

什么是掘金小册



开发者会消费各种各样的内容，例如资讯、问答讨论、经验文章、最佳实践、开源库、文档等等，每一种内容都有自己的消费场景和它背后的真实需求，掘金作为一个帮助开发者成长的社区一直专注于如何更好地满足开发者这些内容消费需求。

但是只有碎片化的内容是不足以完成读者系统性学习的需求的，尤其是无法帮助那些有明确需求并渴望尽快知道完整答案的读者。以“如何准备 BAT 校招面试？”这是一个可以找到成千上万文章的问题，作为一个即将毕业的软件学院的学生，也是一个明确的需求。可是，网络中无穷无尽的博客文章、知乎万赞回答、微信爆款头条是无法系统性地帮助一个年轻学生清清楚楚地找到这个问题答案的。

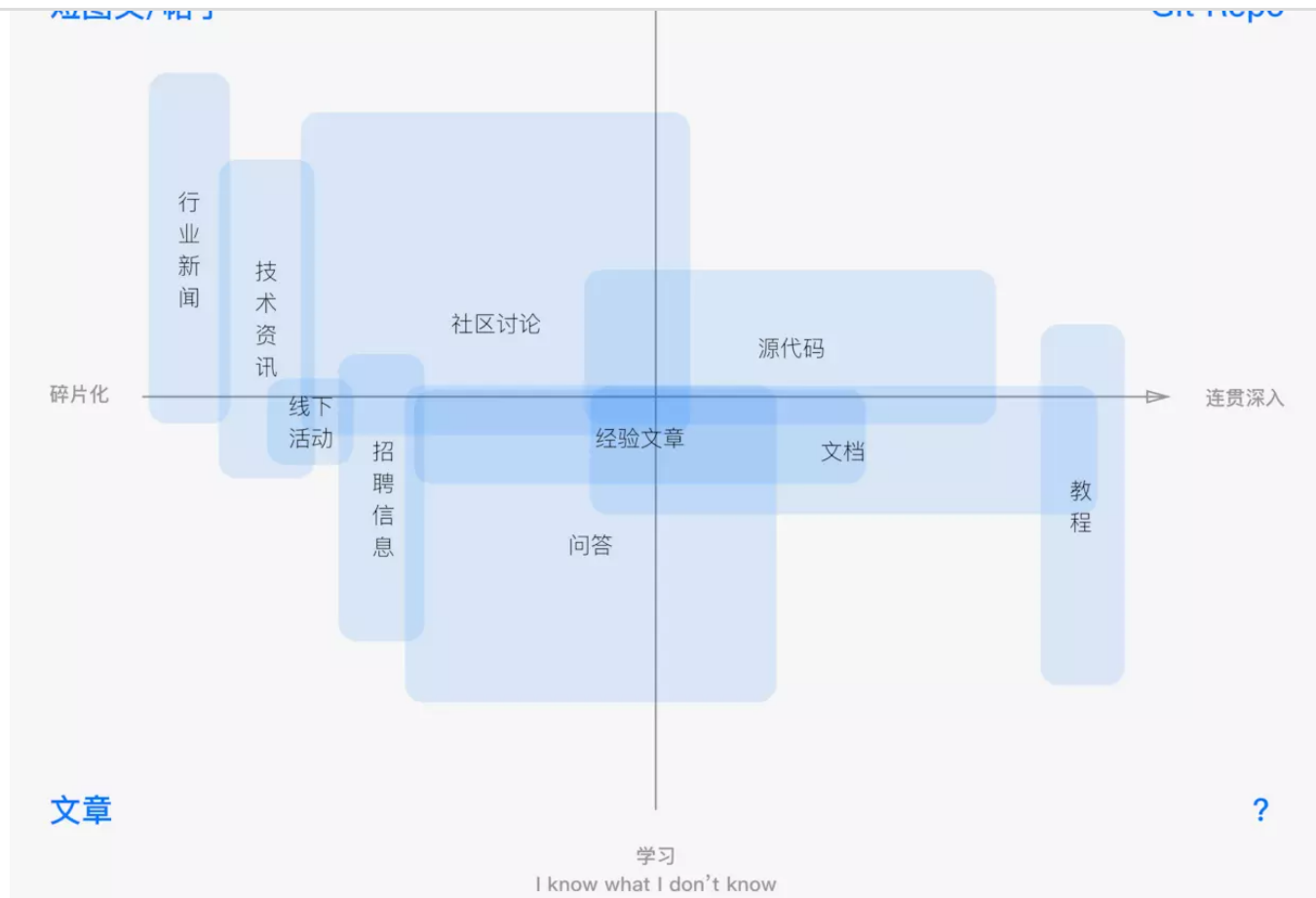
需要一本厚重大部头的书吗？似乎成本太高；一些散乱的文章呢？肯定不够；那怎样的内容才是我们想要的，才是可以帮助到读者的呢？而另外一端，是否有作者有能力写出这样的文章？他们是否得到了应有的财务回报呢？如果出版一本书籍是否时间成本过高？

基于这样的思考，我们想要通过一个产品去解决这个问题，而它就是：**掘金小册**

技术内容的载体

我们把开发者需要的内容如下排列：





从图中可以看到，我们把这些内容根据两个坐标轴来排列：

- X 轴：从碎片化内容到需要长时间连贯深入阅读的内容
- Y 轴：从我明确知道自己需要获取什么信息的学习需求，到非常不明确自己需要获取什么信息的探索需求

而开发者需要的各式各样的内容也根据它们不同的使用场景和内容形式被放到了不同的内容载体中去传播。

比如，行业新闻、资讯还有社区各种闲杂的讨论，用户每天去获取这样的信息往往是上下班、工作间隙的闲暇时间看一看，不带有任何的具体的目的一定要获得什么，只是期待相应的产品可以稳定的符合预期的得到自己想要的信息。可能是新的、有趣的、有用的等等。因而，类似于微博、资讯文章这样的内容载体就很适合来承载这部分内容。

又比如，作为业务开发者，我们需要看源代码，了解一个开源库的安装、使用方法、接口参数，甚至深入去了解它的最近的更新等等，而承接这么多内容需求的就是通过一个个 Git 仓库来实现的。

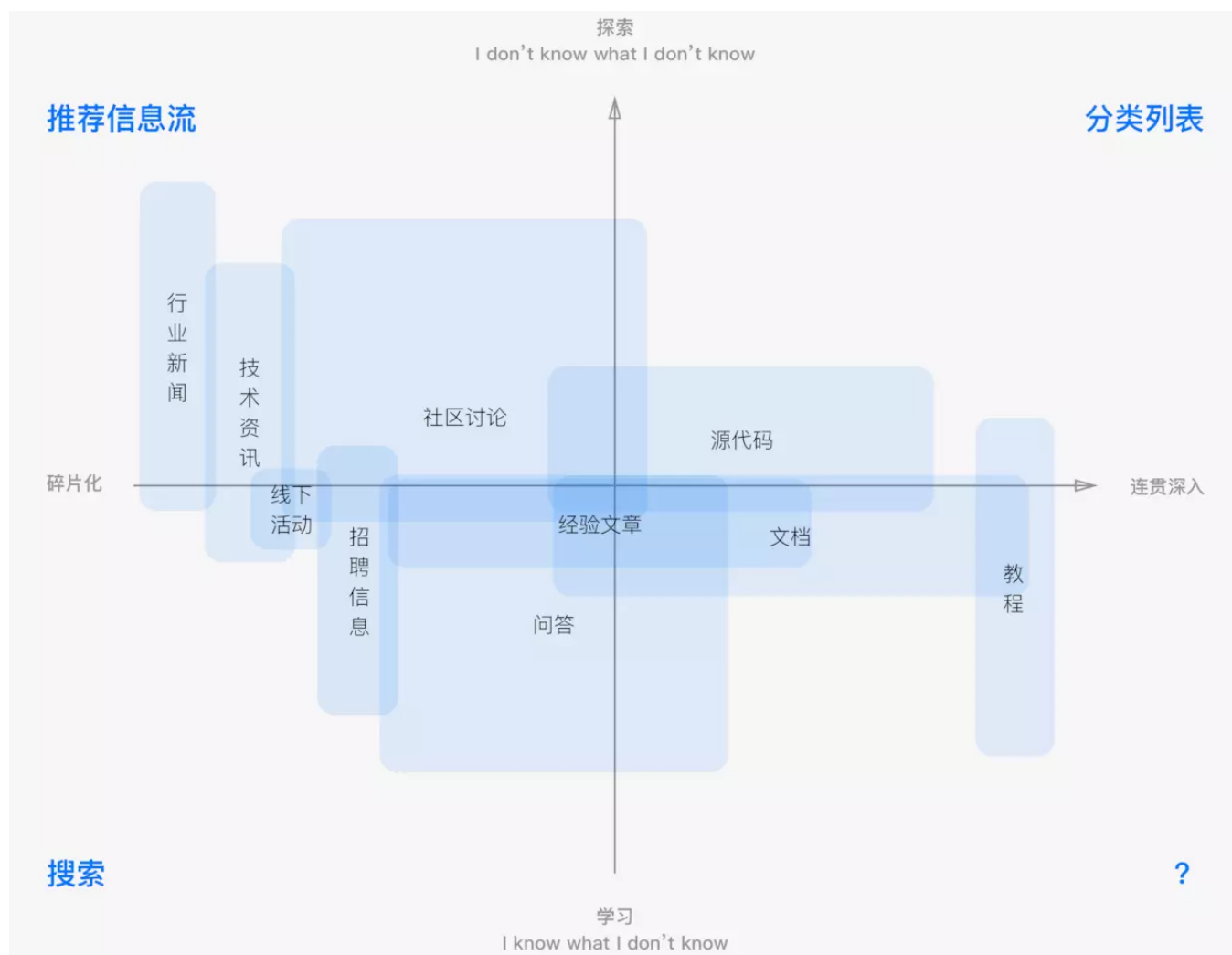
当然，所有开发者都需要连贯地、深入地并且带有明确目的性地去获取信息，而这样的内容载体会是什么呢？



如何写一本掘金小册

容，刷刷网站首页、搜索一下或是去到某个天汪的高手的页面、某个标签下去找内容。这一切的目的，都是希望帮助用户更方便的看到自己需要的东西。

那么，如果我们还使用之前的 X、Y 轴去分拆内容，那么不同种类的内容应该如何来分发呢：



借由这些不同的内容分发模式，用户可以在不同的场景下来消费内容。例如，早晨上班想看看最新的技术资讯、行业八卦或者是自己需要了解的经验文章，刷刷首页推荐的内容；写代码的时候需要找到某一个具体的库，在某一个编程语言的库下面一点点去找；遇到了解决不了的问题，去搜索一下问题的答案等等。

那么，当开发者持续连贯地学习某种需要深入理解的知识的时候，是怎样的分发模式在帮助读者一层一层地去持续地消费呢？

掘金小册

围绕着上述两个核心的问题：

1. 是什么内容载体可以去满足开发者明确的、系统性的内容需求？
2. 是什么内容分发模式可以满足开发者快速有效地获得这样的内容？



如何写一本掘金小册

1. 每一本小册是一本解决具体问题的成体系的内容，确保读者有问题可以在小册内容下得到完整的解答。
2. 小册之间应该有明确的体系关系，让读者根据自己的需求可以一个一个连续阅读消费，而这样的完整的学习路径可以成为读者选择下一个去消费的内容的主要依据。

举个例子：当我想要明确地学习如何用 Python 做一个爬虫系统的时候，掘金小册里可以找到它，但是这个内容可能需要你去了解 Python 的基本语法、也可能需要你去了解一些前端的基本知识，这一个个的具体的内容需求形成一个内容的网络，当这些小册交织成一个完善的内容网络，也就成为了一个需要深度学习内容的分发模式。

在这个核心功能需求之上，我们同样希望解放作者的劳动力，让作者专注于内容，提高生产内容的效率。只要能完整地解答小册核心的内容需求，作者不再需要长篇累牍地完成数百页的写作，因为里面大多数的内容可以在掘金的其他内容库中找到。而线上售卖的效率、作者更高的收入比例、读者作者间的社群关系，可以更好地帮助作者完善内容，获得应得收入。



小篇幅



高浓度



成体系



有收益

一本掘金小册，应该是**小篇幅**、**高浓度**、**成体系**、**有收益**的内容。既满足读者的学习需求、也满足作者的创作和收益需求，而个中效率的提升就依赖着掘金小册产品模式的优化。



每一本掘金小册都有专属的掘金设计的封面

特别值得一提的是，每一本小册都是作者的作品，掘金官方会为它单独设计一个漂亮的封面。而我们也期待有千百个小册被创作出来，成为一个完整的技术内容库。

评论将在后台进行审核，审核通过后对所有人可见

