功能说明

这个项目的主要目的是为了让方便团队进行项目管理, 虽然这里以项目管理为背景进行描述，但是描述的过程里，理念是基于项目管理的三个基本元素，时间，质量和资源来描述的，因此我们应该可以轻易地把这个工具推广到项目管理以外的领域。

[功能说明 1](#_Toc277542865)

[愿景 2](#_Toc277542866)

[长期愿景 2](#_Toc277542867)

[短期愿景 2](#_Toc277542868)

[工具背后的理念 3](#_Toc277542869)

[IT行业 3](#_Toc277542870)

[功能说明 4](#_Toc277542871)

[配置管理 4](#_Toc277542872)

[术语说明 5](#_Toc277542873)

[界面说明 7](#_Toc277542874)

[集成团队管理 9](#_Toc277542875)

[风险 11](#_Toc277542876)

[界面说明 11](#_Toc277542877)

[普通的认证模式 11](#_Toc277542878)

[在线团队协作沟通 12](#_Toc277542879)

[获取团队成员的联系信息 12](#_Toc277542880)

[团队成员权限分配 12](#_Toc277542881)

[团队时间管理 13](#_Toc277542882)

[界面说明 13](#_Toc277542883)

[团队成员的行程安排 14](#_Toc277542884)

[团队成员获取自己的任务 14](#_Toc277542885)

[使用分类法管理时间 15](#_Toc277542886)

[安排团队会议 15](#_Toc277542887)

[接受会议邀请 15](#_Toc277542888)

[会议记录 16](#_Toc277542889)

[将工作任务转化成行程安排 16](#_Toc277542890)

[即时聊天工具可以根据成员行程安排显示忙闲状态 16](#_Toc277542891)

[需求管理 16](#_Toc277542892)

[界面说明 17](#_Toc277542893)

[需求之间的联系 17](#_Toc277542894)

[需求的属性描述 18](#_Toc277542895)

[需求/任务和最终产出物的跟踪关系 18](#_Toc277542896)

[项目进度管理 18](#_Toc277542897)

[界面说明 19](#_Toc277542898)

[设置需求的里程碑 19](#_Toc277542899)

[计划项目进度 20](#_Toc277542900)

[跟踪项目进度 21](#_Toc277542901)

[风险控制 21](#_Toc277542902)

[界面说明 22](#_Toc277542903)

[创建风险 22](#_Toc277542904)

[监控风险 22](#_Toc277542905)

[关闭风险并存入风险数据库 22](#_Toc277542906)

[决策管理 22](#_Toc277542907)

[界面说明 23](#_Toc277542908)

[回复工作任务 23](#_Toc277542909)

[团队知识分享 24](#_Toc277542910)

[界面说明 25](#_Toc277542911)

[文档模板 25](#_Toc277542912)

[流程模板 25](#_Toc277542913)

[站点模板 26](#_Toc277542914)

[资料管理 26](#_Toc277542915)

[项目产出质量控制 26](#_Toc277542916)

[成本和预算管理 26](#_Toc277542917)

[团队培训 26](#_Toc277542918)

[网络笔记 26](#_Toc277542919)

[要求软件 26](#_Toc277542920)

# 愿景

对于这个工具，我的远景是希望它能够作为在因特网上运行的，团队协作工具。中小团队，可以通过因特网连接到系统，将项目进行过程的数据保存在服务器上，避免了自己内部维护系统的需要。另外，一些中小团队，可以通过我们的平台将他们好的项目管理流程、经验分享给其他团队。

对于一些希望在自己组织内部搭建服务器的团队，这个模式我们也支持，因此这个程序的安装应该比较简便，便于其他组织快速部署。

# 工具背后的理念

我认为一个团队，之所以在一起工作，就是为了一个目的，产出能够达到市场需求，最后通过客户购买产出获取利润并赢利。这个团队之间进行合作，完成一个产品，或者服务，并最终获取利润的整个过程，叫做项目。因此，我们这里所说的项目管理不仅仅局限于IT行业的项目管理，也可以扩展到其他行业，例如建筑行业的项目管理，甚至还可以扩展到营销行业的客户管理上。

团队在从事项目的过程当中，为了实现赢利，团队需要在三个要素之间进行平衡：在有限的资源，有限的时间下，提供满足客户需要的质量水平的产品和服务。 下面举例说明：

## IT行业

资源 — 为了完成产品和服务所需要的一些生产资料，包括：

1. 团队组成人员。
2. 电脑，软件等软硬件。
3. 办公室、会议室等资料。

时间 — 要么是客户在项目进行时要求的最后完成时间，要么是基于市场竞争，整个团队自己定义的时间。

质量 — 最终产品和服务的质量。

对于项目管理者来说，可以将资源看成其得到的预算，而项目管理者平衡资源，也就是控制相应的成本，因此他在做出采购以及招聘人员时，都必须要考虑相应的成本。

对于时间这个要素的平衡，项目管理者实际上应该是通过管理好整个团队的时间（而不仅仅是做好自己的时间管理就可以了），来达到监控项目完成进度的目的。

对于最终产品或者服务的质量，项目管理者需要一系列的数据在项目进行的过程中，了解到自己的产品/服务的质量。

为了最好这三个要素（时间、质量、资源）的监控，项目管理者需要考虑以下几个方面：

1. 项目关系人管理 — 资源
   1. 团队—资源
   2. 项目团队所在组织其他团队的支持（例如人事、高层等）—资源。
   3. 项目的客户
2. 成本控制—资源
   1. 预算（或者说成本）。
   2. 处理突发事件所需要付出的成本。
3. 团队成员的时间分配—时间
4. 项目各个工作任务的进度—时间
5. 了解客户的需求，以便最终产品或服务满足客户的要求—质量。
   1. 还需要做项目团队能力分析（CMMI术语里面的CAM），确保项目组的能力可以满足客户的产品或者服务要求。
6. 处理客户在项目进行过程中的需求变更 — 质量和资源（成本）。
7. 定期的风险控制，确保项目的三个要素在可控的范围里。
8. 配置管理，确保所有的数据、数据变动、需求变动都是在团队范围内进行分享，所有成员可以立即获取关于项目的最新信息。
9. 决策分析过程，确保团队成员明白项目各个决定的决策过程以及结果。
10. 重大事故处理—资源。
11. 创建学习型的团队，因为当今世界的技术和知识更新太快，整个团队需要不断地学习新技术、新知识才有可能在竞争处于不败的地位。如何通过工具以及团队组织内部的人事制度来提升整个团队的竞争力，也是组织里人事经理需要考虑的事情。

团队最佳实践或者经验的累积，经验的累积过程不是通过口口相传完成的，而应该是采用工具的方式，以模板和示例的形式，是团队所有成员能够方便、快捷、容易地即时采用最新、最好的团队经验。

# 用户场景描述

# 功能说明

我们的短期目标是将这个程序做成一个项目管理方面的软件，长期目标是将这个程序扩展成分布式团队协作的平台。程序的首要目的是为了让团队成员能够快速分享各个成员对项目的贡献以及知识财富。初步划分出来，程序需要实现以下这些功能：

1. 配置管理
2. 集成团队管理。
3. 团队时间管理
4. 需求管理
5. 项目进度管理
6. 成本和预算管理
7. 风险控制
8. 决策管理
9. 团队知识分享
10. 项目产出质量控制
11. 团队培训

## 配置管理

配置管理的目的，是为了让团队能够快速地分享项目相关的信息，然而：

1. 由于有一些信息比较重要，不能在整个团队范围内公开，因此，需要在工具的支持下实现团队成员的**权限控制**。
2. 另外，因为信息需要在团队里面分享，分享的内容不应该是经常更换的。因此在信息发布之前，成员可以对信息草稿做多次变更，然后根据组织内部定义的**审批（review和approval）机制**发布给团队其他成员。
3. 一个团队的信息，通常由多个文件组成，这些文件可能由多个人负责维护，当这些文件都处于一个稳定状态时，团队可以将这些文件的状态保存为一个**基准线**，以便后续更改的时候能够有个对照的版本。
4. 对于新进团队的成员，或者项目其他干系人来说，为了了解团队信息，决策过程以及历史信息，团队分享的所有文件、工作任务的状态的修改都是自动有备案的。

### 术语说明

在项目管理术语里面，配置管理里有可配置项和非配置项的区别。其中可配置项需要做版本控制，而非配置项则没有这个要求。可配置项的定义是：

*软件配置是指一个软件产品在软件生命周期各个阶段所产生的各种形式（机器可读或人工可读）和各种版本的文档、程序及其数据的集合。*

*该集合中的每一个元素称为该软件配置中的一个配置项。*

*配置项主要有两大类*

*属于产品组成部分的工作成果；*

*项目管理和机构支撑过程产生的文档。*

*每个配置项的主要属性有：名称、标识符文件状态、版本、作者、日期等。*

例如

1. 项目计划，软件产品的代码，都是配置项。而周报之类的文档则不算是配置项。

**在我们的程序里，为了简便起见，默认将所有放在系统维护的数据，都看成是配置项**，只不过配置项之间可以采取不同的版本控制策略、以及审批策略。下面举例说明：

1. 在系统维护和管理的需求、任务、风险、人员等信息都需要做版本控制。团队成员对它们的属性域的更改都需要跟踪，跟踪方式是：

谁 在 什么时候 将 哪些域 从 原始值 更改到 新值。

这些信息的版本控制是在我们的数据库中处理的。

1. 项目执行过程中产出的文档，比如说测试计划、代码、报告等文档，这些文档的版本控制，暂时不打算由系统进行控制，而是由SVN或者Mercurial来实现。这些现成的开源版本控制服务器，都已经实现了很细粒度的版本控制，例如版本，基准线设置等。而且SVN等软件提供了很好的编程接口，供其它程序远程调用。

由于我的愿景是系统可以在因特网上，为其他中小团队提供远程服务（或者就说是云计算吧），因此使用Mercurial作为后台版本控制服务器的价值要好于SVN。

*当前我比较倾向于使用Mercurial来做，但是我还没有配置过Mercurial，所以先通过SVN实现，但是在技术实现上，我们应该能够将与版本控制服务器的连接抽象出来，即后面可以方便地从SVN更换到Mercurial上面。*

1. 版本控制和审批策略，即使在同一个组织，同一个项目过程里，也许也有不同的版本控制以及审批策略。为了实现最大程度的弹性化的版本控制以及审批策略，程序可以采用SharePoint里面文档库的概念来达到这个目的。另外，文档库还有一个优点就是，它可以通过支持文档模板的方式来快速复制团队成功经验。

#### 风险

SVN和Mercurial的版本控制理念比较复杂，非IT技术行业（甚至包括很多IT行业）的人对它们的操作感觉非常迷惑，因此如何简化这个版本控制过程是成功推广程序的一个重要因素。

### 界面说明

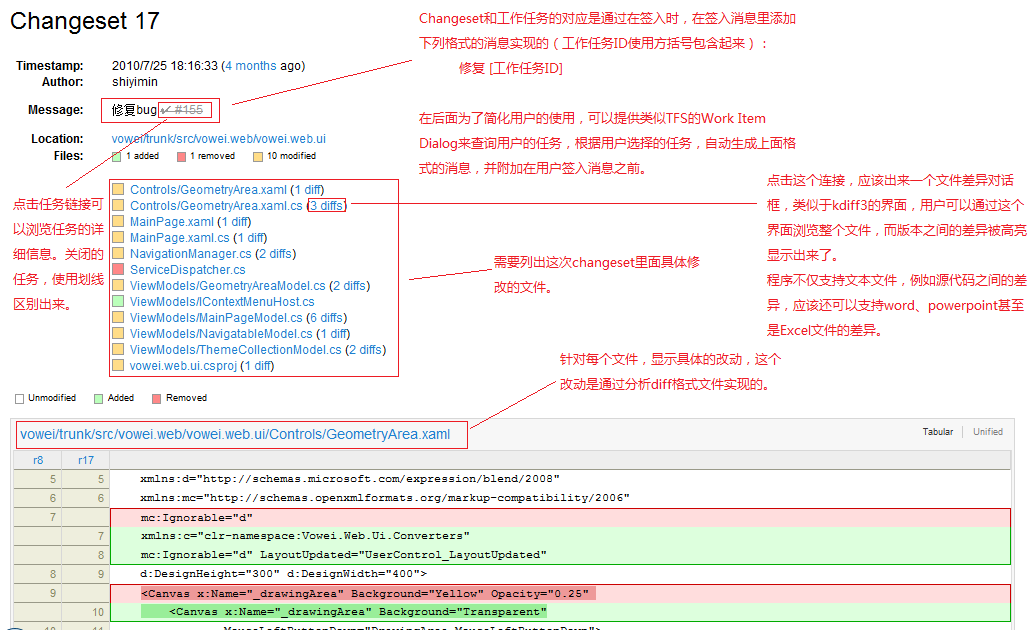
##### 可以将程序实现的版本控制与其他版本控制软件联系起来



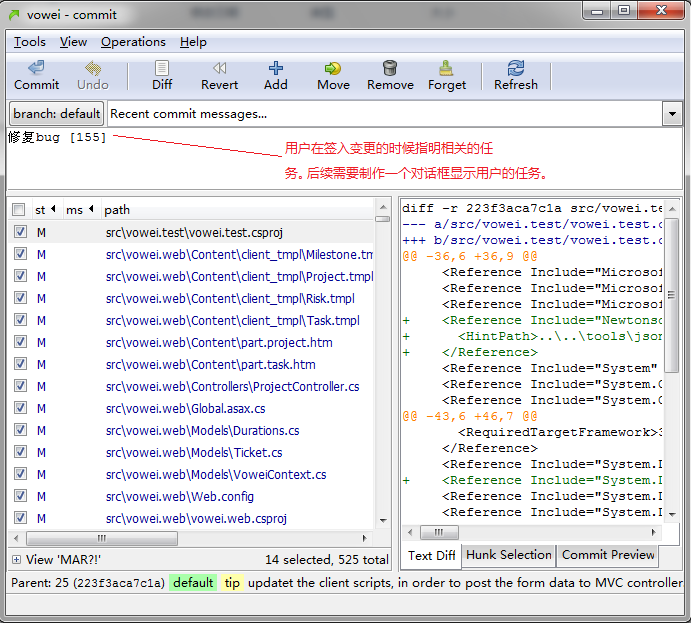
##### 记录对配置项属性的变动

参见上图

##### 浏览项目产出物在不同版本之间的变更



##### 签入变更



##### 设置变更基线

对于文档之类的配置项，直接使用SVN或者Mercurial现有的Tag和Branch功能就可以了。对于工作、bug等任务项目的配置项，不做变更基线，因为没有意义。

##### 审批流程

所有的配置项，例如文档修改后，发布之前都需要经过审批，这个过程可以仿造SharePoint的审批工作流的实现方式。实际上SharePoint支持下面三种工作流：

1. 审批工作流，即文档在发布之前，都需要经过某个具有审批权限的人审阅批准之前才可以发布。
   1. 文档发布之后，再次改动文档，则会进入新一轮草稿模式，团队其他成员看到的，还只是改动前的文档。
2. 收集反馈工作流，即团队成员上传文档后，可以通过SharePoint向指定的同事发邮件，收集他们对文档的意见，统一意见以后，发布文档。
3. 自定义工作流。

#### 尚待解决的问题—审批流程与外部系统的整合

因为涉及到用户数据保密的问题，如何在审批流程中集成外部系统需要研究一下。举个例子，比如说团队要求，关闭一个任务，需要在产出物签入源代码系统之后才能关闭，而且这个团队在内部部署了SVN系统，而且产出物最终签入到SVN系统之前，需要经过其他人审阅才行。具体场景是：

1. 一个团队成员做完一个任务，有一个产出物的草稿—比如说代码。
2. 他将代码签入到SVN系统里，在签入的时候，选择要关闭的任务。
   1. SVN系统自动使用我们写好的插件向我们的网站发送一条消息。
3. 网站接受到消息后，启动审阅流程，通知团队其他成员审阅改动过的代码，在邮件里，会附上原始任务，和代码变动的diff链接。
4. 团队成员审阅通过以后，在网站上，点击一个通过按钮。
5. 网站这时才通知SVN插件将变动过的代码签入到SVN系统。
6. 网站关闭任务。

而如果团队内部使用的SharePoint进行管理，又需要为SharePoint写一个这样的插件，风险比较大。

##### 文档检索

一个团队在项目进展过程中，总是会产生大量的文档，而如何高效地管理这个文档是一个让人很头疼的问题。当前有两种解决方案：

1. 规定好目录结构，以及文件放置标准，团队成员按照这个标准将文件保存到相关的位置里。比如说，所以团队建立了一个这样的文件夹结构：一层目录是“计划”，二层目录是“测试计划”和“开发计划”，那么团队成员将所有测试计划类的文档保存在“测试计划”里，将所有开发计划类的文档保存在“开发计划”里。

这种方式的缺点是，即使你定义了很详细明确的目录结构和文件放置标准，基于团队成员的背景不一致，对同样的文档，可能会有不同的分类观点。而且，按照团队定义的标准，有些文档可能可以放在多个分类下，例如，C#方面的书籍，是归于“.NET”分类合适些，还是归于“编程语言”分类更合适些呢？

1. 另外一种方式就是使用全文搜索，这种方式的好处是，在保存文档的时候，你不需要考虑文档放置的文件夹。只要文档里面有匹配的关键字，那么就可以找到文档。比如说，在保存C#方面的电子书时，只需要将它保存在“电子书”文件夹，在检索的时候，用户可以通过“C#”，“.NET”等关键字搜索到文档。

这种方案的缺点是，有时检索的内容过于宽泛，得到很多的搜索结果。

综合上面两个方案的优缺点，我建议的方案是，采用标签和全文搜索结合的方式来实现文档管理。

1. 以收藏夹的形式管理所有的任务和文档，可以将收藏夹看成是一个虚拟文件夹，里面保存的都是任务和文档的超链接。
   1. 收藏夹可以创建子文件夹。
   2. 收藏夹可以有类似博客里的标签。
   3. 收藏夹的操作最好做成类似资源管理器的拖拉操作。
2. 标签又可以看成是另外一种虚拟文件夹，任何在系统管理的文档、虚拟文件夹都可以有标签。
   1. 标签里面可以创建子标签，点击父标签，可以采用类似资源管理器文件夹的显示方式，也可以采用合并所有子标签里面文件内容的方式显示。
   2. 标签和收藏夹的区别是：
      1. 收藏夹可以是用户个人的，用户可以选择不跟团队成员分享私有收藏夹，但是标签只能是公开的。
      2. 标签可以直接放在文件和任务一层，但是收藏夹不行。
      3. 收藏夹可以保存系统外部的内容，例如外部CMS系统的链接，但是标签不行。
3. 最后，所有的文档、任务、标签、收藏夹都可以使用全文搜索的方式检索。

# 集成团队管理

集成团队管理，不仅仅包括团队自身成员的管理，还有将会影响团队工作的其他组织成员的管理。例如，团队所在的公司内部的人事部门的一些制度定义，可能会影响到员工的士气；公司内部的行政部门与团队成员的交互的方式，也可能影响到员工士气；而培训部门是否能够制定行之有效的培训计划，对整体团队的技能提高也有很大的影响。最后，除了公司内部的其他团队以外，团队的客户，以及客户高层都会对整个团队的工作有着很大甚至是决定性的影响。作为项目经理，做好干系人管理，以及与这些干系人保存流畅（定期或者不定期）的沟通，对于项目的成功至关重要。

因此在程序里，我更愿意将团队管理泛化成干系人管理，即项目经理管理的应该是一个虚拟的，由客户、组织内外成员组成的团队。而项目经理根据虚拟团队成员的特点，分别采取不同的沟通以及管理技巧，另外，虚拟团队成员对项目组的资料和信息的访问权限也是不同的，比如说，客户就不应该可以访问项目团队内部成本的信息。

既然团队管理需要泛化成一个虚拟团队，**因此要求所有团队成员在系统里有一个账号就有点不合时宜**，比如说你没有办法强求客户在系统中注册一个账号。实际上，账号无非就是唯一标识团队成员的方法，既如此，我们完全可以使用电子邮件地址作为虚拟团队成员的唯一标识。只要访问系统的成员，能够通过一种身份认证方法，表明自己就是这个电子邮件的合法拥有者，而且，项目经理也在项目中，对该电邮地址分配了访问权限，那么这个人就可以访问系统[[1]](#footnote-1)。因此身份验证有以下两种策略：

1. 一种是**普通的认证方式**，团队成员使用自己的邮件地址，一般来说，如果是公司成员的话，最好使用自己的公司邮箱注册。设置好密码以后，就可以登录系统了。
2. 另外一种是**委托认证模式**，即使用现有的Windows Live、Google Talk、Facebook等（中国的QQ也会适应这个潮流提供类似的服务的）委托身份认证机制，通过其相应的服务提供商的认证服务验证用户身份。

这种实现方式的好处是：

1. 用户不需要重复注册，以及记住这么多网络平台的用户名和密码，从而给用户带来不必要的麻烦。
2. 团队成员可以使用现有的聊天工具，以及自己现有的联系人进行在线沟通，不仅避免了我们重新实现一个聊天工具的负担，而且也避免了用户重复添加联系人到自己联系列表中的麻烦。
   1. 而且，现在很多聊天工具提供商，例如google talk, Windows Live Messenger都支持手机客户端。这样做可以让我们的系统可以迅速支持世界范围的用户。
   2. 另外，我们还可以仿造MSN机器人的模式，可以在google talk，或者windows live messenger上，根据用户的行程安排，发送即时消息提醒。而且这个机器人甚至应该可以让用户通过聊天工具汇报自己的进度等信息。
3. 可以复用先有的聊天工具、Active Directory和LDAP技术，根据电子邮件获取到用户的详细信息，不需要在我们的系统中保存重复记录。
   1. 而且，用户只要在一个地方更新联系人或者自己的信息，就可以复制到其他网络系统里。

## 风险

1. 虽然上文描述中，猜测QQ会实现类似于Windows Live的技术，然而这一切尚不明朗。另外很多人注册QQ的方式依然采用的是QQ号[[2]](#footnote-2)的模式，为了兼容QQ号和系统采用电子邮件进行身份认证的方式，无论用户是否申请了QQ邮箱，默认情况下都会在用户的QQ号后面附上qq.com这个域名。
2. 如何实现在采取多个聊天服务提供商服务的用户之间的沟通进行同步，是一个技术难题。但是好在：
   1. 团队成员，即使是泛化的虚拟团队成员会采用同一个聊天服务提供商的服务。
   2. Windows Live正在试图通过与多个网络服务商之间的连接，例如Live与facebook, linkedin的连接，Live本身就处理了多个网络服务之间的同步问题。

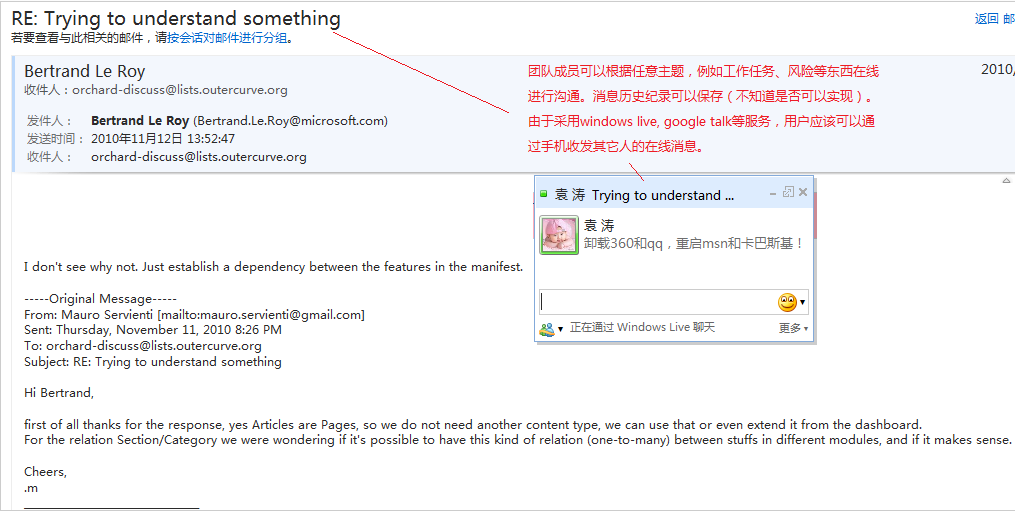
## 界面说明

下面的界面说明里的截图取自其它网站。

### 普通的认证模式



### 在线团队协作沟通



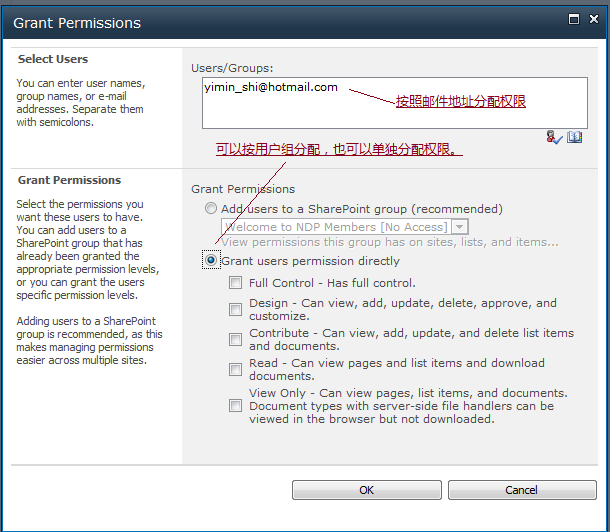
### 获取团队成员的联系信息

如果团队成员使用的是公司信箱进行注册，我们应该可以提供接口，由团队所在的公司的LDAP或者活动目录服务器自动获取成员的详细信息。

而如果成员是使用windows live, google talk等服务注册，我们可以通过微软或者google提供的公开函数调用接口获取到这些信息。

### 团队成员权限分配

这一点可以参考SharePoint的权限分配模式实现。



# 团队时间管理

在市场竞争面前，及时推出产品和服务是一个关键的成功因素。因此有效管理好团队的时间至关重要，但是问题是很多职业人士并不擅长于管理好个人时间，更毋庸说管理好团队的时间了。我们认为，一个团队的时间是由以下几个部分组成的:

1. 团队成员从事真正生产的时间，例如程序员编码，测试人员执行测试用例，营销人员推广产品等活动，都可以看成是为了团队推出产品进行盈利所消耗的时间。
2. 团队成员用于沟通协调的时间，例如为了解决团队中针对产品功能的某些争议，所需要进行的会议；团队定期召开的进度会议等都可以看成团队之间进行协调所耗费的时间。
3. 团队成员自身用于提高技能的时间，例如团队成员为了提高自身的技能，利用工作空闲时间定期花费在学习上面的时间。

为了有效地管理和跟踪整个团队的时间，项目经理需要掌握上面所有的时间，然而另外有一个问题，对于第一、第二种时间，项目经理是有权限访问和了解的。而对于第三种时间，为了保护团队成员的隐私，项目经理并没有权限去了解详情，但是项目经理在协调团队会议的召开时，需要了解到某一特定时间，某位成员是否有时间参加。

从团队成员的角度，他/她本身也需要在三种时间里做平衡，因此团队成员可以采用**时间管理**里面的分类法进行有效管理。比如说，团队成员可以为“紧急且重要”、“重要不紧急”、“紧急不重要”和“不重要不紧急”分别标注好不同的颜色，对每一个工作任务、会议邀请通过分类并自动标注颜色的方式，快速判断自己需要优先处理的工作。

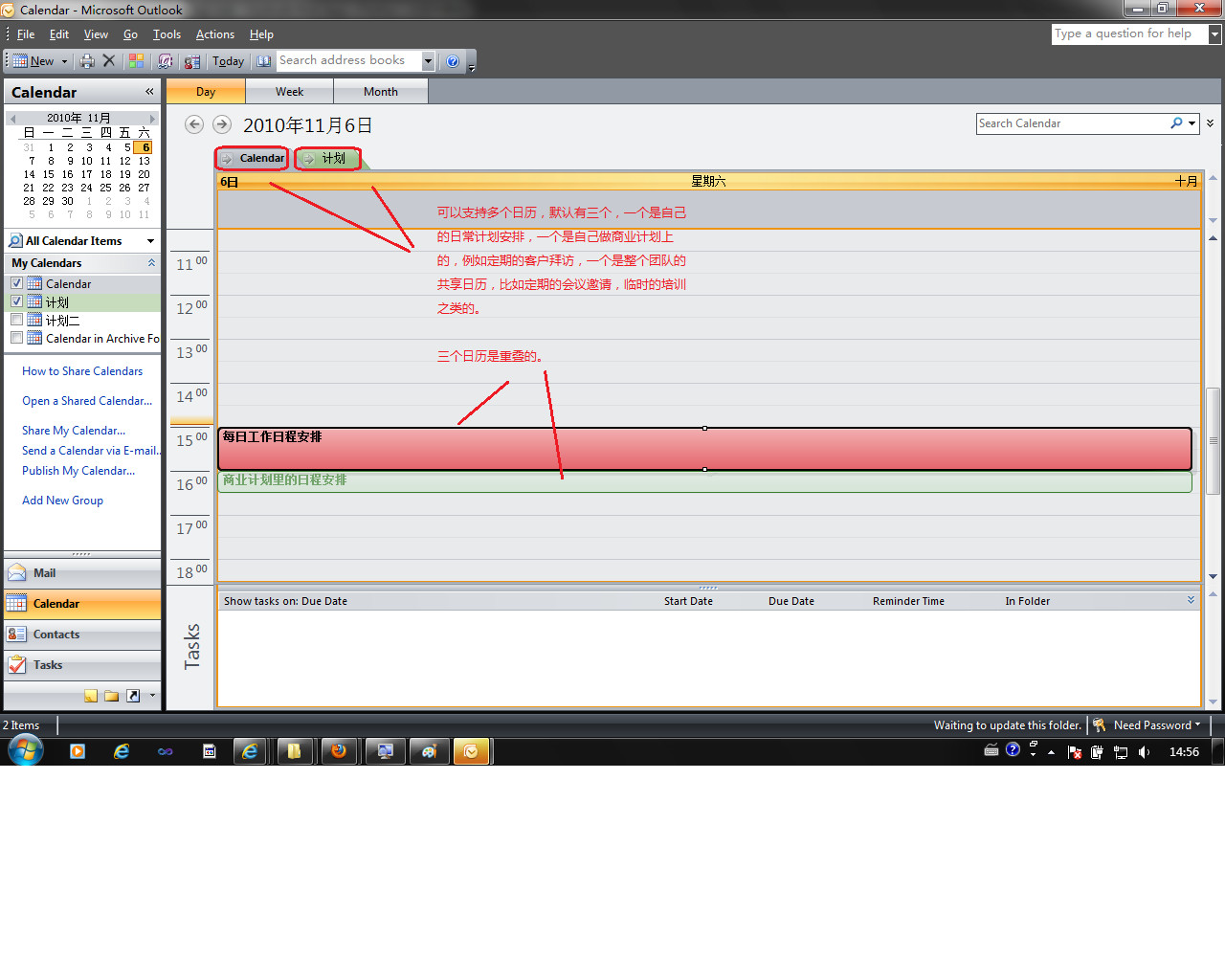
由于在接受任务的时候，团队成员可能比较难快速地给工作任务进行分类，程序可以将用户自定义的工作任务映射到时间管理的类别上去，比如说，团队成员可以自己定义分类“老板的事情”，将其映射到“重要且紧急”的类别上去。

为了简化团队成员（包括项目经理）在时间管理上的操作，程序可以支持团队成员使用电邮、即时消息工具或者手机进行时间管理。为了能够在多个工具之间同步，团队的形成安排分享可以使用iCal格式[[3]](#footnote-3)传输。

## 界面说明

下面的行程安排使用Outlook的日历界面进行说明，大部分日历软件，例如Google Calendar,各式手机上的日历软件都支持类似的格式。

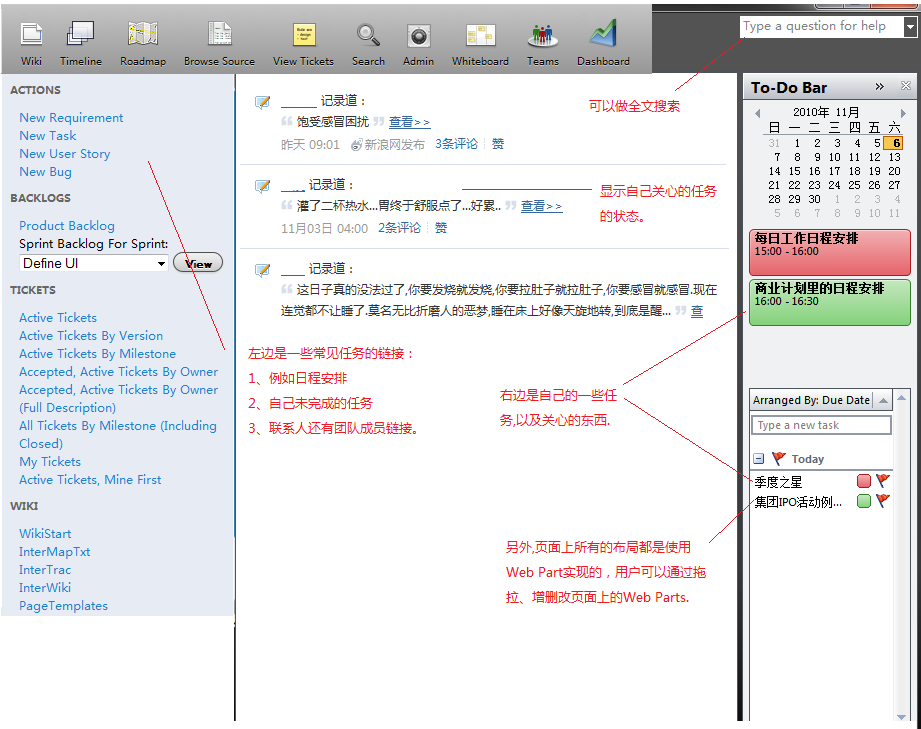
### 团队成员的行程安排



### 团队成员获取自己的任务

团队成员既可以通过程序提供的界面来获取自己的任务列表（详细参考项目进度管理），也可以将自己常用的软件连接到网站，通过软件收发任务：

1. 对于支持任务的软件，可以采用发送任务格式的数据给客户端软件，如下图：



1. 对于不支持任务的软件，但是可以收发Feed的软件，可以通过Feed的模式传输。

### 使用分类法管理时间

注意上面图片里，工作任务和行程安排是用不同颜色标识的，这样方便团队成员一目了然任务的优先级。

### 安排团队会议

参照Outlook的会议邀请界面。另外，团队成员可以就任何一个任务、风险或者宽泛一点说，一个工作任务（Ticket）来创建会议邀请，如果使用这种方式创建的话，会议邀请将自动附上相关工作任务的链接。

### 接受会议邀请

参照Outtlook的接受、拒绝会议邀请和建议会议新时间的界面。

### 会议记录

会议结束后，会议主持人一般要给与会者发送会议摘要。在会议行程项上会有一个按钮，方便会议主持人创建会议记录，会议记录与会议是互相可链接的，这就意味着在会议记录里有原始会议行程的链接，而会议行程里有会议记录的链接。关于会议记录的实现方式，请参看网络笔记的实现方式。

### 将工作任务转化成行程安排

团队成员可以将任务转化成一个行程安排，这样团队其他成员在安排会议邀请时，在邀请界面可以看到该成员在指定时间段是忙碌状态。

### 即时聊天工具可以根据成员行程安排显示忙闲状态

如果团队成员使用的是即时聊天工具，例如Windows Live, Google Talk等工具的登录系统，程序可以根据团队成员的行程安排，设置其忙闲状态。例如，可以根据会议邀请设置那个时间，团队成员是离开状态；对于处理任务的环节，设置其为忙碌状态。

# 需求管理

所有的项目，都必须要有需求才有可能进行生产和交付操作。作为项目经理，有效地管理和跟踪需求管理，是保证最终产品和服务满足顾客要求的先决条件。一般来说，一个项目的需求并不一定都是相当明确的，可能顾客只会含糊地说，做一个产品，功能和某某产品类似就可以了。作为项目团队，一定要将这种模糊的需求细化到明确的需求。**因此一个需求有可能会引出多条需求出来，程序需要跟踪需求之间的联系性**。

另外，需求由以下几个元素组成：

1. 必须要有其来源，在系统中，需要跟踪到需求的来源干系人，项目团队应该有这个干系人的详细联系信息。
2. 必须要有来源文档，例如，通常在合同里，会明确下来一些需求，那么程序应该将这个文档当作配置项保存在配置系统（参见配置管理）中，而且需求项里应该有该文档的链接，保证一一对应的关系。
3. 必须要有唯一的负责人，每个需求必须要有一个负责人负责，这样确保得到最大的执行力。通常来说，一个大的需求，细分后，可能大需求的负责人无法独立完成，那么他可以通过组建一个团队，并将细分的需求分配给团队成员的方式实现。

既然需求可以分配给团队成员，那么需求还可以进一步细分，细分到任务级别，这种方式，可以保证需求到任务的联系是一气呵成的，详情请参看项目进度管理。

1. 必须要有接受决策分析过程的历史记录，这一条请参见决策管理。
2. 可以考虑添加需求的优先级，要么采用时间管理里面的分类法进行管理，要么采用商业价值法（即虚拟团队—包括客户讨论，以价值的方式）进行管理。

至于需求变更管理，我们将其泛化到需求管理里，即创建项目时，新的需求，我们也认为是一个需求变更。需求接受或者否决的过程，可以通过决策管理来实现。

## 界面说明

### 需求之间的联系

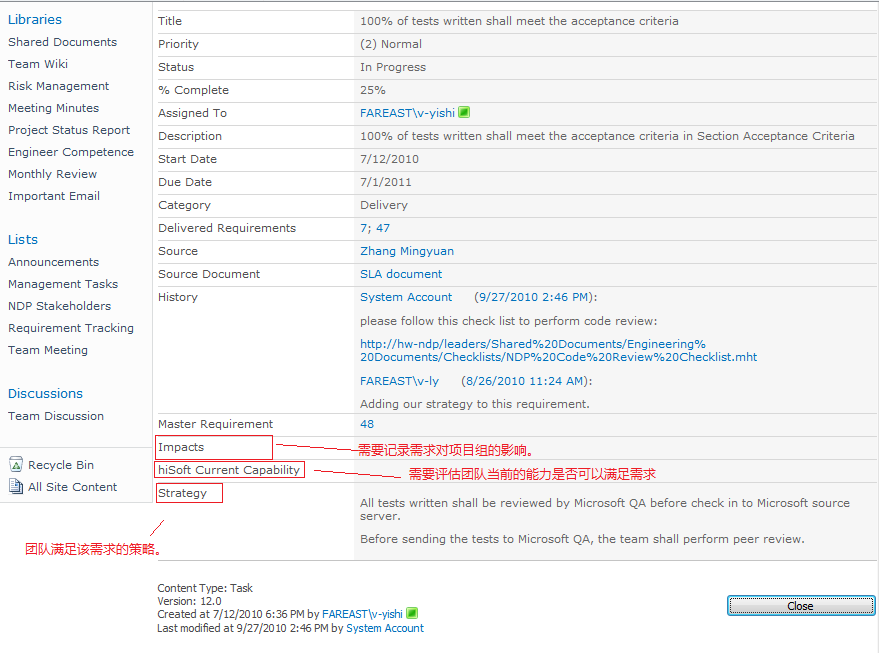




### 需求的属性描述

除了在前面描述的需求必填属性以外，项目经理需要还需要指明需求应该在那个里程碑里完成，详情请参考项目进度管理。

每一个需求，都需要说明这个需求的影响因子，即对成本的影响，对项目进度的影响，以及团队是否有实力完成这个需求。



### 需求/任务和最终产出物的跟踪关系

为了有效跟踪需求的实现，需要将需求/任务和最终产出物联系起来，这样整个团队才能方便快速的了解整个项目的进展程度。可以通过在将最终产出物签入配置系统（参见配置管理）的时候，在签入消息里链接相关的任务，而该消息将以评论（comment）的方式显示在需求/任务的详细信息页面上。请参见可以将程序实现的版本控制与其他版本控制软件联系起来和签入变更里面的截图。

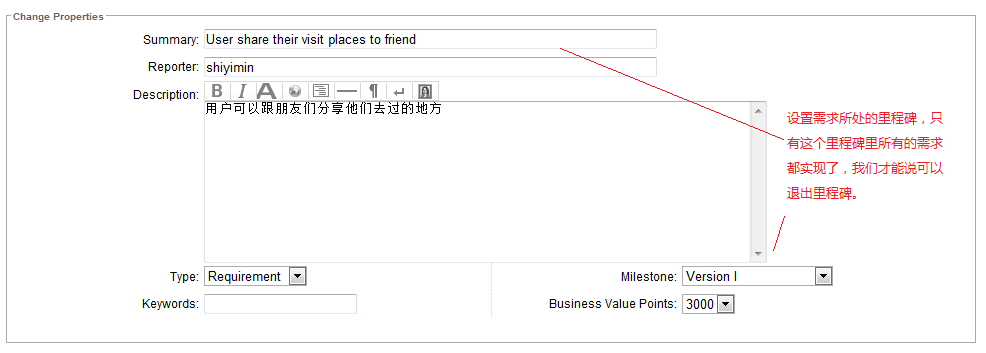
# 项目进度管理

一个项目，一般会划分为多个里程碑进行交付。划分里程碑的方式，可以让项目团队知道自己的产品质量已经达到什么程度。里程碑的定义，通常是在计划阶段，虚拟项目团队会讨论那些需求完成以后，才能算是达到某个里程碑—即退出标准。就IT行业来讲，当前有多种项目管理模式，比如说SCRUM，传统的瀑布模型等，虽然里面对于项目执行阶段有不同的术语，我们在系统中将其泛化为里程碑，也许更能让非IT行业的人员接受。

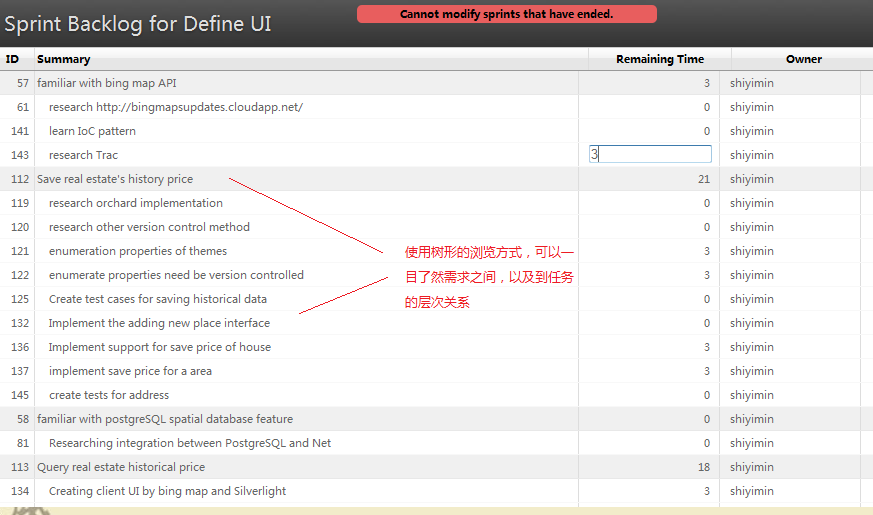
里程碑里可以继续细化成多个里程碑，这样的方式，应该可以支持类似SCRUM的sprint模式。

## 界面说明

### 设置需求的里程碑



### 计划项目进度



### 跟踪项目进度



# 风险控制

在项目执行过程中，并不一定总是一帆风顺的，在执行的过程中，总是或多或少会有一些因素影响着项目的成败。风险控制是项目经理在日常工作中需要执行的一个很重要的工作。风险控制在执行的过程中，应该是定期的、由项目经理召集相关干系人根据项目的进展情况评估风险。一般来说，团队所处的组织有一个**历史风险数据库**，项目经理可以先从这个历史风险数据库开始，逐条评估项目在进展的过程中，是否会发生类似的风险。如果组织里面没有风险数据库，那么一开始，项目经理可以和干系人一起进行头脑风暴，将项目里可能的风险找出来。例如，人员的流失就是一个需要考虑的风险。

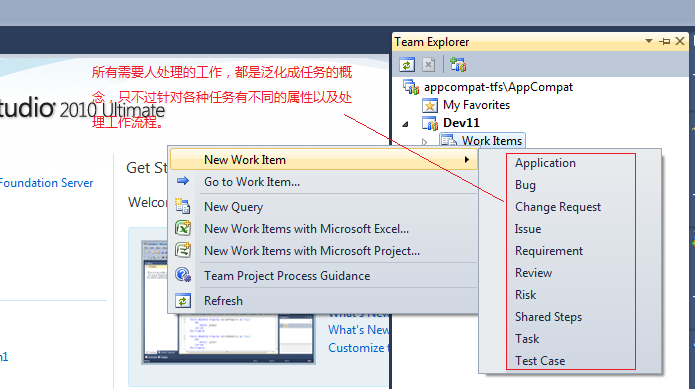
当风险明确后，团队需要评估风险发生的可能性（取值1 - 10）以及风险发生后的影响程度（取值1 - 10），两个值的乘值就是风险值。**为了节省团队的精力，团队可以对风险设置一个阀值**，例如，平常状态下，团队每个月评估一次项目风险。如果阀值有大于64的风险的时候，那么项目进入高风险状态，项目经理可能采取不同的沟通措施，比如说汇报给高层以及通知客户来采取相应的措施，并且团队可能会每周进行一次风险监控和评估。如果阀值有大于36以上的风险，可能团队会每两周进行一次风险监控。

针对需要监控的风险，项目团队需要考虑预防风险发生的预防措施，并考虑如果风险真的发生，团队应该采取的应急计划。针对同一风险，可以有一套预防措施和多套应急计划备案，以便在一套应急计划不能消除风险带来的负面影响时，有其他备案可以参考。当下次风险评估关闭风险时，团队可以记录下到底是因为预防措施处理得到消除风险，还是因为有一套应急计划降低风险的影响，记录好以后，即可将风险存入组织的风险数据库当中。

## 界面说明

### 创建风险

因为需要定期监控风险，所以风险控制并不是做完一次风险计划就完事的任务。它应该和普通任务一样，定期执行。在系统里，我们实际上是将风险看作一个任务对待的。

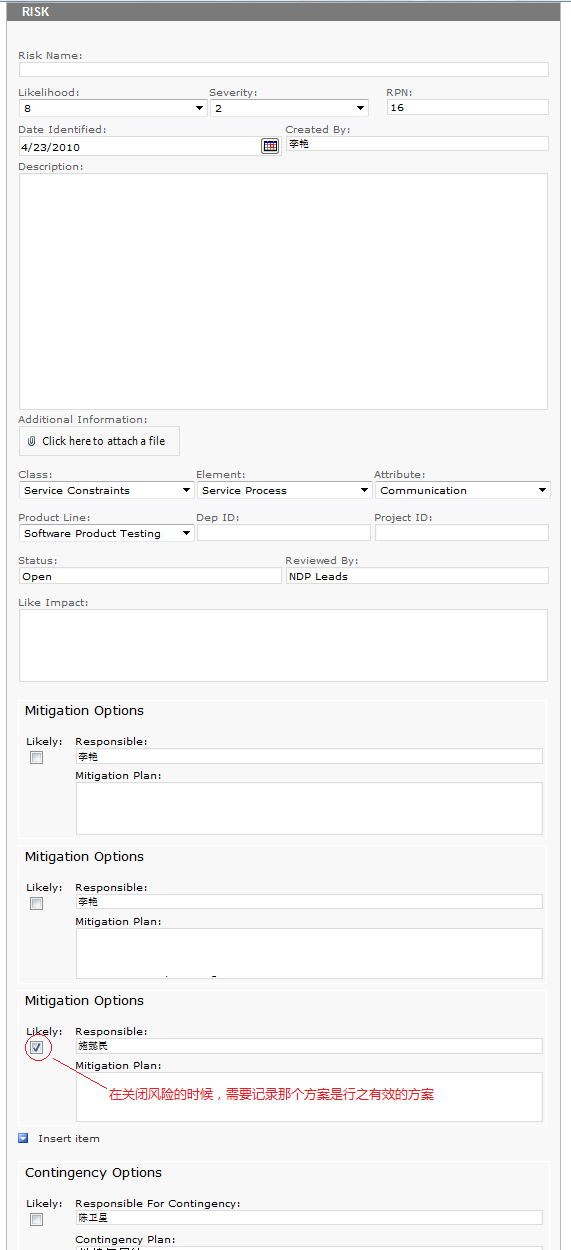


### 监控风险

在定期监控风险的过程中，团队的目的就是降低风险的RPN值，因此必然会涉及到风险的各个属性的变更，以及做出变更时，当时团队的决策过程。这一点，采用在配置管理和决策管理当中描述的方式实现。

### 关闭风险并存入风险数据库

无论风险是否消除，或者是真实发生，都要关闭风险。然而，关闭风险时，要记录关闭风险的原因，比如说预防措施是否得当啦，哪套应急计划是行之有效的之类。



# 决策管理

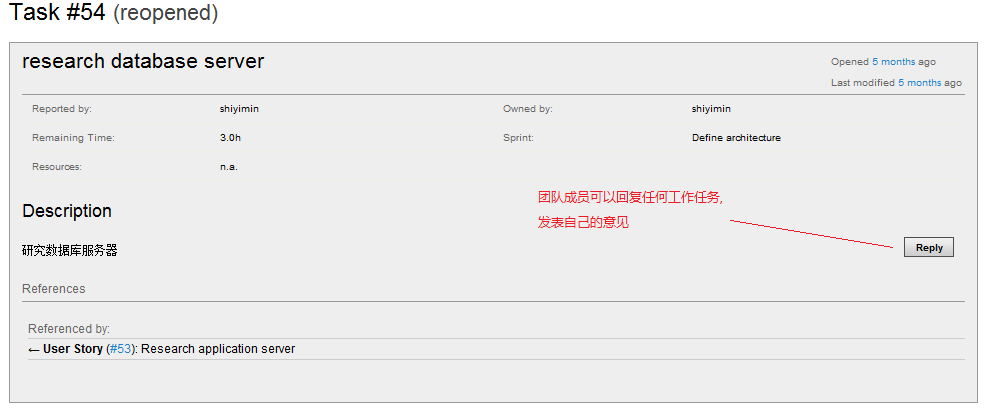
在项目执行的过程里，总会有决策发生，无论这个决策是一个人拍脑袋想出来的，还是团队经过正规流程商量实现的，都应该记录下当时决策人以及决策历史。

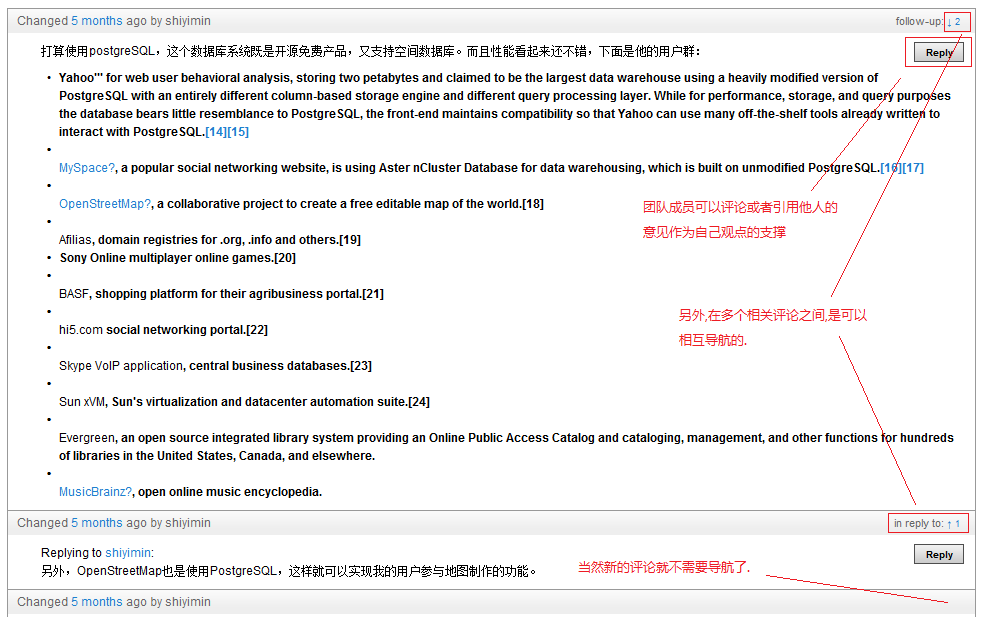
为了简便决策管理的实现，系统采取评论（或者说回复，comment）的形式记录决策过程— 有点像论坛的跟贴。

## 界面说明

### 回复工作任务

只要是系统中保存的内容，团队成员均可以采用回复的形式发表自己的意见，同时也可以采用引用的方式引用他人的观点支撑自己的意见。





### 最终决定方案

**当前的决策过程中，暂时先不支持决策方法论，只采用回复加投票的方式实现**。以后在考虑支持决策方法论。



# 团队知识分享

我们认为，团队的知识是由两个部分组成的，经验与资料。之所以这么区分，是因为很多资料并没有及时或者说有效地转化成团队的经验。比如说，你可以写一个文档，说明团队里的文档应该如何编写，但如果这个文档写出来以后，无人参考，那么这个文档就应该定义成资料。而如果有人依据这个文档编写指南写出最终的文档，那么我们才可以说这个员工有了团队如何编写文档的经验。

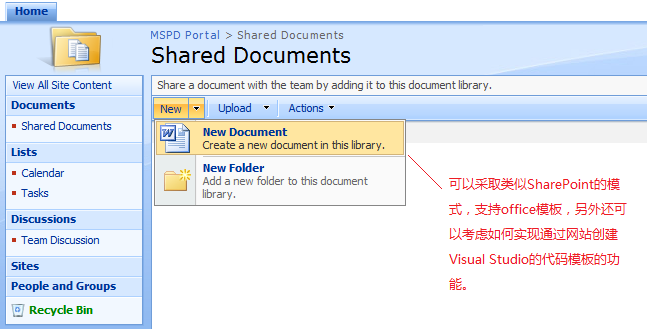
继续以上面的例子说明问题，虽然在管理过程中，我们可以强迫所有员工必须在写文档之前，先看文档编写指南，但问题是，确保这一条指令的执行，往往需要耗费相当大的精力。另外一个问题是，即使员工看完指南，根据员工自身的知识背景以及学习能力，他在写文档的时候，依然会犯下种种小的错误，这个问题同样影响着我们的指南的执行力。

因此我认为，将团队的知识文档化，最差情况下，实际上团队的知识资料化。为了有效地将团队知识迅速转换成经验，并快速地在团队内部得到最高的执行力，我认为，应该将这些知识以模板和示例的方法设计出来。比如说，与其费力气写文档编写指南，并在团队内部强制执行，不如直接将写的最好的文档，制作成一个模板，将可替换内容使用浅色显示，这样团队新成员根据这个模板创建文档的时候。默认情况下，模板就已经给员工提供了一个很好的实例，如果再把指南的超链接放在模板里，这样经验的复制将得到最好的效果。

## 界面说明

### 文档模板

对于项目执行过程中，所需要的文档模板，完全可以复用现有工具的功能，例如微软的办公系列工具里，都有将文档保存为模板的功能。针对于程序开发领域，Visual Studio同样支持代码和项目模板。



### 流程模板

一个团队的流程，实际上就是一个工作流的确立，确定工作流中每个节点由哪个团队成员负责，每个节点所需要的输入文件和数据是什么，而节点又应该有哪些输出数据，以及节点之间的依赖次序。因此流程模板，实际上就是将团队中，选中的工作流的上面4个要素记录下来，分享给其他团队。

### 站点模板

除了可以将文档作为模板，如果一个公司有多个项目，这个公司完全可以将管理最好的团队使用的项目管理站点，转化成一个模板，这样新团队在成立的时候，就可以立即采用最新最好的管理流程、和文档模板了。站点模板，实际上就是整合下面这些东西：

1. 团队所有的文档模板
2. 团队所有文档库的模板、版本控制、审批制度的设置。
3. 团队文档结构的设置。
4. 团队的流程模板。
5. 团队的用户组与权限设置。

### 资料管理

团队成员可以通过博客、Wiki等方式在团队内部实现信息的分享，这个功能可以采用市面上流行的开源博客或者Wiki软件实现。

# 项目产出质量控制

# 成本和预算管理

# 团队培训

# 网络笔记

# 要求软件

在初步的版本里，我们需要以下几个软件：

1. 开源的SVN系统，

1. 这个技术实现，可以参考OpenID技术。 [↑](#footnote-ref-1)
2. QQ号这种模式，不适用于其它国家，因为其它语言念数字不象中文这么容易。 [↑](#footnote-ref-2)
3. http://en.wikipedia.org/wiki/ICalendar [↑](#footnote-ref-3)