# 引言

早在2008年github的建立为编程人员带来了一种新的开发方法——一起组队开发。Github不仅是采用了git作为代码仓库管理和web的管理界面，还为开发人员提供了订阅、讨论组、文本渲染等方便的功能。正因为这些便利的功能，在时间的积累下，github人气逐渐升高，并形成了一种社交编程文化。[1]截止2013年7月，Github公司的收入每年增长300%，公司员工总数也从达到了158，用户中甚至不乏美国航空航天制造商Lockheed Martin、Twitter、Facebook以及白宫、世界500强公司这种“大玩家”。[2]github的兴起与成功有许多原因，但是确实、充分地说明了众包思想的实用性，和用户对这种基于思想的系统的需求。

由于定位或是其他原因，Github面向的用户至少需要入门的编程水平，当一个完全不懂编程的用户有自己的需求并希望以众包形式实现时，用户并不能使用github开始他的想法。另外即使是一个有一定编程能力的用户，因为用户的项目不会被社区推荐，当新用户进入社区需要帮助时，也只能通过自己的宣传来让其他用户知道此项目并予以帮助。亦或是一个希望找到新鲜想法的新用户，没有其他用户的推荐，用户也很难高效地查看自己需要的项目，等等。

这些种种对用户的定位不同而未实现的功能正是本文将要介绍的系统所力求并提出具体方法解决的方向。

近年来，团队式协作开发变得越来越重要，开发的方法也变得多种多样。其实早在1991年开始的Linux内核项目就是至今最著名、最成功的开源团队开发的项目之一。[6]芬兰的Linus Torvalds尝试在英特尔x86架构上开发的类似Unix的自由免费的Linux内核虽然只是完整的linux系统中比较小的一部分内容，但是这部分内容决定了系统如何工作，正如其名，是linux中不可或缺的一部分。[7]

在开发者的内部版本管理上，Linux采用Git版本控制系统进行版本与代码管理。[6]Git作为一个版本控制系统，它的优势主要体现在其“去中心化”的设计思路，每个开发者的版本库都是自己的中心版本。同时，Git具备机器优秀的分支创建、操作以及删除的能力，衍生出了非常多的基于分支的高效的协同工作流模式。

# 相关研究

“众包”这一概念从提出至今得到了工业界和学术界的广泛关注。众包(crowdsourcing)此概念是美国杂志《连线》的记者杰夫·豪(Jeff Howe)在2006年5月提出的。他对“众包”的定义是：“一个公司或机构把过去由员工执行的工作任务，以自由自愿的形式外包给非特定的（而且通常是大型的）大众网络的做法．众包的任务通常由个人来承担，但如果涉及到需要多人协作完成的任务，也有可能以依靠开源的个体生产的形式出现．”[3]

相似的另一个概念“外包”是指用合同的而形式把任务派发给另外的个人或组织来完成。外包业务一般是借助计算机来完成的任务，比如软硬件的开发和维护。软件提供方通过整合硬件、程序来完成外包业务，比如提供软件服务、外包数据库、制作网站等。但是有一些任务不是算法能够解决的，这类“机器难问题”就可以通过众包的形式来解决，将任务分发给互联网上未知的大众来解决，比如wikipedia、recaptcha等。现实生活中有许多类似场景，这也就意味着众包思想广阔的发展空间和光明的前景。

类似recaptcha[4]是如何运用众包的思想来实现的。Recaptcha是由卡内基梅隆大学所发展的系统，主要目的是通过captcha[5]技术来帮助典籍数字化的进行，这个项目将电脑扫描下来且无法识别的文字显示在captcha问题中，通过人类在完成captcha问题的同时进行识别。

Github的用户间协作完成项目亦可近似地认为是众包的形式。一个用户提出需求并做出初步实现，其他用户帮助其进行更具体的实现。但是在众包更加广泛应用的现在，用户群体开始变大，不再仅仅局限于拥有编程能力或是解决能力的人，用户也可以是提出自己需求、展示市场痛点的人。这些用户只能通过语言或文字来表现自己的想法，无法通过初步的结构来给帮助他的用户展示自己的想法，那么此时就应该给予用户这个功能，拓展面向的用户群体。

与其说是拓展用户，开拓新用户群体更像是帮助这些用户让他们能够通过众包的形式来实现自己的想法，或是通过与其他用户的交流发现这个想法其实是天马行空、并不现实的。

基于这类用户的需求，一个提供给用户以文字描述自己需求、能够方便的浏览高质量内容、其他用户能方便地浏览并在需求和代码都能给予帮助的平台是有很大的发展空间的。

Scrum

当想法提出者有一个灵感的时候—仅仅是一点小小的需求，此时可以直接向系统提出此创意，此时系统会由标题抓取关键词，并首先自动向提出者提供相似的创意。如果提出者觉得系统推荐的以往的创意符合自己的预想，那么可以自行关闭创意；如若觉得并不符合自己的需求，可以再添加更加具体的描述，然后发布创意。此时开始进入第一阶段，讨论可行性以及再进一步的润色创意。如果在此阶段，经过讨论发现此创意实现难度极大甚至基本无法实现，提出者可以自行关闭创意。如果经过润色补充后，创意逐渐成为一个充实、可执行的项目，那么提出者可以广招贤士，开始进入第二阶段—组建项目团队阶段。

在组建项目团队阶段，提出者需要首先选择一个项目负责人，此负责人在此后的进度中拥有和提出者相当的权限，个别操作可以超过提出者作为此项目的平台管理之外的最高权限。在确定项目负责人后，平台会开放平台全用户对此创意开发团队的申请权限，所有的用户都可以向开发团队提出申请加入开发团队。同时，系统会自动地向关注此创意的用户发出邀请，收到邀请的用户也可以通过邀请进入此创意，然后选择是否加入开发团队。

在一段时间（1d，3d，5d，7d）的申请阶段同时，项目负责人开始审核用户发来的申请，在申请中会有用户的自我描述，建议此时尽量展示自己的优势以及对项目有帮助的方面，增加自己申请通过的几率。在项目人员基本确定后，项目开始正式进入开发阶段。如果负责人认为项目开发人员并不够，可以再延长最后一期申请时间，也可以直接裁定开发失败，做好记录后关闭开发。

进入正式开发阶段后，正如上文所论述的，瀑布模型、快速原型模型、增量模型、螺旋模型、喷泉模型灵活性不高，在本平台上的项目需求和人员的变动相对较大，不适合使用这些开发模型；而rational过程和微软过程虽然比较全面，但是相对工作比较繁琐，所以平台推荐使用scrum敏捷开发过程模型，并为此提供了充分的作业空间。当然，用户也可以不使用scrum敏捷开发过程模型，充分使用平台提供的功能空间进行开发也是允许的。

在正式开发阶段，项目组需要先正式地第一次确定需求，分配工作并进行记录。讨论阶段，平台提供项目组专属的讨论组，并提供一定的存储空间存放记录的项目文档，需要注意的是，这些文档意味着正式的记录，是可以被每一个用户浏览的。

正如“基于 Scrum 的知识共享模型研究[2]”中提到的，在开发阶段会有计划会议、每日会议、评审会议、回顾会议等会议。在这些会议中，讨论部分都可以在讨论组中进行，而重要的决策需要记录员写入文档后存储在此项目的文件库中，此文件库如上文所说是可以被全体用户查看的，也是这个创意共享的关键。

作为一个共享想法、协同实现的平台，idea的内容采用了cc by-sa 3.0 的思想，将想法的版权也共享开。但是共享并不意味着人人都可修改。每一次对文件库的修改操作只能由项目组人员进行。在项目结束后，项目组也将被收回对文件库的修改权限。

# 项目成熟度机制

项目的关注度（浏览量、评论量）+需求完善程度（是否硬件项目+硬件方案成熟度+软件方案成熟度）+项目组成熟度（项目组成员层次划分）

为了准确及时地满足用户的需求，一个周全的成熟度分析机制是必要的。在此平台，成熟度主要是为了帮助用户发现精品内容，然而我们无法对项目的执行过程进行评价，不过我们可以从三个方面来判断成熟度，一个关注度高的项目、一个需求考虑完善的项目或是一个项目组成员层次分明、分工明确的项目都可以看作是成熟度不错的项目。

实际上，以上三个方面也是本系统着重考虑的因素，由此我们将三个因素作为成熟度考量的三个指标：关注度，需求完善度和项目组成熟度。正如图一所示，三个指标是综合影响最后评分的，不过关联的系数比重可能不同，将于下面进行具体分析。

1. **项目关注度**

一个精品项目往往是具有较高的关注度的，这些项目也是值得向用户推荐的。从项目的浏览量和评论量可以简单有效地表现出项目的关注度，我们采用分级的策略进行项目的评分，按照浏览与评论数量的多少分成5到7级，即0-100，101-500，501-1000，1001-2000,2000+分成5个阶级，由在某个阶级获得相应的基础分，再根据数量获得一定的加分，加分后一定不会超过下一级的基础分，最后综合浏览量和评论量的分数来计算关注度的分数。由此策略对项目的浏览量和评论量进行评分得出项目的关注度。

但是同时不乏存在一些有着大量浏览量和评论量却没有实际实用内容的项目，因此同时还应该从以下的需求完善度和项目组成熟度两个指标判断整体的成熟度，而我们会适当降低关注度的占比，来达到合适的评分标准。

1. **项目需求完善程度**

一个项目的出现来源于一个需求的产生，一个项目要完善，自然就要考虑到是否有足够完善的需求方案。从最初的一个想法，一个需求开始，逐渐细化，到界面、功能的每一个细节，全部体现在需求方案中。我们无法判断对错，是否细致，我们能做的就是判断项目需求方案是否有足够的需求文档，文档内容是否丰富，同时根据游客浏览后给出的评价来进行判定需求完善程度。

1. **项目组成熟度**

项目组成熟度，关系到项目组成员的分工，但是如何对分工进行评分？不能根据人员的多少来确定分工是否合理，那么就只能根据成员素质决定，那么如何判定素质？

我们决定加入等级制度，用户在平台上参与的活动获取积分，根据积分获得等级。同时加入分类标签制度，将用户所擅长的技能分门别类，作为用户的个人标签，在项目分工时也更方便，而且标签也同样有新手到大牛的等级划分。若在CSDN等较为出名的网站上有较高的知名度，平台可以直接赋予其一定的积分和高等级标签。

加入等级制度之后，通过项目组成员等级来判定项目组的成熟度显得简单许多。一个项目启动之后，要根据需求对项目组成员进行分工，平台根据分工是否达到详细的层次划分给予一个基础分，在根据项目组成员的等级和标签给予加分，最终得出项目组成熟度分数。

# 基于成熟度评分的管理过程

在系统根据算法计算出项目对应的成熟度之后，平台就可以将用户更想看到的项目呈现出来。

首先，还在最初想法形成阶段的想法不具有成熟度，仅有关注度。平台单独列出一页对热门和最新的想法进行排序。

然后，对于已经由想法转变为项目的项目，则开始计算成熟度评分，项目将会按照成熟度进行排序并在列表中从高到低地展示给用户。项目也会划分为三个模块进行排序：热门、精品和最新，热门和精选由于成熟度的三个指标的占比不一样，排序结果也不一样。热门是比较受大多数用户关注的项目，则关注度占比会较高；精选偏向展示优质的项目，则需求完善度和项目组成熟度占比会较高。最新则顾名思义是最新出炉的项目。

最后，针对一些难免漏洞和盲点出现的劣质的项目，我们只能采取举报投诉的措施来制止这种现象。

## 引用

[1] GitHub:全民一起“打怪升级”

[2] 使用Github实现高效的团队协作开发

[3] 众包技术研究综述

[4] <https://zh.wikipedia.org/wiki/ReCAPTCHA>

[5] <https://zh.wikipedia.org/wiki/captcha>

[6] Lillux内核开发者协作模式研究

[7] KroahHartman G， Corbet J。 McPherson A． Linux kernel development[J]． the Linux foundat ion． 2008．

[8]

[9]