|  |
| --- |
|  |
| **畅想器** |
| **[软件开发计划书]** |
|  |

目录

[目录 1](#_Toc13718)

[1．引言（Introduction） 3](#_Toc9289)

[1.1 目的（Purpose） 3](#_Toc22826)

[1.1.1软件的目标 3](#_Toc18766)

[1.1.2软件规范、方法和标准的选择 3](#_Toc22399)

[1.1.3软件工作量的估计 3](#_Toc12352)

[1.1.4软件进度表的制定 3](#_Toc32365)

[1.1.5软件风险的估计 3](#_Toc9035)

[1.1.6软件项目培训计划 3](#_Toc31857)

[1.2 范围（Scope） 4](#_Toc30698)

[1.2.1软件规模估计 4](#_Toc5681)

[1.2.2人力资源计划 4](#_Toc22778)

[1.2.3工作模块计划 4](#_Toc2225)

[1.2.4工作模块计划 4](#_Toc24200)

[1.3 术语定义（Terms Glossary） 4](#_Toc1666)

[1.4 参考资料（References） 5](#_Toc18633)

[1.5 版本更新记录（Version Updated Record） 5](#_Toc28480)

[2.项目概述（Project Summary） 5](#_Toc8228)

[2.1项目的目的（Project Purpose） 5](#_Toc7185)

[2.2项目的范围（Project Scope） 5](#_Toc18995)

[2.2.1主要功能点列表 5](#_Toc12546)

[2.2.2主要性能点列表 6](#_Toc25636)

[2.2.3主要接口 7](#_Toc8827)

[API类 7](#_Toc29934)

[ServerAPI类 7](#_Toc22303)

[2.3 项目的使用对象（Project Reader） 9](#_Toc24096)

[3．项目组织（Project Organization） 9](#_Toc20942)

[4.软件生存周期（Software Life Cycle） 10](#_Toc26725)

[4.1项目生存期框图 10](#_Toc27818)

[4.2项目生存期说明 10](#_Toc28913)

[4.2.1可行性研究： 10](#_Toc13642)

[4.2.2需求分析： 11](#_Toc10088)

[4.2.3软件设计： 11](#_Toc15233)

[4.2.4编码： 11](#_Toc2058)

[4.2.5软件测试： 11](#_Toc28516)

[4.2.6软件维护： 11](#_Toc23586)

[5．规范、方法和标准（Criterion，Means，Standard） 12](#_Toc20596)

[6. 任务与工作产品（Task and Work Products） 12](#_Toc6924)

[7.工作产品、任务规模、工作量估计（Estimates of Work Product，Task Size and Workload） 14](#_Toc32529)

[8．成本估计（Estimates of Costs） 15](#_Toc6220)

[9．关键计算机资源计划（Critical Computer Resource Plan） 15](#_Toc2729)

[10．软件项目进度计划（Software Project Schedule） 15](#_Toc3988)

[10.1软件项目每个阶段的进度 15](#_Toc27200)

[10.2设定的里程碑 16](#_Toc26017)

[11．风险分析（Risks Analysis） 16](#_Toc14910)

[11.1 风险来源 16](#_Toc27991)

[11.2 规避风险的方法 16](#_Toc29096)

[12．设备工具计划（Equipment and Tools Planning） 17](#_Toc14994)

[12.1所需的设备 17](#_Toc4657)

[12.2基本的要求 17](#_Toc6258)

[13．培训计划（Training Planning） 17](#_Toc12741)

[13.1 团队培训的目的 17](#_Toc10042)

[13.2 团队培训的目标 17](#_Toc29053)

[13.3团队培训时间 17](#_Toc1531)

[13.4团队培训计划 18](#_Toc21789)

[13.5 团队项目技术交流计划 18](#_Toc20116)

# 1．引言（Introduction）

## 1.1 目的（Purpose）

### 1.1.1软件的目标

目前，即时通信软件以国内的QQ和国外的MSN为代表，都注重的是聊天功能的体验增强，虽有一些小的工具，但功能不强。本软件目的是建立一套统一的多人工具集，便于扩展和增强，使得多人联网的应用开发变得简单易行。

### 1.1.2软件规范、方法和标准的选择

本软件采取插件式系统，便于后期扩展，定义了统一的插件开发模式，信息通信模式，定义了通信握手接口。基于本平台开发多人联网应用，简单易用，软件规模可大可小，重用性好，接口统一。

### 1.1.3软件工作量的估计

软件工作量一部分在于多人联网平台的搭建上，网络通信、服务器和主客户端界面工程量大，项目复杂。另一部分工作量在各个插件上，插件本身结构精简，可重用性好，但结构设计不易，实现难度大，数量多。

### 1.1.4软件进度表的制定

根据组员的水平合理安排项目的进度，使用project 2010绘制甘特图。

### 1.1.5软件风险的估计

软件风险主要来源于四个：

一、软件的工期，由于时间很紧，并且项目的规模较大，因此存在无法按时完工的风险性；

二、软件的可伸缩性，由于硬件的飞速发展和软件开发周期较长的矛盾，软件的升级和移植可能会非常困难，存在着软件的生命周期很短的可能性；

三、平台稳定性差，异常多，使得用户体验下降；

四、各个插件功能不足，数量少，无法满足用户需求。

### 1.1.6软件项目培训计划

针对团队内成员技术水平参差不齐的现状，计划每周进行一次技术交流，每两周进行一次技术讲座，对技术资料进行整理，改进团队整体的技术水平。

## 1.2 范围（Scope）

### 1.2.1软件规模估计

我们的软件由一个主登陆服务器，一个分布式消息传递服务器，一个主客户端，多个插件，多个插件关联的应用组成，是一个庞大的软件系统。

主客户端由通信模块、界面显示模块、插件系统、好友群组管理器、消息管理器，五大部分组成。

服务器由数据库通信模块、请求响应模块、在线用户数据管理器，三大部分组成。

分布式消息传递服务器由数据库通信模块、请求响应模块、数据互联通信模块，三部分组成。

插件由主启动类，插件UI，插件逻辑等部分组成，具体结构由具体插件决定。

### 1.2.2人力资源计划

软件开发由一个编码组，一个测试组，一个维护组组成，每个组的成员有交集，编码组负责代码的更新维护，而测试组则负责进行定期和不定期的测试，给编码组反馈测试结果和bug列表。维护组负责维护和更新文档，组内技术资料，召开定期会议。

### 1.2.3工作模块计划

采取各基础类库分别开发的方式，降低系统耦合性，大量整合开源应用库，维护成熟的语音编码压缩库，音频处理库，浏览器渲染库，Markdown格式库，WPF控件库，WPF主题库。

本软件采取原型开发的方式，采取以周为单位的迭代方式，以现有原型进行扩充，改进，然后进行代码测试，修复bug，由维护组负责更新文档，整理资料。

### 1.2.4工作模块计划

5-10周为初级原型开发期，11-13周为系统扩展插件开发期，14-15周为系统整体测评期。

## 1.3 术语定义（Terms Glossary）

[1] 主服务器：专指用户登录信息维护服务器

[2] 消息服务器：专指用户数据通信传递分布式服务器

[3] 插件系统：指本软件中的基于.net可组合框架MEF开发的多人联网插件系统

## 1.4 参考资料（References）

[1]谭强. 汉字手写笔迹的实时墨水仿真算法研究. 哈尔滨工业大学. 2008

[2]唐家德.基于MATLAB的非线性曲线拟合. 计算机与现代化，2008,(6)

[3]于东.基于笔触特征三角形的手写汉字书法效果美化方法. 计算机科学,2013,(2)

[4]刘海香.散乱数据点曲线拟合的研究及二次曲线拟合的一种新方法. 山东大学. 2005

[5]方逵,吴涛. 书法艺术的计算机模拟及其实现. 模糊系统与数学. 1996,10(1):71~74

[6]黄树东. 数字笔迹书法风格生成技术研究及实现. 华南理工大学. 2013

[7]王钲旋. 一个笔划填充算法及其在计算机书法中的应用. 计算机辅助设计与图形学报，1994,(3)1.5

## 1.5 版本更新记录（Version Updated Record）

版本更新见代码托管平台：

https://sunxiaofan.visualstudio.com/DefaultCollection/DrawTogether/\_versionControl/changesets

# 2.项目概述（Project Summary）

## 2.1项目的目的（Project Purpose）

目前的在线交流软件主要的交流方式就是文本对话，以及在线语音。这些软件并没有实现除了文本，语音之外的其他辅助交流方式。本项目就是想在原有基础上加上一个共有的画布，在同一个群组里的用户将共同完成对同一张图像的绘制。这样它可以集实用性和娱乐性于一身，在用户之间交流的同时，可以画图辅助说明，而且其他人也可以随时添上一笔，使得交流更加无缝契合。而一起画图，也仅仅是其中一个功能，此项目的功能是可定制的，可以应会场要求而做相应调整，以及相关功能的添加。

我们想通过畅想器联网软件实现多人联网绘图、语音通讯、联网自动化办公、多人产品设计、网络会议、网络教学等多人工作应用。

## 2.2项目的范围（Project Scope）

### 2.2.1主要功能点列表

|  |  |  |
| --- | --- | --- |
| **模块编号** | **名称** | **模块功能描述** |
| SS1-1 | 用户注册 | 游客选择注册，系统返回注册页面，游客输入用户名密码，用户完善其他信息，用户提交注册信息，注册成功，成为用户。 |
| SS1-2 | 用户登录 | 用户输入用户名和密码，登陆系统。 |
| SS1-3 | 修改密码 | 用户将个人密码丢失后，用户选择找回密码，系统返回修改密码页面，用户输入新密码，系统进行验证应用新密码。 |
| SS1-4 | 修改个人信息 | 用户登陆后选择修改信息，系统返回信息修改界面，用户修改信息并提交，系统进行验证，系统保存用户新信息。 |
| SS1-5 | 用户注销 | 用户离开系统时，进行注销。 |
| SS2-1 | 查找好友 | 用户选择查找，输入要查找的用户名，系统匹配并显示用户名及其相关信息。 |
| SS2-2 | 添加好友 | 用户在查找界面选择添加好友，系统将好友关系存入数据库，系统提示添加成功。 |
| SS2-3 | 删除好友 | 用户选中删除好友，数据库删除对应好友关系，并将删除的用户名从好友列表中清除，系统提示删除成功。 |
| SS3-1 | 发送消息 | 用户输入聊天内容，点击发送按钮，系统将聊天内容发送到对方聊天窗口 |
| SS3-2 | 创建群组 | 用户选择创建群组，选择要添加到群组中的好友，系统提示添加成功，可以开始群组聊天。 |
| SS3-3 | 共享文件 | 用户选择共享文件，选择要共享的文件夹，系统提示共享成功，群组中的其他好友可以从中取出文件。 |
| SS4-1 | 用画笔绘图 | 用户调整想使用的画笔颜色和宽度，在面板上绘出图形。 |

### 2.2.2主要性能点列表

|  |  |  |
| --- | --- | --- |
| **编号** | **性能名称** | **性能描述** |
| 1 | 查询相应的用户名 | 在数据库中查找相应的用户信息 |
| 2 | 密码，信息的修改 | 在数据库中修改相应信息 |
| 3 | 聊天记录查询 | 在数据库中查找相应聊天记录 |

### 2.2.3主要接口

### **API类**

API类是整个软件的核心控制类，内有一些控制主应用程序的函数接口，用于控制主应用程序的行为，但由于应用程序的行为并不多，所以API类不大。

/// 得到主界面上的你可以控制的WPF节点对象，str是对象的名字，obj是你想添加的WPF节点

public static void AddUserWindowChild(string str,UIElement obj)

/// 用来获取自己的数据

public static Friend Me

///用来实现登陆功能

public static bool Login(string username,string password)

///测试用登陆，用于非联网

public static void TestLogin()

/// 打开一个打开文件对话框，用于打开图片

public static string OpenImageFile()

### ServerAPI类

ServerAPI类是软件联网通信的核心接口类，都是静态函数，且每个函数都是线程同步的，内部还有ServerAPI初始化的函数。

这些函数都是远程调用的形式，主要用于通知服务器和从服务器获取数据。使用时建议新开线程，采取异步的方式调用。

///初始化ip和端口号，构建客户端对象

public static void InitServerAPI()

/// 注册

///发送格式：UserMessage(1, RegisterData(User(1, null, username, nickname),password);

/// 接收格式：int，注册成功返回用户id，否则返回-1

public static int Register(string username,string password,string nickname)

/// 登录

/// 发送格式：UserMessage(2, LoginData(username, password))

/// 接收格式：LoginReturn,要求LoginReturn各成员变量不为null

public static LoginReturn Login(string username, string password)

/// 确认在线操作，并获取自己在服务器端消息盒子里的新消息

/// 发送格式：UserMessage(3, id)

/// 接收格式：List&lt;UserMessage&gt;

public static List<UserMessage> Checking(int id)

/// 邀请好友

/// 发送格式：UserMessage(4, List<int>); List第一项为插件id，第二项为邀请人id，接下来的都是被邀请人id

/// 接收格式：bool 消息成功发送true，消息发送失败false

public static bool InviteFriends(int plugin\_id,int my\_id,SortedSet<int> id\_list)

/// 测试名字是否可用

/// 发送格式：UserMessage(5, name)

/// 接收格式：bool 是true ，否false

public static int TestName(string name)

/// 根据名字查找好友

/// 发送格式：UserMessage(6,name);

/// 接收格式：List<User>;

public static List<User> SearchFriends(string name)

/// 加好友（请求人id, 被请求人id）

public static bool AddFriend(int my\_id ,int f\_id)

/// 更新我自身的数据，向服务器提交我的最新数据

/// 发送格式：UserMessage(8, User);

/// 接收格式：是否更改成功，true or false

public static bool UpdataMyInfo(Friend data)

public static bool UpdataMyInfo(User data)

/// 更新Password（用户id，旧密码，新密码）

public static int UpdataMyPassword(int id,string oldp,string newp)

/// 根据id获取指定人的全部数据

public static User GetFriendData(int id)

/// 获取指定群组的信息

public static GroupData GetGroupData(int id)

/// 获取一组人的数据，是上面两个方法的整合

public static ReturnData GetGroupData(List<int> friend\_list,List<int> group\_list)

/// 回应一个增加好友的请求

/// 发送格式：UserMessage(13, int[3]{请求人id,被请求人id,0或1(拒绝与否)});

public static bool AgreeAddFriend(int my\_id,int friend\_id,int attitude)

/// 确认某条通知类信息已经处理过了

public static bool confirmMessage(int user\_id,int msg\_id)

## 2.3 项目的使用对象（Project Reader）

畅想器的使用者主要为有在线交流需求的普通用户和有联网工作需求的工作者，对画图软件要求高的人员不建议使用。本软件简单易用，对使用者的教育水平和技术水平要求不高，只需有基本的计算机操作常识即可使用。所需要的硬件设施为可以连接网络装有畅想器的计算机。

系统维护人员为项目开发团队， 对于此系统的相关部分比较熟悉，团队内具有数据 对于此系统的相关部分比较熟悉，团队内具有数据 库、计算机网络较为熟悉的人员，维护难度不是很大。

管理人员为开发团队制定的人选，负责软件日常的管理和监管。

软件的使用频率取决于工作者的工作量和工作需要。

# 3．项目组织（Project Organization）



# 4.软件生存周期（Software Life Cycle）

## 4.1项目生存期框图

可行性研究

需求分析

软件设计

编码

软件测试

软件维护

## 4.2项目生存期说明

### 4.2.1可行性研究：

首先，我们开发小组明确了所要开发项目的目标，即在联网状态下，实现多人之间的状态交互的一个平台，包含画图、文件分享、实时聊天等。同时，我们明确了以原型模型为开发模型，计划六人采用一个月的时间完成初期开发。此外，我们还对项目开发在技术、经济、操作、市场前景等方面做了评估。

### 4.2.2需求分析：

在项目开发初期，我们开发小组的目的是实现在联网状态下多人的UI共同设计，也就是多人共同完成画图操作。但随即我们发现我们的功能可实现扩展，因此将我们的目标系统的功能确定在一个功能集上，即实现多人间的画图操作、文件分享、实时聊天等多项功能。

### 4.2.3软件设计：

设计分为概要设计和详细设计。关于概要设计，由于旨在建立系统的总体结构，DrawBitmap是整个系统的核心工程；Server是核心的服务器端；MuticastNetwork是基础通信类；Plugin，UserApplication则实现所有的插件工程以及所有独立的应用工程。详细设计则是在结构总体上加以细分，为后续的编码工作提供最直接的依据。

### 4.2.4编码：

我们小组用VS进行WPF项目开发，主要分成UI组和内部功能实现组进行同步编码，同时，为了确保代码的质量和代码的风格，我们开发小组会指定一系列的开发标准。

### 4.2.5软件测试：

为了保证软件的质量，我们开发小组专门设置了软件测试小组，分别进行单元测试、集成测试。由于我们的项目仍然处于后期开发状态，因此我们打算在整个项目完成后再重点进行系统测试和验收测试。

### 4.2.6软件维护：

我们小组的开发人员会在项目完成后，根据用户需求、使用中发现的各种错误以及市场环境的变化进行版本升级以及功能的维护更新。

# 5．规范、方法和标准（Criterion，Means，Standard）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **方式** | **类型** | **名称** | **发送数据格式** | **回应数据格式** |
| 客户端发给服务器 | 1 | 注册 | UserMessage(1,RegisterData(User,password)); | 成功则返回id,否则-1 |
| 2 | 登陆 | UserMessage(2,LoginData(username,password)); | LoginReturn(类的各个成员) |
| 3 | 获取消息 | UserMessage(3, id); | List<UserMessage> |
| 4 | 邀请好友使用插件 | UserMessage(4, List<int>); | Bool,成true败false |
| 5 | 注册用户名校验 | UserMessage(5, name) | Bool可用则true |
| 6 | 按名字搜索好友 | UserMessage(6,name); | List<User> |
| 7 | 添加好友 | UserMessage(7,AddFriendData(invitor\_id, asked\_id)); | Bool,成true败false |
| Etc… | Etc… | Etc… | Etc… |
| 服务器发给客户端 | 1 | 上线 | Id | 无 |
| 2 | 下线 | Id | 无 |
| 3 | 收到好友邀请插件 | int[itype,id] | 无 |
| 4 | 收到好友添加请求 | User | 无 |
| 5 | 更新资料 | User | 无 |
| 6 | 你请求加的人表态了 | List{User(被请求人), bool(同意或者拒绝)} | 无 |
| Etc… | Etc… | Etc… | Etc… |

# 6. 任务与工作产品（Task and Work Products）

项目进度采用Project 2010进行设计，计划安排如图6.1.1、6.1.2



图6.1.1



图6.1.2

计划甘特图如图6.2.1、图6.2.2和图6.2.3所示。

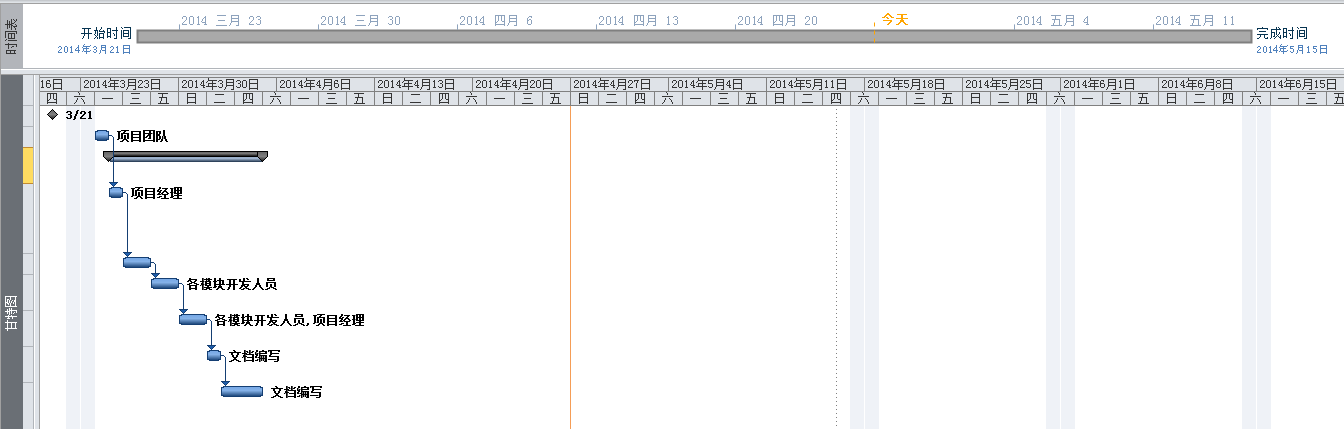


图6.2.1

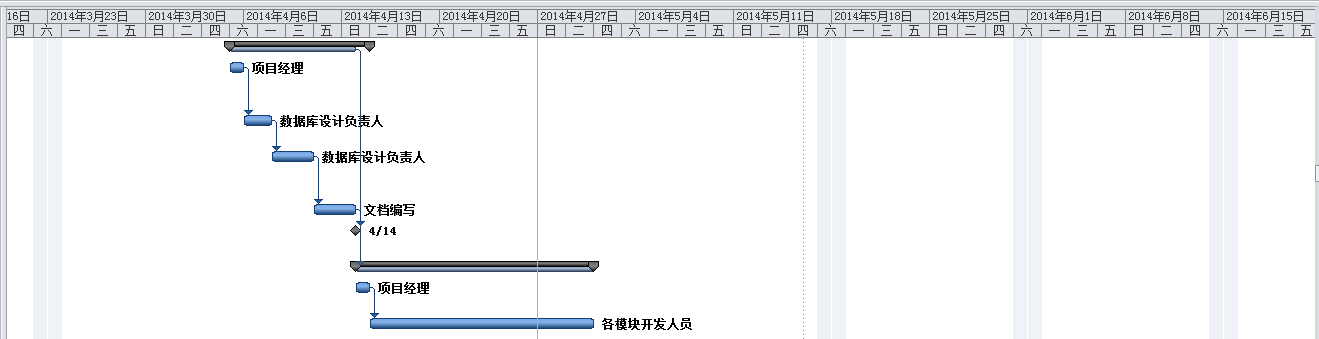


图6.2.2

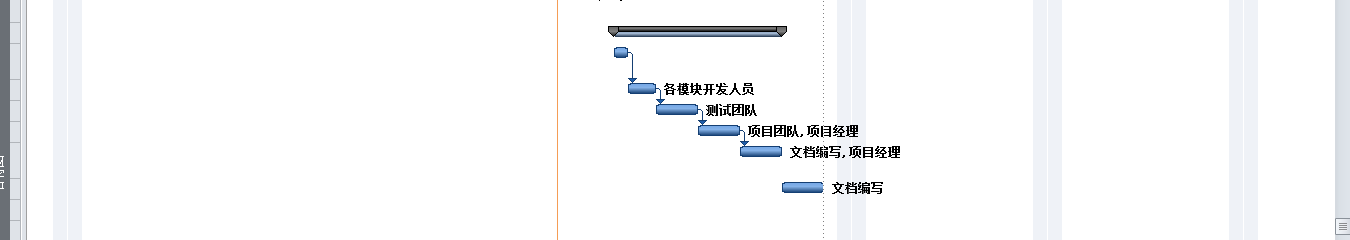


图6.2.3

# 7.工作产品、任务规模、工作量估计（Estimates of Work Product，Task Size and Workload）

|  |  |  |
| --- | --- | --- |
| **项目规模估算** | | |
| 阶段/任务 | 估计（人时） | 估计假设 |
| 阶段A/可行性研究 | 12 | 这是为后续的软件开发做必要的准备工作，初步打算是开一次六人全体会议，时长两小时 |
| 阶段B/需求分析 | 12 | 为了明确目标系统需要做什么，以小组例会进行开发讨论 |
| 阶段C/软件设计 | 20 | 因为涉及到软件开发的指定方案，因此主要需要编码人员进行讨论分工，研究出软件的总体结构、接口和全局数据等 |
| 阶段D/编码 | 240 | 项目工程既需要实现基础通信，也需要实现各种平台应用，规模比较大，所需的代码量也大 |
| 阶段E/软件测试 | 30 | 由专门的测试人员按照测试点的不同进行多项多次测试 |
| 阶段F/软件维护 | 60 | 由于在初期开发的时候我们定制了比较高的标准，使得我们的项目质量较高，因此预期后续的软件维护工作量会相对较小 |

# 8．成本估计（Estimates of Costs）

成本估计，是指对项目完成过程中耗费的人力、物力、财力资源的估算。成本估计应按类别进行估算，可能的成本估算类别

成本主要由时间成本和管理成本组成。

时间成本为团队成员从第五周开始到第十五周，共十一周的开发时间。

管理成本则为管理组成员负责召开会议和场地成本、队内协作成本。

# 9．关键计算机资源计划（Critical Computer Resource Plan）

项目使用.net4.0平台，采用Visual Studio开发

测试环境1为Win7系统64位，CPU-i5，内存4GDDR3，硬盘500G7200转

测试环境2为Win8系统64位，CPU-i5，内存8GDDR3，硬盘500G5400转

最低配置要求，Windows Vista及以上版本，安装.net4.0平台

建议配置，Windows7及以上版本，安装.net4.0平台

# 10．软件项目进度计划（Software Project Schedule）

## 10.1软件项目每个阶段的进度

项目分为三阶段5-10周为初级原型开发期，11-13周为系统扩展插件开发期，14-15周为系统整体测评期。

原型开发期，主要目标是建立可以使用的软件通信模型，实现插件模型的核心功能，建立并测试基础库，实现通信接口规范等。

系统插件开发期，着重开发各插件的功能，对各个功能点逐个攻破，有重点的开发实用插件。

系统整体测评期，进一步完善软件整体架构，必要时对个别类进行重构，增强扩展性和稳定性，对代码进行单元测试和压力测试，对服务器架构进行优化。

## 10.2设定的里程碑

1. 完成软件的主启动框架，实现和服务端的通信
2. 完成软件的多人联网绘图功能
3. 完成多人聊天功能并搭建聊天服务器
4. 完成文件分享功能
5. 软件稳定运行时间达到10小时以上

# 11．风险分析（Risks Analysis）

## 11.1 风险来源

软件风险主要来源于四个：

一、软件的工期，由于时间很紧，并且项目的规模较大，因此存在无法按时完工的风险性；

二、软件的可伸缩性，由于硬件的飞速发展和软件开发周期较长的矛盾，软件的升级和移植可能会非常困难，存在着软件的生命周期很短的可能性；

三、平台稳定性差，异常多，使得用户体验下降；

四、各个插件功能不足，数量少，无法满足用户需求。

## 11.2 规避风险的方法

规避风险的主要方法是:

一、在可行性研究及软件设计时进行系统的任务分工，另外采取小组例会的沟通方式，使得开发进程严格按照计划走；

二、在开发的过程中尽量使得代码的质量高，可移植性好；

三、单元测试提供平台稳定性，对每一个平台细节都应认真处理问题；

四、插件开发应借助他人的力量，采取开放接口，开放API的方式，使得对本平台感兴趣的人加入到插件制作中，增加平台的用户群和开发人数，当然前提是将平台稳定性做好，性能优异，用户体验好。

# 12．设备工具计划（Equipment and Tools Planning）

## 12.1所需的设备

设备：6台符合配置要求的电脑

工具：Visual Studio 2010 或 2012 或 2013

文档编辑：Sublime Text3 + Pandoc + LaTeX

性能测试工具：ANTS Performance Profiler 8

反编译工具：.net Reflector 8

代码托管：Visual Studio Online

UI设计工具：Blend for Visual Studio

脚本语言：IronPython、IronRuby

## 12.2基本的要求

基本要求：Visual Studio 、Blend、Visual Studio Online

# 13．培训计划（Training Planning）

## 13.1 团队培训的目的

团队培训主要的目的是教会大家团队开发的模式，熟悉软件整体架构，锻炼大家面向对象开发的方法，使用工具的能力，各类实用的技术点。

## 13.2 团队培训的目标

培训后应能让大家对整个工程有充分的了解，能独立开发插件，能独立修改自己插件中的bug，能与测试人员合作对代码修改，能熟练使用C#进行开发。

## 13.3团队培训时间

每周六上午，单周组内进行技术交流讨论，双周组织进行培训讲座。

## 13.4团队培训计划

1. C#组合式架构详解 MEF框架到插件式系统
2. WPF界面开发技巧
3. C#网络高级编程

## 13.5 团队项目技术交流计划

1. 开源库的使用技巧
2. Markdown语法和文档写作技巧
3. Blend UI设计