	用户信息管理	管理用户信息
	访问权限管理	控制管理员可以访问的节点。该页面中的节点，由“系统>菜单管理”创建时自动生成
	分类授权	控制管理员在后台“内容”下分类的访问权限，当管理员获得了某个分类的权限后，登陆后台时即可在“内容”中看到这些分类，并且拥有了对该分类下的文档的删除，修改，状态设定的权限。
	行为日志	对用户行为日志进行管理
	用户行为	对用户行为进行管理
	成员授权	用来向用户组加入和移除管理员
	前台管理	管理前台界面显示
普通用户	查看信息	登录查看相应信息
	修改密码	通过合法途径修改自己的登录密码
	内容管理	按照相应的授权对内容进行管理
	用户行为	查看用户行为
	行为日志	查看用户行为日志
	文件操作	具有对文件的增、删、改、查、回收站文件恢复等管理功能

## 2.3 系统性能

### 2.3.1 时间特性

为满足用户高效要求，数据的响应时间、更新处理时间、数据转换与传输时间、运行时间都应在 1-2 秒之内。管理员对用户权限的修改应在 1-2 秒内生效。

### 2.3.2 适应性

系统支持子文档的定义和修改，使得内容管理系统使用灵活方便，准确性好，能够满足用户添加各种文档的适应性。

## 三 测试计划

### 3.1 测试目标

有效的测试是软件系统质量保障中的重要一环,有效的测试本身就是一项耗时耗力的工作,通过对测试进行合理的计划和跟踪监控,可以保证在最低的时间、人力成本耗费下,实现最优的测试执行效果。尽量做到从用户、开发者、管理者的角度全面评测系统,尽可能多的发现并修正系统的错误、可能发生的故障等,保证系统的质量。

我们小组的待测项目为是一个基于ThinkPHP框架开发的内容管理系统，因此提出了以下测试目标：

- 1、发现系统中没有发现的 BUG;
- 2、指出系统中已经存在的 BUG 并且向开发者提出;
- 3、对系统的功能完善程度进行测试;
- 4、对系统的附属文档的完善性进行测试;

## 3.2 人员分工

Name	Start Date	End Date	Duration
王帽懿	4/26/16, 8:00 AM	5/25/16, 1:48 PM	> 1w 2d 4.75h
环境部署以及熟悉项目	4/26/16, 8:00 AM	4/26/16, 5:00 PM	1d
单元测试之分类授权测试	5/1/16, 8:00 AM	5/2/16, 5:00 PM	2d
集成测试	5/5/16, 8:00 AM	5/5/16, 5:00 PM	1d
系统测试	5/10/16, 8:00 AM	5/11/16, 5:00 PM	2d
文档验证	5/20/16, 8:00 AM	5/20/16, 5:00 PM	1d
测试评估	5/25/16, 8:00 AM	5/25/16, 1:48 PM	> 4.75h
张庭瑞	4/26/16, 8:00 AM	5/25/16, 1:48 PM	> 1w 4d 4.75h
环境部署以及熟悉项目	4/26/16, 8:00 AM	4/26/16, 5:00 PM	1d
单元测试之个人中心模块测试	5/1/16, 8:00 AM	5/2/16, 5:00 PM	2d
单元测试之成员授权测试	5/3/16, 8:00 AM	5/4/16, 5:00 PM	2d
集成测试	5/5/16, 8:00 AM	5/5/16, 5:00 PM	1d
系统测试	5/10/16, 8:00 AM	5/11/16, 5:00 PM	2d
文档验证	5/20/16, 8:00 AM	5/20/16, 5:00 PM	1d
测试评估	5/25/16, 8:00 AM	5/25/16, 1:48 PM	> 4.75h
何林	4/26/16, 8:00 AM	5/25/16, 1:48 PM	> 1w 4d 4.75h
环境部署以及熟悉项目	4/26/16, 8:00 AM	4/26/16, 5:00 PM	1d
单元测试之分类数据模块测试	5/1/16, 8:00 AM	5/2/16, 5:00 PM	2d



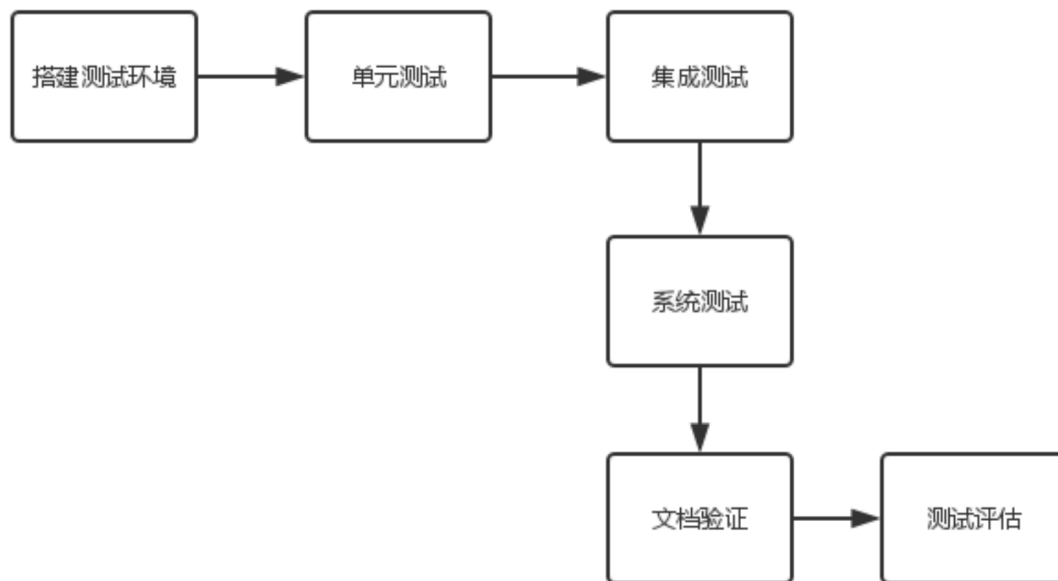
单元测试之行为日志测试	5/3/16, 8:00 AM	5/4/16, 5:00 PM	2d
集成测试	5/5/16, 8:00 AM	5/5/16, 5:00 PM	1d
系统测试	5/10/16, 8:00 AM	5/11/16, 5:00 PM	2d
文档验证	5/20/16, 8:00 AM	5/20/16, 5:00 PM	1d
测试评估	5/25/16, 8:00 AM	5/25/16, 1:48 PM	> 4.75h
张栋	4/26/16, 8:00 AM	5/25/16, 1:48 PM	> 1w 4d 4.75h
环境部署以及熟悉项目	4/26/16, 8:00 AM	4/26/16, 5:00 PM	1d
单元测试之回收站模块测试	5/1/16, 8:00 AM	5/2/16, 5:00 PM	2d
单元测试之用户行为模块测试	5/3/16, 8:00 AM	5/4/16, 5:00 PM	2d
集成测试	5/5/16, 8:00 AM	5/5/16, 5:00 PM	1d
系统测试	5/10/16, 8:00 AM	5/11/16, 5:00 PM	2d
文档验证	5/20/16, 8:00 AM	5/20/16, 5:00 PM	1d
测试评估	5/25/16, 8:00 AM	5/25/16, 1:48 PM	> 4.75h
安剑锋	4/26/16, 8:00 AM	5/25/16, 1:48 PM	> 1w 4d 4.75h
环境部署以及熟悉项目	4/26/16, 8:00 AM	4/26/16, 5:00 PM	1d
单元测试之访问授权测试	5/1/16, 8:00 AM	5/2/16, 5:00 PM	2d
单元测试之用户信息模块测试	5/3/16, 8:00 AM	5/4/16, 5:00 PM	2d
集成测试	5/5/16, 8:00 AM	5/5/16, 5:00 PM	1d
系统测试	5/10/16, 8:00 AM	5/11/16, 5:00 PM	2d

文档验证	5/20/16, 8:00 AM	5/20/16, 5:00 PM	1d
测试评估	5/25/16, 8:00 AM	5/25/16, 1:48 PM	> 4.75h

3.3 测试计划

Task	Start	End	Duration	Completed	Dependencies	Total Cost	Assigned	Planned Start	Start Variance	Constraint Start	Planned End	End Variance	Constraint End
0) test-plan	4/26/16, 8:00 AM	5/25/16, 1:48 PM	> 5w 4d 4.75h	0%									
1) 环境部署以及熟悉项目	4/26/16, 8:00 AM	4/26/16, 5:00 PM	1d	0%			王韬懿; 张庭瑞; 何林; 张栋; 安剑锋						
2) 单元测试之个人中心模块测试	5/1/16, 8:00 AM	5/2/16, 5:00 PM	2d	0%			张庭瑞			5/1/16, 8:00 AM			
3) 单元测试之分类数据模块测试	5/1/16, 8:00 AM	5/2/16, 5:00 PM	2d	0%			何林			5/1/16, 8:00 AM			
4) 单元测试之回收站模块测试	5/1/16, 8:00 AM	5/2/16, 5:00 PM	2d	0%			张栋			5/1/16, 8:00 AM			
5) 单元测试之访问授权测试	5/1/16, 8:00 AM	5/2/16, 5:00 PM	2d	0%			安剑锋			5/1/16, 8:00 AM			
6) 单元测试之分类授权测试	5/1/16, 8:00 AM	5/2/16, 5:00 PM	2d	0%			王韬懿			5/1/16, 8:00 AM			
7) 单元测试之成员授权测试	5/3/16, 8:00 AM	5/4/16, 5:00 PM	2d	0%			张庭瑞			5/3/16, 8:00 AM			
8) 单元测试之行为日志测试	5/3/16, 8:00 AM	5/4/16, 5:00 PM	2d	0%			何林			5/3/16, 8:00 AM			
9) 单元测试之用户行为模块测试	5/3/16, 8:00 AM	5/4/16, 5:00 PM	2d	0%			张栋			5/3/16, 8:00 AM			
10) 单元测试之用户信息模块测试	5/3/16, 8:00 AM	5/4/16, 5:00 PM	2d	0%			安剑锋			5/3/16, 8:00 AM			
11) 集成测试	5/5/16, 8:00 AM	5/5/16, 5:00 PM	1d	0%			王韬懿; 张庭瑞; 何林; 张栋; 安剑锋			5/5/16, 8:00 AM			
12) 系统测试	5/10/16, 8:00 AM	5/11/16, 5:00 PM	2d	0%			王韬懿; 张庭瑞; 何林; 张栋; 安剑锋			5/10/16, 8:00 AM			
13) 文档验证	5/20/16, 8:00 AM	5/20/16, 5:00 PM	1d	0%			王韬懿; 张庭瑞; 何林; 张栋; 安剑锋			5/20/16, 8:00 AM			
14) 测试评估	5/25/16, 8:00 AM	5/25/16, 1:48 PM	> 4.75h	0%			王韬懿; 张庭瑞; 何林; 张栋; 安剑锋			5/25/16, 8:00 AM			

### 3.4 测试阶段



#### 3.4.1 搭建测试环境

所测项目为一个 PHP 开发的项目，由于 PHP 天生对 Linux 友好，因此我们搭建了以下的测试环境：

环境	项目	要求
硬件环境	内存	>1G
	CPU	Intel I3 以上系列
	硬盘容量	>10G
软件环境	操作系统	Ubuntu/Mac os x
	PHP 运行环境	PHP 5.5.26
	服务器	Nginx 1.8.0
	数据库	Mysql 5.6.24

#### 3.4.2 单元测试

测试模块	待测功能	负责人
个人中心模块	我的文档	张庭瑞
	草稿箱	
	待审核	
分类数据模块	文档归类	何林
	子文档	
	文档新增	

	文档移动	
	文档复制	
	文档粘贴	
回收站模块	文档假删除	张栋
访问授权模块	管理员授权	安剑锋
分类授权模块	分类授权访问权限	王韬懿
成员授权模块	用户组和移除管理员功能	张庭瑞
行为日志模块	搜索系统的用户行为日志	何林
用户行为模块	行为的禁用与开启	张栋
用户信息模块	基本的新增、禁用（启用）、删除	安剑锋

### 3.4.3 集成测试

项目	描述
测试目标	使用通过单元测试的功能模块按照系统设计的原则构造完整的系统,并对其进行功能性和非功能性测试,测试模块 组装之间的关系以及可能出现的故障,及时解决尽量多的 潜在性运行错误。
重要程度	集成测试是紧随单元测试的测试活动,是整合性的测试,关 乎系统能否正常运行. 并且能够间接验证概要设计的可行 性。
测试方法	黑盒测试、动态测试
测试内容	对于已经通过单元测试的各个单元进行组装得到的系统进 行测试,包括单元之间的接口测试以及集成系统的功能测 试、数据库测试等。
完成标准	1.成功执行了测试计划中所有规定的集成测试 2.修正所有的发现的错误 3.测试结果通过了专门小组的专业评审 4.形成规范化测试报告

### 3.4.4 系统测试

项目	描述
测试目标	将已经通过集成测试的软件系统与计算机硬件、外围设备(如果有的话)、网络(如果有的话)等结合起来,对整个系统进行一系列严格的测试,验证软件系统是否符合需求 规格的定义,发现系统潜在的问题,保证系统的正常运行。
重要程度	是一个完整测试过程的最后阶段,面向的测试对象不仅仅是系统本身,还包括一系列相关的硬件、网络等必要环境。因而系统测试十分重要并且十分必要。
测试方法	黑盒测试
测试内容	对内容管理系统的各个功能模块集合起来进行系统测试
完成标准	1.成功执行了测试计划中所有规定的系统测试 2.指出了系统测试中的错误 3.测试结果通过了专门小组的专业评审 4.形成规范化测试报告

### 3.4.5 文档验证

项目	描述
测试目标	将内容管理系统的功能以及文档对应起来,测试出文档的不完整性,督促开发者完善项目文档。
重要程度	文档是一个项目的重要资产,对于非专业人士,文档起到了项目说明、介绍、功能描述等不可或缺的作用,因此文档的作用可见一斑。
测试方法	黑盒测试
测试内容	对项目提交的所有文档进行测试

完成标准	1.成功执行了测试计划中所有规定的文档测试 2.指出了系统中文档的错误与不足之处 3.测试结果通过了专门小组的专业评审 4.形成规范化测试报告
------	--

### 3.4.6 测试评审

测试评测方	评审权重	计算公式	说明
项目经理	30%	项目经理分数 *0.3	1.分值为 0-10 分; 2.在每次评审过程中每人仅有一次表决机会; 3.根据得出的分数决定评审结果; 4.分数 $\geq 7$ :通过; 否则评审未通过; 5.评审规则可依据项目实际进行相应的变更完善
开发方代表	40%	+ 开发方代表分 数 *0.4	
测试负责人	40%	+ 测试负责人分 数 *0.4	

## 3.5 测试规范性

### 3.5.1 测试流程规范性

任何阶段的测试都要遵循以下流程: 测试计划、测试设计、测试用例、执行测试、测试评审。

### 3.5.2 测试计划规范性

“5W1H”原则,明确测试内容和过程,对测试形成实质性的指导作用。

- (1) What 明确测试的范围和内容
- (2) Why 明确测试的目的
- (3) When 明确测试开始和结束日期
- (4) Where 明确给出测试文档和相关资料的位置
- (5) Who 明确测试人员的任务分配
- (6) How 明确指出测试的方法和测试工具

### 3.5.3 测试用例规范性

测试用例设计方法

- (1) 等价类划分法
- (2) 路径覆盖法
- (3) 边界值分析法

- (4) 流程分析法
- (5) 场景法
- (6) 其他设计方法

### 3.5.4 缺陷管理规范

本测试项目使用专门的缺陷管理工具----禅道

## 3.6 缺陷定义

软件缺陷有四种级别，分别为：致命的(Fatal)，严重的(Critical)，一般的(Major)，微小的(Minor)。

**A 类—致命的软件缺陷 (Fatal):** 造成系统或应用程序崩溃、死机、系统挂起，或造成数据丢失，主要功能完全丧失，导致本模块以及相关模块异常等问题。如代码错误，死循环，数据库发生死锁、与数据库连接错误或数据通讯错误，未考虑异常操作，功能错误等

**B 类—严重错误的软件缺陷 (critical)：** 系统的主要功能部分丧失、数据不能保存，系统的次要功能完全丧失。问题局限在本模块，导致模块功能失效或异常退出。如致命的错误声明，程序接口错误，数据库的表、业务规则、缺省值未加完整性等约束条件

**C 类—一般错误的软件缺陷 (major)：** 次要功能没有完全实现但不影响使用。如提示信息不太准确，或用户界面差，操作时间长，模块功能部分失效等，打印内容、格式错误，删除操作未给出提示，数据库表中有过多的空字段等

**D 类—较小错误的软件缺陷 (Minor)，** 使操作者不方便或遇到麻烦，但它不影响功能过的操作和执行，如错别字、界面不规范（字体大小不统一，文字排列不整齐，可输入区域和只读区域没有明显的区分标志），辅助说明描述不清楚

**E 类—建议 问题的软件缺陷 (Enhancemental)：** 由问题提出人对测试对象的改进意见或测试人员提出的建议、质疑。

## 3.7 风险管理

软件测试的风险是指软件测试过程出现的或潜在的困难或问题，软件测试风险的发生可能导致软件产品的测试不够充分或者测试结果的不准确。

软件测试的风险是不可避免的，它存在于整个软件测试过程之中，所以对软件测试风险的管理非常重要，必须尽力降低软件测试中所存在的风险，最大程度地保证质量和满足用户的需求。

### 3.7.1 风险管理步骤

风险管理，一般可以分成 5 个步骤

1. 风险识别：用系统化的方法来确定威胁项目计划的因素。就是要知道风险是什么，导致风险的原因，在什么地方最易出现风险以及风险显示出的特征。

2. 风险分析：量化风险。

3. 风险计划：风险标识、风险描述、风险影响程度、责任、资源、时间、活动、应对措施、结果、负责人。

4. 风险控制：主要方法有风险避免、风险弱化、风险转移和风险承担。
5. 风险跟踪：在风险受到控制以后, 要及时进行跟踪, 做好风险跟踪。

### 3.7.2 风险及应对策略

1. 时间进度风险
2. 产品认识风险
3. 人的风险
4. 质量目标风险
5. 测试环境和依赖风险
6. 测试充分性风险
7. 工具的风险
8. 变更管理

## 四 测试内容及方法

### 4.1 测试方法

本次测试方式以黑盒测试为主, 个别位置采用少量的白盒测试, 对项目的功能完成情况、流程规范以及其软件质量进行测试。

本次的测试对象为内容管理系统, 采用的测试技术为: 压力测试、功能测试、回归测试、易用性测试、性能测试

**压力测试:** 压力测试是对系统不断施加压力的测试, 是通过确定一个系统的瓶颈或者不能接收的性能点, 来获得系统能提供的最大服务级别的测试。本次测试主要测试该软件在大量的信息请求负荷下, 何时系统的的影响会退化或者失败。

**性能测试:** 性能测试是对软件性能的评价。简单的说, 软件性能衡量的是软件具有的响应及时度能力。因此, 性能测试是采用测试手段对软件的响应及时性进行评价的一种方式。根据软件的不同类型, 性能测试的侧重点也不同。

**功能测试:** 按照测试软件的各个功能划分进行有条理的测试, 在功能测试部分要保证测试项覆盖所有功能和各种功能条件组合。

**回归测试:** 回归测试是指修改了旧代码后, 重新进行测试以确认修改没有引入新的错误或导致其他代码产生错误。本次测试在使用其他方法发现软件的问题后, 尝试进入代码修改, 修改后重新进行测试。

**易用性测试:** 易用性测试: 易用性测试是指用户使用软件该功能时是否感觉方便。该软件针对的是广大消费者, 用户面广, 易用性是一个重要的测试点, 本次测试主要针对软件界面的链接方式、事件响应方式进行测试。

### 4.2 测试内容

内容管理系统的功能模块包括: 个人中心模块、数据分类模块、回收站模块和权限模块。本次的测试会涵盖所有模块。

本次测试分配如下:

### 4.2.1 个人中心模块

压力测试：由于同时在线人数的压力测试与服务器性能也有关系，因此个人中心模块的压力测试不做太多要求，只进行文件大小的压力测试。

性能测试：主要测试系统的交互速度，对每一个操作进行性能测试，根据系统反应的迅速程度进行评估。

功能测试：根据需求进行对系统所实现的功能的测试，评估标准为系统功能的完成度，测试功能包括我的文档、草稿箱、待审核。

### 4.2.2 数据分类模块

压力测试：对可分类数据的大小和数量以及子文档嵌套层数进行压力测试。

性能测试：主要测试系统的交互速度和分类速度，对每一个操作进行性能测试，根据系统反应的迅速程度进行评估，主要对移动、复制、粘贴这三个功能进行性能测试。

功能测试：根据需求进行对系统所实现的功能的测试，评估标准为系统功能的完成度，测试子文档有关功能和文件的移动、复制粘贴功能。

### 4.2.3 回收站模块

压力测试：对回收站容量进行压力测试。

性能测试：对回收站的删除速度和还原速度进行性能测试，监测交互时间。

功能测试：对回收站的删除功能和还原功能进行模拟测试。

### 4.2.4 权限模块

压力测试：由于管理员数量较少权限模块不做压力测试要求。

性能测试：无

功能测试：访问授权，分类授权，成员授权，用户行为，用户信息，测试以上功能的完成情况。

## 4.3 安全可靠测试

1. 用户信息安全：测试对用户信息的保密情况。
2. 用户权限限制：考察不同的用户权限限制情况。
3. 留痕功能：系统是否有操作日志,操作日志记录的操作情况的全面性和准确性,是否在发生非正常情况下可以进行恢复或查找到责任人信息。
4. 账号安全：对于登陆错误的提示信息，防止恶意操作如 SQL 注入等, 还有数据库的安全问题。
5. 异常情况处理：在程序运行过程中如果发生为定义操作（强制关闭）对于现场的保护以及数据恢复等。



## 4.4 易用性测试

1. 用户界面的友好性：用户界面做到布局合理、简单明了，使用户能够轻松理解界面模块内容，方便操作，功能布局与需求规格说明书规定的功能需求一致。
2. 易操作性：软件使用人员对于计算机技术的掌握情况不同，因此需要操作简便友好。
3. 界面风格的一致性：用户界面的风格和操作是否保持一致。
4. 错误提示信息：对于用户的错误操作是否有合理的提示，以及对程序运行错误的报告。

## 4.5 速度测试

1. 用户的注册速度。
2. 内容模块的基本的操作：新增、编辑、禁用（启用）、删除、查询、排序等操作的速度。
3. 回收站的删除速度和还原速度。
4. 分配权限的速度。
5. 用户行为的日志查询速度。
6. 用户信息的调用速度。

## 4.6 可扩展性测试

1. 权限管理的可扩展性：在新增功能后权限管理是否能够应对新增功能的权限管理。
2. 软件结构的可扩展性：是否能够灵活的支持新增功能

## 4.7 回归测试

对每一个阶段所作出的针对测试内容的修改是否达到了预期的水准进行测试评估，对于没有达到预期水准的部分需要重新修改。

## 4.8 兼容性测试

对于当前的主流浏览器进行兼容性测试，包括 IE 内核的浏览器，Google Chrome，Safari，Fire Fox 等。

## 4.9 数据库测试

对于数据库内部结构进行测试，数据库的设计范式是数据库设计所需要满足的规范，满足这些规范的数据库是简洁的、结构明晰的。此处检测数据库内每张表满足的范式。

## 4.10 文档检查

1. 规范性检查：用户文档描述是否规范，是否有版本控制和修改记录。
2. 完整性：用户文档是否齐全，内容是否完整。
3. 一致性：文档的描述和软件实际的功能是否基本一致（既不可超出，也不可少于）。
4. 正确性：文档是否存在会影响用户阅读的语法和字词错误。
5. 易理解程度：用户手册编写逻辑是否合理，是否图文相结合，对生僻词或自己使用的一些标记词语等是否有解释。

## 4.11 代码评审

静态测试技术通常在代码完成并无错误地通过编译或汇编后进行。代码评审（Code Review）也是一种有效的测试方法。我们提倡编写功能单一、结构清晰、接口简单的函数，因为过于复杂的函数会带来很多问题：加深其他开发人员理解代码的难度；不方便测试人员对其编写测试用例；容易隐藏错误；出现问题难以定位等等。

此处会采用一种代码评审工具，目前尚未确定。