# 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。

# 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2016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。

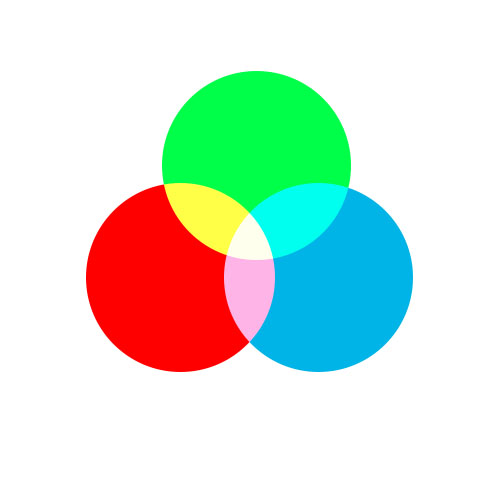
作为一个共享想法、协同实现的平台，idea的内容采用了cc by-sa 3.0 的思想，将想法的版权也共享开。但是共享并不意味着人人都可修改。每一次对文件库的修改操作只能由项目组人员进行。在项目结束后，项目组也将被收回对文件库的修改权限。

# 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，一个关注度高的项目、一个需求考虑完善的项目亦或是一个项目组成员层次分明、分工明确的项目都可以看做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，正如图一所示，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。



（图1，三原色图）

正如其名，成熟度至少标志了一个项目是否“成熟”，也就是进度。进度愈趋近于完成，那么一个项目的成熟度应该是更加的高；反之，更加低。为了描述进度这一可以感知却不可具体描述的指标，算法引入了需求完善度和项目组完善度两个重要指标。

一个项目要完善，就要综合讨论它将遇到的各种情形，以及预测可能出现的bug，为了预防这些错误，也就意味着项目组在项目早期就应该有一个比较完善的需求列表，同时也就意味着更多的需求。所以，更多的需求往往代表了项目更加的完善。

除了需求完善度，还有项目组分工的完善度。当项目组层次分明时，那就说明项目组已经有了一个明确的开发计划，那么正式开发也就是下一步行动；即使一个项目完成了，但是相关的项目组分工并不明确，那么这种开发模式并不受系统推荐，相应的项目组成熟度也应该低一点。

除了这些项目本身表现出来的成熟度指标，一个精品项目往往是具有高关注度的，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。但是作为一个可操作的指标，项目的关注度可能存在泡沫成分，甚至上文提到项目需求也存在此问题，考虑到这些情况，在算法中会做特殊处理，当数据太离谱时，成熟度分数的上升将不再明显。

值得注意的是，以上部分数据可能会超出计算的范围，如上千条的评论，这种情况是有可能超过阈值，是成熟度以超过满分的形式存在，这是不被允许的。所以，在这些项目，我们会进行特殊的算法处理，尽量让它们不超过允许的范围。

# 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，项目将会按照成熟度进行排序并在列表中从高到低降序地展示给用户。

在基于成熟度评分的排序系统外，平台也提供基本的按时间展示最新项目、按热门展示评论和点赞最多的项目两种基础的排序系统给用户选择。

为了更加精准地向用户展示用户感兴趣的项目，在排序过程之后，平台向用户提供二级筛选，按照分类筛选出用户感兴趣的方向。

## 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]

[7]

[8]