# GitHub 灵活的工作制度

GitHub是一家源代码托管和发布站点，其主要作用是供开发者发布自己的源代码和实现开发者之间的交流与合作。2012年获得1亿美元的巨额融资，曾估值7.5亿美元。

Github 在代码托管领域是先行者，现在强力的竞争对手也有不少，包括 Gitlab（局域网部署）、Bitbucket（免费账号不限 private 项目个数）、GitCafe（对国内开发者来说可能有墙内优势）。

但 Github 从诞生之初，就有着另外两个非凡的使命： 

#### 开源社交平台



#### 企业项目管理平台

这两部分业务是最核心粘性最大的，也是 Github 的核心竞争力。代码可以下载下来再传到别处，企业版本控制的安全性和稳定性其他竞争对手也可以做的不错，但只有基于人的社交关系和项目知识经验积累是没办法转移到别处的。

开源社区这一点无需多说，Github从一开始就把开发者的身份、他的关注、他的代码贡献、讨论和活动时间线作为核心功能来做。每个开发者都有一个属于自己的 profile 页面，上面列出了参与和关注的各种项目，和最近一段时间的公开活动（时间线）。通过感兴趣的项目，可以关注感兴趣的人，通过感兴趣的人，又能了解到更多有趣的项目。在这个基础上，Github 成为全球开发者学习、合作、认识基友、甚至消遣（explore 更多项目）的最大社交平台。

接下来，Github 在各种开发者大会上，宣扬 Github 公司自产自销的工作流程哲学： How GitHub Uses GitHub to Build GitHub ，基于 Repo-Issue-PullRequest 的开发流程不光是为开源社区而生，也适合用来开发像 Github 这样的大型项目。具体什么内容我这里就不赘述了，链接里的演讲说的很清楚，github员工团队目前就是按照差不多的流程在执行，是团队成立以来 chemistry 最好、效率最高的项目管理实践方式。

1911年，弗里德里克·温斯罗·泰勒写下《科学管理原则》一书。在书中，他第一次用科学的方法实现了共一生产的效率最大原则。时间就是金钱、越快越好、耗时越多质量越高。

在许多行业中，工时是判断生产效率的一个重要方法，但是在代码界却不同。在互联网创业企业中工作的体验与在工厂中的体验截然不同，编程问题绝对不能靠多花时间来解决。同样，编程是一项创新性的行为。只有在程序员脑袋非常清楚的时候才能写出高质量的代码。

GitHub创办于2008年，被市场研究公司Forrester称作“开发者的Facebook”，有逾1000万名注册用户在协作开发2500万个软件项目。个人开发者利用GitHub存储、编写代码，并与其他开发者进行协作；企业和政府机构则利用它托管和管理软件项目。GitHub对不公开的软件项目收费。机构可以通过付费在内部托管GitHub，方便内部人员分享代码。GitHub其他服务则是免费的。北京时间2015年7月30日，开源代码托管网站GitHub从知名硅谷风险投资公司融资2.5亿美元。

## 灵活的工作制度

俗话说，时间就是金钱，速度越快越好。时间越多越好。

## 跟程序员算小时很扯

在很多工业企业中，时间是决定生产效率的一个主要因素，但对于GitHub不是。在一个创业公司中，你不可能在一个问题上投入更多的时间，以期得到彻底解决。代码才是努力的方向。你需要具有正确的思维模式，以便创造出高质量的代码。 回想最近一次令你印象深刻或生气的事。你的工作效率如何？回想一下你最近真正高效的一次经历。代码从指间如飞般产出。当你具有正确的思维模式时，某一天高效的代码编程，可以胜过你一周受挫的编程工作。

github员工希望员工尽可能在这一状态下工作。限定员工在办公室中办公的时间会影响他们的工作状态。如果要求我在9点之前必须到达办公室，我是很难保持这种高效状态的，但对于在GitHub的一半员工来说，上午可能是他们工作状态最好的时间。 允许员工更加灵活的工作时间，可以营造一个令员工兴奋的工作环境。在这个环境中，他们可以工作更长的时间，并一直保持高效的工作效率。

### 一天的工作

在GitHub，每个人的一天的工作时间安排是不同的。同样我的每天也是不同的，下面是一个大概的时间安排： 1.上午十点钟左右起床； 2.坐公共汽车去上班，并在中午或下午一点去吃午饭； 3.从下午一点找个地方工作，下午六点到九点在办公室里工作； 4.之后回家，并在家中沙发上工作或休息到早上两点。有时也会和同事出去吃饭。 有的同事可能在上午7点就来到办公室工作；也有的在下午3点才来。有的同事觉得在家工作比较高效。如果员工不喜欢在办公室工作，他们可以不用每天来公司（虽然大多数情况下，每个人都会来公司）。 github员工一天的工作时间为什么如此“松散”，原因有二，一是工作在宽松的环境中，可以使github员工在github员工喜欢的时间和地方工作；二是github员工希望创造一个可以使员工最高效率工作的工作环境。因为每个人高效工作的时间都不相同，所以github员工不会强迫任何一个人。 限制性工作 2013年GitHub有35名员工，并且还在增长中。这个方法带来了很好的效果。但是管理者仍喜欢固定的工作时间，原因有一：他们有一种错觉，认为时间是衡量员工工作的标准。 如果对工作时间很难把握，你需要寻找其他的衡量方法。他们的代码写得很出色吗？他们把Bug都处理完了吗？他们全身心地投入到工作中了吗？工作上更大的灵活性是否激发了他们工作的积极性？ 对此很难做出定性的判断，但是相比起“在工作日的十小时内把这件事做好”，上面这些方法更有价值。因为当你用时间来衡量工作时，他们的工作就会变成更多的时间编写更少的代码。

## GitHub的异步工作模式

GitHub所有事情都是在异步工作模式下完成的。 GitHub创办之处的两年里都没有自己的办公室，工作都是在Campfire聊天室内合作完成的。现在已经进入了第二个办公室，但依旧在用Campfire聊天室。为什么对Campfire依依不舍？因为聊天室是异步的。异步交流指的是，我可以跑出去吃个午饭再回来看看漏掉的聊天内容，我可以向一个同事提问但是不要求他速速回答也不担心他会漏掉信息，我可以随时跑回家工作不担心错过办公室内发生的事情。 GitHub的绝大部分开发流程都要使用Pull Requests（GitHub的一种讨论方式）。如果我想在代码库中加入一个新功能或者做些改变，我就利用Pull Request创建一个请求并发送给所有同事，他们可以查看这一请求，然后根据以下三点做出选择：1、这一变动是否影响到其他人的代码；2、对这个主题是凑感兴趣；3、是否有足够的时间来参与。完成申请之后，github员工再在不同的电脑上进行测试，如果可行，就上传到主服务器中提供个网友使用。 因为Pull Request的存在，我不会要求所有人都参与到某一活动中来，只要那些对这个功能感兴趣有时间的员工来开会就够了。对了，关于开会，我想说：开会真他娘的烦人！

### 开会真烦人

只有在非常紧急的情况下，GitHub才会开会。因为每次开会一定会有一些不相干的员工来参与，即使你对会议的主题感兴趣，你也会因为开会把你从之前的工作中拉出来又去讨论还没开始的内容而感到沮丧。 另外，开会的内容也让人记不住。即使你做笔记，也无法几下所有的内容。当下你可能还记得没记下的内容是什么，但是几周之后，再看着笔记便会茫然无措。如果有会议纪要就不会出现这样的问题。不让员工乱想，而是让他们看打印的句子，更有助于集中精神进行思考。我用十个指头都能数出最近一年半开过几次会。

### 喝喝酒，好朋友

GitHub花大力气创造创新环境，鼓励员工去做一些花边的内容，这可以让员工对自己的工作感到兴奋。兴奋是可以传染的，一个工作小组的工作状态可以影响到其他小组。即使花边内容不能盈利，但是兴奋的工作状态可以为GitHub带来更多钱。 好吧，GitHub中有很多“酒鬼”。办公室里随时都有啤酒供应，但是喝酒并不意味着喝醉，酒的意义大于喝酒。 喝酒的时候可以遇到许多人。有时github员工在洛杉矶喝酒的时候就遇到来自世界各地的人们，github员工跟他们介绍GitHub，让人们了解GitHub。（酒后氛围缓和，不会对任何人构成威胁，介绍产品更安全）

喝酒的时候可以了解同事。他们在一起工作，但关系不仅限于同事，他们还是朋友，那种上刀山下火海的朋友关系。喝着啤酒，聊天、谈工作，界限非常模糊。关系亲近之后，无话不谈，有什么想法就直接说，帮助他们帮助自己。所以说，究竟是社交润滑剂，是有一定道理的。 喝酒的时候可以进行头脑风暴。酒吧就是一个适合讨论“如果”、“假设”的地方。GitHub的许多功能都是在酒吧里头脑风暴出来的。

### 鼓励不同

GitHub员工喜欢用Ruby编程，但是公司并不限制他们用什么编程软件。GitHub员工每月都有免费的健身课程，还从哲学角度讨论Twitter粉丝谁更多。github员工鼓励员工进行讨论，不管是编程中的难题，还是下一次郊游地点。让每个员工都有不同的生活，让他们对工作兴奋，让他们拥有不同的思维方式，既提高了自身，又助力了企业。

### 创新意味着自我引导

在GitHub，自我引导非常重要。如果你对一件事情非常感兴趣，那就随着兴趣去做。许多伟大的作品往往都是出自对它有热情的人之手。 在GitHub，每个人都有机会参与讨论产品。Pull Request也给了一个非常好的平台，让github员工可以尝试新想法，避免哗众取宠。如果你真的对有些东西感兴趣，就放手去做吧。GitHub会随着你的兴趣帮助你实现这一产品。 如果有这么家任性的公司，没有所谓"经理人"这一层，人都在做自己喜欢的事情，并且创造价值，而其他的事情，就顺其自然让他发生。这里能节省多少官僚主义带来的浪费？这样的公司得跑得有多快？得有多少无谓的冲突消解于无形？能形成多惊人的创新文化啊？

GitHub就是一家尝试无线趋近这种理想主义的公司！程序猿（媛）们应该都知道GitHub，GitHub是一个开发员的协作平台，截至2014年统计，有超过340万用户，而且正在迅速商业化。

GitHub是一家建立在开源软件上的私营的盈利公司，但是整个运作过程中，他们都在实践开源协作的精神——整个公司都倡导志愿工作。还是不明白到底怎么玩的？且听我给你解释。

## 开放式工作分配

大多数公司工作都是这么安排的：公司高层想到一个项目，于是经过立项，财务预算，及人员调配之后，自顶向下把项目分配到各个团队和业务单元。在GitHub可没这样，他们的人采用"开放式工作分配"的方式——人自行解决自己项目分配的问题，他们把自己分配到自己想做的项目上去，用不着任何正是的申请或者管理层干预。

您得吐槽了，GitHub才多少人啊？就175个员工，当然玩的转这种方式了。不过您瞧瞧Valve软，他们有400员工，而公司毛收入达100亿美元，Valve的组织形式跟GitHub非常接近。这总可以说明点什么事吧。

## 公司的结构？

古时候刀耕火种的人们就在思考，用自己过剩的产能能做点啥第二第三产业？比如，打猎之余，种点庄稼，缝点衣服啥的拿到集上去卖。有人做过研究了，那些过剩的产能中，最有可能造成影响的，是人脑子里面那些过剩的创造力！

于是Google著名的20%的时间可以做点业余项目。有个叫克莱·舍基的人，写了一本书，书名叫《认知盈余-网络时代的创造与繁荣》，他有个观点，闲暇时间给人机会创造有价值的东西。

直观上，这简直是胡说，就把人放到那里，从头到尾做他自己想做的事情，然后他就自然而然创造价值了？您心里估计在犯嘀咕，要是把那谁谁谁放到这里，她估计得7\*24小时逛淘宝。

不过，事实证明，这样的工作条件，对于那些有创造力的人确实有诱惑。比如说，有个叫Chrissie Brodigan的UX研究员，她有套测试方法叫Deprivation测试，以前是做火狐的用户体验评估的，就决定加入GitHub了。

GitHub的员工反馈，说他们被开放式工作分配这种方式吸引，主要还是因为它没有什么边界——这些人喜欢自由选择跟自己气场相投的人一起做一些很酷的东西。他们甚至可以一起设计自己的可编程办公室。

## 创新的四大难题

有人得说了，这玩意真费事，搞这么乱干嘛，我才懒得管那些呢，公司告诉我做啥，我给他保质保量做好不就行了？ github员工得说，这个论点在有些行业当然是行得通的，比如前些年，在广东，浙江一带，有很多制衣厂，对于一个有资深制衣行业背景的人而言，如果你有明确的目标，比如目标人群，衣服的款型，销往哪个区域，这些能确定之后，他就能告诉你需要投多少钱，建多大规模的流水线，招什么样的技工，甚至每个环节的技工该做什么。 不幸的是，今天，公司如果需要存活相当长一段时间的话，就需要面对未知的创新：如何创新？如何让这样的创新可以反复发生？传统公司会建立一个独立的"创新部"，这里的人负责创新，而其他的人更像在制衣厂工作。GitHub则不一样，他们把全公司都设立在一种以创新为核心的工作模式上的。这么做，是考虑到了让一批人能够持续不断成功创新的四大难题：

#### 管理注意力:

一般组织的设计，主要目的是基于已有的成功，一方面坐享其成，另一方面保护既有成功，而不会专注于开拓新点子。

#### 步调一致:

一般创新点往往来源于个人或者小团体，而执行往往要倾尽全公司之力去做。而由于种种原因（比如政治，社会因素），让剩下一大群人都买账很难。

#### 让人共同工作:

当新产品和服务开始结果了，人、想法、交易都会快速增长，于是要招更多的人。这样一方面人就更容易掺和进来做事，同时也更容易导致一些人只见树木不见森林，对愿景和当前的情况没有整体把握。

#### 僵化的管理:

有些创新着实是需要组织结构和管理方式能够从根本上改变的，而大多数公司的基础结构没这么灵活——这直接导致了组织结构难以跟创新保持同步。

### GitHub的结构到底是怎么样的？

沟通！沟通！沟通正是开放式工作分配的核心要素。换句话说，GItHub公司的结构就像是一张分布式的信息传递网络，网络上传递的是公司的目标，以及公司如何工作，当然也包括它应该要做些什么。

创新公司的策略是不断调整的，而对于某一时期这些调整增量，让所有人的认知保持同步至关重要，否则有些微小的调整很难到达某些单元或者团队，当有些团队没有做出相应的调整，而最终累积到一定的时候，不同团队之间的就形成了鸿沟般的认知差距。你得说了，这没什么啊，我看到的公司都是这样的。举个例子，上个世纪80年代的时候，乔布斯带着一帮人，做成了比较成功的Macintosh，另一支团队则哼哧哼哧鼓捣出来Apple III，Apple III大家都懂的。

良好的沟通，也可以帮助团队发现一些新的，或者相关的商机。举个例子，在这张网络里面，如果有一个新项目很酷，在网络里面传导到公司的高层了，高管们就可以利用这些信息修正航向，而修正的结果同时又通过网络传到各个单元，让他们明白，"某某项目"现在开始帮助实现公司的商业了。

这怎么玩得起来？

#### 1. 让人自己选择自己感兴趣的事情

GitHub首先得弄清楚，自己作为一家公司，到底什么是最重要的，并把这些体现在公司的策略上。然后每个团队都应该做类似的事情，"这是公司的策略，这是github员工的产品，那对github员工团队，什么最重要？"，销售，技术支持，运维等团队都做了这样的事情，他们把自己团队的目标和公司战略目标做了匹配。而前面说到的信息网络，让这种组织形式更加高效。

团队找到了工作重点，个人也要找自己的兴趣点，探索自己的兴趣跟这些策略的契合点，并把自己分配到那些能贡献的重要的事情上面去。所以，对个人来说，只要是对公司重要的事情，自己又有强烈的兴趣，就可以决定去做了。如果运转得好，所有人都在重要的事情上工作，不需要有人一层一层分配。这就是所谓的开放式工作分配方式(Open Allocation Approach)。

举个GitHub的例子，有段时间3D打印机很火，有个叫Mike Skalnik的在公司买了台3D打印机之后就开始感兴趣了，他慢慢的从原来的项目里面抽出经历来鼓捣3D打印机。一开始的时候，天知道3D打印机跟一个开源项目代码协作平台有什么关系？而Mike渐渐鼓捣明白3D打印机了，后来他发现，可以用协作的方式跟其他人共同维护3D打印的模型文件，而这正好又跟GitHub的方向完全吻合的。

Mike后来就抽出越来越多的时间去尝试3D打印，到后来他成了3D打印项目的带头人。

#### 2. 让更接近问题的人有决策权

GitHub尝试把决策权移交给更加接近真实问题的人，在GitHub的项目中，每个项目都有一个主要负责人。

你得说了，当然，github员工团队每个团队也有个产品负责人啊。问题是，如何产生这样的人和这样的团队呢？一般公司，都有一个中高层经理，他把自己手底下的人分成小的团队，让他们每个团队负责一个区域，而每个区域，又有一个直线经理负责人员的调配，于是这个直线经理指派了一个人，说："你，就你了，作为产品负责人负责这个产品"。于是产品负责人像一个牵线玩偶一样负责这一摊产品。这里有个巨大的问题，在于位于顶端的话事人，他得要预测一个比较成功的愿景，来告诉大家产品或者公司的方向。

在开放式工作调配的方式中，前面这个过程基本上可以说是反过来的。GitHub认为，最接近待解决的问题的人应该是对问题最了解的人，他们很大可能性是最知道如何解决这类问题的。GitHub让这类人有这样的决策权。

#### 3. 招那些喜欢这种方式的人

我知道，还是有质疑的说，这不就是一片混乱吗？如何避免有人每天只逛淘宝？github员工再回到前面说的，如果要有人花时间和精力去监控这些事情，或者搞办公室政治，那就说明要么是组织，要么是个人出问题了。

很重要的一点是选合适的人，把他们引入团队。何谓合适的人？这些人应该本性就是积极主动，对技术和自己的事业一腔热情的人。这些成年人做的事情最终必将给给公司带来收益。当然肯定是有前提的，一般没有人说，"好，我有热情，今天我花一天时间给同事们烤面包吃。"（逢年过节可能有例外），每个人都深信，自己做的事情一定会在某种程度上给推进公司往前走。

#### 题外话

世界真的是平的吗？

有一段时间业界都在纠结，要把自己的组织打造成一个平的组织结构，其实大可不必纠结于多平才算是平，而是要把组织里面的各个点连成网，让信息能够自由流动。

#### 开源精神

开源精神是说，一个人发现别人做的有个东西很酷，然后就给他们发了个消息，告诉他们，自己对这事有兴趣，希望能够贡献自己的力量，接下来项目的热就告诉这位热心的观众什么东西比较重要，后者再决定自己如何贡献。GitHub的工作模式也是这样的，也就是前面说的开放式工作分配。

#### 关于透明

有人做组织转型的时候很纠结透明这件事情，而GitHub在倡导透明的同时，也留给团队和个人一些个人空间，如果团队希望暂时保留目前正在做的事情，到适当时机再公开，这样的决定也会被尊重的。

引用： 作者：王楠 链接：https://www.zhihu.com/question/28976652/answer/42780910 王楠游戏开发话题优秀回答者 管理一个纯 github https://www.zhihu.com/question/28976652 原文链接：How GitHub Works: Hours are Bullshit