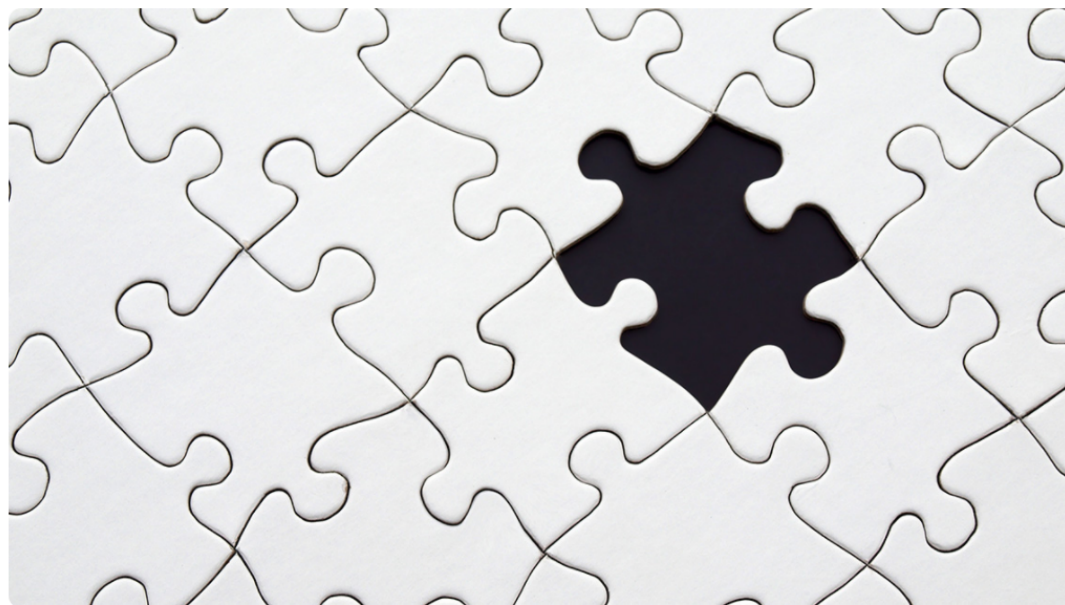


11 | 向埃隆·马斯克学习任务分解

郑晔 2019-01-23



00:00

讲述：郑晔 大小：12.08M 时长：13:11

1.0x

你好，我是郑晔。

这次我们从一个宏大的话题开始：银河系中存在多少与我们相近的文明。我想，即便这个专栏的读者主力是程序员这个平均智商极高的群体，在面对这样一个问题时，大多数人也不知道从何入手。

我来做一个科普，给大家介绍一下德雷克公式，这是美国天文学家法兰克·德雷克（Frank Drake）于 1960 年代提出的一个公式，用来推测“可能与我们接触的银河系内外星球高等文明的数量”。

下面，我要放出德雷克公式了，看不懂一点都不重要，反正我也不打算讲解其中的细节，我们一起来感受一下。

$$N = R^* \times F_p \times n_e \times f_l \times f_i \times f_c \times L$$

其中

N 代表银河系内可能与我们通讯的文明数量

R^* 代表银河内恒星形成的速率

F_p 代表恒星有行星的可能性

n_e 代表位于适居带内的行星的平均数

f_l 代表以上行星发展出生命的可能性

f_i 代表演化出高智生物的可能性

f_c 代表该高智生命能够进行通讯的可能性

L 代表该高智文明的预期寿命

不知道你看了德雷克公式做何感想，但对于科学家们来说，德雷克公式最大的作用在于：它将一个原本毫无头绪的问题分解了，分成若干个可以尝试回答的问题。

随着观测手段的进步，我们对宇宙的了解越来越多，公式中大多数数值，都可以得到一个可以估算的答案。有了这些因子，人们就可以估算出银河系内可以与我们通信的文明数量。

虽然不同的估算结果会造成很大的差异，而且我们迄今为止也没能找到一个可以联系的外星文明，但这个公式给了我们一个方向，一个尝试解决问题的手段。

好吧，我并不打算将这个专栏变成一个科普专栏，之所以在这讲解德雷克公式，因为它体现了一个重要的思想：任务分解。

通过任务分解，一个原本复杂的问题，甚至看起来没有头绪的问题，逐渐有了一个通向答案

的方向。而“任务分解”就是我们专栏第二模块的主题。

马斯克的任务分解

如果大家对德雷克公式有些陌生，我们再来看一个 IT 人怎样用任务分解的思路解决问题。

我们都知道埃隆·马斯克（Elon Musk），他既是电动汽车公司特斯拉（Tesla）的创始人，同时还创建了太空探索公司 SpaceX。SpaceX 有一个目标是，送 100 万人上火星。

美国政府曾经算过一笔账，把一个人送上火星，以现有技术是可实现的，需要花多少钱呢？答案是 100 亿美金。如果照此计算，实现马斯克的目标，送 100 万人上火星就要 1 万亿亿。这是什么概念呢？这笔钱相当于美国 500 年的 GDP，实在太贵了，贵到连美国政府都无法负担。

马斯克怎么解决这个问题呢？他的目标变了，他准备把人均费用降到 50 万美元，也就是一个想移民的人，把地球房子卖了能够凑出的钱。原来需要 100 亿美金，现在要降到 50 万美金，需要降低 2 万倍。

当然，降低 2 万倍依然是一个听起来很遥远的目标。所以，我们关注的重点来了：马斯克的第二步是，把 2 万分解成 $20 \times 10 \times 100$ 。这是一道简单的数学题，也是马斯克三个重点的努力方向。

先看“20”：现在的火星飞船一次只能承载 5 个人，马斯克的打算是，把火箭造大一点，一次坐 100 人，这样，就等于把成本降低 20 倍。如果你关注新闻的话，会发现 SpaceX 确实在进行这方面的尝试，

再来看“10”：马斯克认为自己是私营公司，效率高，成本可以降到十分之一。他们也正在向这个方向努力，SpaceX 的成本目前已经降到了同行的五分之一。

最后的“100”是什么呢？就是回收可重复使用的火箭。如果这个目标能实现，发射火箭的成本就只是燃料成本了。这也就是我们频频看到的 SpaceX 试飞火箭新闻的原因。

这么算下来，你是不是觉得，马斯克的目标不像最开始听到的那样不靠谱了呢？**正是通过将**

宏大目标进行任务分解，马斯克才能将一个看似不着边际的目标向前推进。

软件开发的任务分解

好了，和大家分享这两个例子只是为了热热身，说明人类解决问题的方案是差不多的。当一个复杂问题摆在面前时，我们解决问题的一个主要思路是分而治之。

一个大问题，我们都很难给出答案，但回答小问题却是我们擅长的。所以，当我们学会将问题分解，就相当于朝着问题的解决迈进了一大步。

我们最熟悉的分而治之的例子，应该是将这个理念用在算法上，比如归并排序。将待排序的元素分成大小基本相同的两个子集，然后，分别将两个子集排序，最后将两个排好序的子集合并到一起。

一说到技术，大家就觉得踏实了许多，原来无论是外星人搜寻，还是大名鼎鼎的马斯克太空探索计划，解决问题时用到的思路都是大同小异啊！确实是这样。

那么，用这种思路解决问题的难点是什么呢？给出一个可执行的分解。

在前面两个例子里面，最初听到要解决的问题时，估计你和我一样，是一脸懵的。但一旦知道了分解的结果，立即会有一种“柳暗花明又一村”的感觉。你会想，我要是想到了这个答案，我也能做一个 SpaceX 出来。

但说到归并排序的时候，你的心里可能会有一丝不屑，这是一个学生级别的问题，甚至不值得你为此费脑子思考。因为归并排序你已经知道了答案，所以，你会下意识地低估它。

任务分解就是这样一个有趣的思想，一旦分解的结果出来，到了可执行的步骤，接下来的工作，即便不是一马平川，也是比原来顺畅很多，因为问题的规模小了。

在日常工作中，我们会遇到很多问题，既不像前两个问题那样宏大，也不像归并排序那样小，但很多时候，我们却忘记了将任务分解这个理念运用其中，给工作带来很多麻烦。

举一个例子，有一个关于程序员的经典段子：这个工作已经做完了 80%，剩下的 20% 还要用和前面的一样时间。

为什么我们的估算差别如此之大，很重要的一个原因就在于没有很好地分解任务，所以，我们并不知道要做的事情到底有多少。

前面我们在 [🔗 “为什么说做事之前要先进行推演？”](#) 文章中，讲到沙盘推演，这也是一个很好的例子，推演的过程就是一个任务分解的过程。上手就做，多半的结果都是丢三落四。你会发现，真正把工作完全做好，你落掉的工作也都要做，无论早晚。

与很多实践相反，任务分解是一个知难行易的过程。 知道怎么分解是困难的，一旦知道了，行动反而要相对来说容易一些。

在“任务分解”这个主题下，我还会给你介绍一些实践，让你知道，这些最佳实践的背后思想就是任务分解。如果你不了解这些实践，你也需要知道，在更多的场景下，先分解任务再去做事情是个好办法。

也许你会说，任务分解并不难于理解，我在解决问题的过程中也是先做任务分解的，但“依然过不好这一生。”这就要提到我前面所说难点中，很多人可能忽略的部分：可执行。

可执行对于每个人的含义是不同的，对于马斯克而言，他把 2 万分解成 $20 \times 10 \times 100$ ，剩下的事情对他来说就是可执行的，但如果你在 SpaceX 工作，你就必须回答每个部分究竟是怎样执行的。

同样，假设我们做一个 Web 页面，如果你是一个经验丰富的前端工程师，你甚至可能认为这个任务不需要分解，顶多就是再多一个获取网页资源的任务。

而我如果是一个新手，我就得把任务分解成：根据内容编写 HTML；根据页面原型编写页面样式；根据交互效果编写页面逻辑等几个步骤。

不同的可执行定义差别在于，你是否能清楚地知道这个问题该如何解决。

对于马斯克来说，他的解决方案可能是成立一个公司，找到这方面的专家帮助他实现。对你的日常工作来说，你要清楚具体每一步要做的事情，如果不能，说明任务还需要进一步分解。

比如，你要把一个信息存起来，假设你们用的是关系型数据库，对大多数人来说，这个任务分解就到了可执行的程度。但如果你的项目选用了新型的数据库，比如图数据库，你的任务分解里可能要包含学习这个数据库的模型，然后还要根据模型设计存储方案。

不过，在实际工作中，大多数人都高估了自己可执行粒度，低估任务分解的程度。换句话说，如果你没做过任务分解的练习，你分解出来的大部分任务，粒度都会偏大。

只有能把任务拆分得非常小，你才能对自己的执行能力有一个更清楚地认识，真正的高手都是有很强的分解能力。这个差别就相当于，同样观察一个物品，你用的是眼睛，而高手用的是显微镜。在你看来，高手全是微操作。关于这个话题，后面我们再来细聊。

一旦任务分解得很小，调整也会变得很容易。很多人都在说计划赶不上变化，而真正的原因就是计划的粒度太大，没法调整。

从当年的瀑布模型到今天的迭代模型，实际上，就是缩减一次交付的粒度。几周调整一次计划，也就不存在“计划赶不上变化”的情况了，因为我的计划也一直在变。

如今软件行业都在提倡拥抱变化，而任务分解是我们拥抱变化的前提。

总结时刻

我们从外星人探索和马斯克的火星探索入手，介绍了任务分解在人类社会诸多方面的应用，引出了分而治之这个人类面对复杂问题的基本解决方案。接着，我给你讲了这一思想在软件开发领域中的一个常见应用，分而治之的算法。

虽然我们很熟悉这一思想，但在日常工作中，我们却没有很好地应用它，这也使得大多数人的工作有很大改进空间。运用这一思想的难点在于，给出一个可执行的分解。

一方面，对复杂工作而言，给出一个分解是巨大的挑战；另一方面，面对日常工作，人们更

容易忽略的是，分解的任务要可执行。每个人对可执行的理解不同，只要你清楚地知道接下来的工作该怎么做，任务分解就可以告一段落。

大多数人对于可执行的粒度认识是不足的，低估了任务分解的程度，做到好的分解你需要达到“微操作”的程度。有了分解得很小的任务，我们就可以很容易完成一个开发循环，也就让计划调整成为了可能。软件行业在倡导拥抱变化，而任务分解是拥抱变化的前提。

如果今天的内容你只记住一件事，那么请记住：**动手做一个工作之前，请先对它进行任务分解。**

最后，我想请你回想一下，你在实际工作中，有哪些依靠任务分解的方式解决的问题呢？欢迎在留言区写下你的想法。

感谢阅读，如果你觉得这篇文章对你有帮助的话，也欢迎把它分享给你的朋友。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

该试读文章来自付费专栏《10x程序员工作法》，如需阅读全部文章，
请订阅文章所属专栏，**新人首单¥19.9**

立即订阅



胡瑶

由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。

精选留言(29)



大彬

我会的任务分解，不仅可执行，粒度还很细。比如说，我要修复一个rpc接口的bug。我会列出每个代码的修改点，要修改的测试，要增加的测试，合并到哪个分支，修改rpc文档，文档中有哪些点要修改。

每一步都非常容易执行，看起来每多少必要，但在我当前的工作环境特别有用：1) 事前思考，不会造成遗漏，2) 任务实施过程中经常被打断，比如，测试有疑问和你讨论，主管找你谈事，紧急会议来了，这种“硬中断”完全打破了节奏，而任务列表，让我知道清楚当前做了多少，该从哪一步继续。

作者回复: 很清楚的做法!

2019-01-25

💬 2

👍 44



pyhhou

实际工作中来看，如果对一个任务，或者说是一项技术不是特别了解的话，确实很难做细致的任务分解，往往就是列一个粗糙的大概计划，然后去执行，在执行的过程中就发现很多计划都存在问题，一个计划里面还涵盖了之前没有考虑到的细节，导致任务项目充满了不确定性。很想听听老师的意见，就是在一个自己不熟悉的，充满未知的项目中该怎么更好地进行任务分解，还有这种分解的思想在平时是否可以通过一些练习和思考来加强，并应用到广泛的工作学习中去，期待~

作者回复: 好问题，我就在答疑里谈谈这个问题吧!

2019-01-23

💬 1

👍 31



Geek_kevin

任务分解和敏捷开发的用user story应该是相似的,首先我们会定义大的feature,这个是大的产品经理关注的,然后我们基于feature分解成不同的user story,最后每个story,再分解成一个个具体的task,我们程

序员就主要解决task。

作者回复: 喂, 110吗? 有人知道的太多了。:)

2019-01-24



12



树根

如果清楚知道接下来怎么做, 任务分解就告一段落。其实清楚就是知道是否可执行, 如何执行, 越是知道每步细节越能把控全局。

但往往就是以为自己“清楚”, 才导致任务评估不准。特别对自己没做过的领域, 做沙盘推演, 以结果导向推导任务推进过程, 做好排坑。

另外一点受益的, 就是分解任务不单单是任务内容的分解。特别是文中提到的特斯拉通过建立公司引入专业人才, 或者图数据库需要学习的知识。真正分解是如何达成任务目标, 分解所需的步骤、资源、风险。

我现在缺的是系统分解目标, 清楚知道自己下一步要做什么, 需要哪些资源。

作者回复: 多练习, 做事之前先分解。

2019-01-23



4



Kăfkă²⁰²⁰

最近在做战略拆解, 都是一样的道理。战略飘在空中遥不可及, 要落地就必须拆解。比如说达成目标有哪几个方面可以努力, 各方面都需要做哪些事, 这是路径。这些路径里哪些优先级最高, 需要配置哪些组织资源。心里有数之后就是制订计划时间表。

作者回复: 你做的非常棒!

2019-01-23



4



北天魔狼

网站积分清算都是系统定时任务，老是担心时间超时任务失败。后来把任务按照时间拍好顺序，保证每个任务都不超过一分钟，运行时间间隔大于5分钟。再也没有出现过运行失败

作者回复: 不错的分享!

2019-01-23



3



木子輕颺

任务分解是一个很好的策略，其实日常做事情也在做任务分解。比如在家里做饭，需要开始准备做什么菜、整理材料、蒸饭、炒菜。根据每个人的理解不同，会有不同的步骤，而且每次都不太一样。这里默认的就已经有任务分解在里面了。

日常生活的自动化，或者称为习惯，大脑会在后台做任务分解反而降低了感知度。想要精细化的控制任务，就需要做可感知的任务分解。明确每一个步骤，做到可执行化。

对于一个大的任务，或者未遇见的任务大脑的默认方式就不起作用了。这时会体现出懵懵的感觉。这种情况应该会有策略在之后的课程中讲解吗？

作者回复: 大任务分解的方式就是一点点分解，但对于不确定的任务，我准备在答疑中，专门讨论一次。

2019-01-25



2



liu

任务分解至自己能够解决的程度，即达到分解任务的粒度。然后以此估算工作时间与工作量。面对个陌生的事务，逐渐将不可控转化为可控，直至最后全面掌握

2019-01-23



2



公号-技术夜未眠

云计算，大数据等底层实现思想都体现了分而治之的分解逻辑。

2019-01-23



👍 2



猫咪没有了魚.🐟

当测试工作任务很多的时候，时间紧迫的情况下，一般会对复杂的测试任务，困难进行的测试任务进行分解成细小，容易推动实现的，一点一点逐步化解，有层次，从上到下，从主到次。

2020-11-30



👍 1



Jerry Wu

我看过一门公开课，里面谈到了一种编程风格：自顶向下。大概的思路是：先想清楚要做什么，再确定核心步骤，然后不断往下细分，直至完成。

这种编程风格，算是把任务分解用在写代码上了？

作者回复: 本质解决问题的思路都是先分解再逐一击破，写代码一样。

2020-04-21



👍 1



javaadu

老师说任务分解知难行义，我倒是觉得知易行难，这个概念很多人知道，但是在实际操作中怎么分解、分解到多少粒度是很难做到刚刚好的，我的经验是：

1. 按照业务模块、业务类型、业务流程进行划分
2. 宁可足够细粒度，不能草草了事

2020-02-16



👍 1



春之绿野

前两天分解一个任务，第一步是写一个template，我想着写一个template算是可执行了吧，但写template主要是为了做成自动化，自动化的话还要考虑怎么自动更新，生成，问题多着呢，最后做着做着就变成来不及了也不自动化了，先做一个出来再说吧

作者回复: 先跑起来再优化。

2019-06-01

💬 2

👍 1



狼

之前做过要把好几个小项目换框架，它们之间又有相互的调用，感觉要动一个就得动整体，无从下手，最后下定决心做的时候，就是分解开，第一步先搭建一个空的能跑起来的项目，第二步将其中一个项目中的某个功能进行实现，比如登录，然后一步一步挪功能，等都挪完了，项目也就换完了，其中还有在具体功能的时候分解，比如登录的时候，第一步先导入包，第二步 进行配置，第三步实现简单的数据通信，第四步修改参数获取项目需要的用户数据，感觉真正做起来的时候，反而没有想象中那么难

作者回复: 你做得很棒！

2019-04-18

💬

👍 1



John Bull

任务分解，的确是知难行易，怎么去分解，如何正确的分解任务很关键，如果分解的不当，可能举步维艰，让每一步都存在很多风险。任务分解对于人生其实也一样，想实现个什么目标，不管是学习成长还是买房买车，如果有个明确的步骤，也会轻松很多。

2020-10-22

💬

👍



2021年薪资目标35K

技术来源于生活

2020-10-21

💬

👍



Laughing

假如你需要逆向了解一个东西，任务分解的思考方式同样也是适用的。

2020-08-25



sam

跟庖丁解牛相似，高手很擅长做任务分解。新手需要对整个事物有多角度的理解后才能做到。

作者回复: 是这个道理

2020-07-01



一个帅哥

任务分解后要可以执行，否则分解出来有什么用呢

作者回复: 非常同意这个观点！

2020-06-10



蓝士钦

最近在做一个系统的数据迁移，我把任务进行分解：具体要迁移哪些表，这些表的数据从哪里来，迁移的顺序，迁移需要跨部门的人共同完成，需要哪些人做什么。把这些都定义好，执行起来就井然有序。

作者回复: 你已经体会到分解的价值了。

2020-06-09



escray

这篇专栏的结构，真的是一个很好的演讲框架，从浩瀚太空中的外星文明，到马斯克的 SpaceX（刚刚实现载人航空），再到程序员日常工作的分而治之、任务分解。

回答小问题是我们擅长的，给出可执行的分解，就不那么容易了。如果采用测试驱动开发的模式，那么

分解成为可执行、可测试的小任务，就更难了。

我自己的问题同样是在与高估了自己的可执行粒度，如果把自己的工作任务划分到番茄（25分钟）不知道会不会更好。

一说到“微操作”就想到了星际争霸，当年玩的不好，主要原因就是微操作跟不上。

其实我觉的大多数程序员在日常工作中都会用到任务分解的想法去处理工作任务，但是主要的差别在于分解的粒度。因为能力水平不同，同样的任务对于不同水平的程序员来说，任务分解与执行可能大相径庭。一般来说，水平越高，就会分解的越精确（但是不一定越细）。

那么问题来了，如何提高任务分解能力，并随之提高技术能力。

我能想到的就是，养成好的工作习惯，每次记录自己的任务分解计划，以及执行情况，然后改进。

作者回复: 然后是练习，大量的练习。

2020-06-04



索旭东

任务分解，知难行易

2020-03-25



若镜O

“物有本末，事有终始，知所先后，则近道矣 ”《大学》 多谢老师了

2020-02-22



Purson

我在开发一个postgresql 有数据插入的时候，就同时写入链数据进行了简单的任务分解：1. 学习postgres；2、监察数据库变化；3、根据数据变化调用api写上链。在应用服务器迁移的时候进行任务分解，备份数据、更新公告、复制环境、移动数据、灰度测试、正式上线、关闭旧服务器。

作者回复: 这个分解很好, 但在执行过程中, 还需要进一步分解。

2020-02-20



潘pan

做一个需求是, 往往需要预估完成时间。但遇到的问题是: 无法正确的分解任务, 以及不知道任务难易程度和自己的解决工时, 就会有无法预估的情况。请问这个情况怎么处理

作者回复: 那就给限定一个时间, 把不确定的部分确定好, 然后再往后走。

2019-10-16



陈斯佳

任务分解其实也像是老师所说的, 事前忙, 而不是事后忙。做事之前, 先在脑袋里过一遍, 把任务拆解成一个个可执行的微操作, 之后就可以相对无脑的执行了。

2019-07-23



(^_^)

让我想起一类面试题, 譬如煎饼摊大妈的收入估算、估算北京一年出租出去的房子数量等等, 将一个原本毫无头绪的问题分解, 分成若干个可以尝试回答的问题

作者回复: 嗯, 是这个意思。

2019-02-12



號國技醬

打卡

2019-01-23





王维

个人认为，要做到精确的任务分解任务，在实际工作中比较困难。如果不能对细节了如指掌，如果不能对全局高屋建瓴，要精确的分解是不可能的。我的指导思想是，在一项任务开始之前，做粗略的任务分解，然后随着任务的进行，边做边完善。说的通俗点就是边走边看。其实不管是做技术，还是给自己定发展目标，都是一样，一开始给自己定长期目标，然后具体到当下，我们就分解目标，边走边计划，边走边看！

作者回复: 分解到什么程度取决于自己的把控能力，不清楚的部分不分解是一种风险。

2019-01-23

