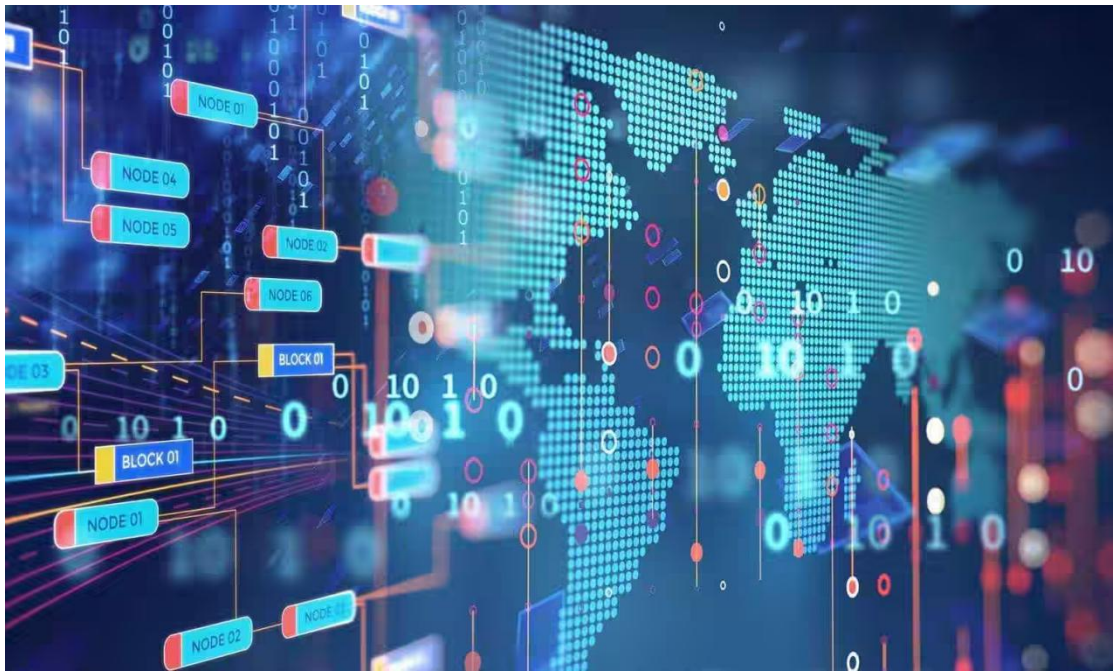


产品经理所需要掌握的技术点

首先强调一下，产品经理需要掌握一些技术点这事不是必备项，但却是大大的加分项，从我目前的经验来看，不懂技术的产品经理很难做产品，但可能有一些天马行空的想法，只不过大多数都会被同行或者技术开发人员说实现不了。



首先强调一下，产品经理需要掌握一些技术点这事不是必备项，但却是大大的加分项，从我目前的经验来看，不懂技术的产品经理很难做产品，但可能会有一些天马行空的想法，只不过大多数都会被同行或者技术开发人员说实现不了。有些产品经理是有研发背景的，即在转行做产品经理之前，有从事过开发工作，这样就非常的有优势，如果之前的开发与现在产品所需的技术语言是一样的话，就比较完美了，可以在设计的时候就进行一定程度的实现性和可行性考虑，

评估所设计的功能是否可以在现有条件和资源下实现，也能在开发人员的系统设计说明书评审会上听懂，可以大致了解是否符合要求。

技术开发人员都比较喜欢和同行交流的，就像我们自己喜欢和产品经理同行交流一样，因此做过开发的产品经理在和开发人员的沟通上有优势，但需要注意的是，千万不能不懂装懂，不要以为自己做过开发了不起，就指手画脚的参与系统设计，这样反而会令人反感，要记住你的技术背景只能停留在产品设计阶段和 PRD 沟通阶段，不要过多的给出技术方面的意见或建议，术业有专攻，况且你都转行了，说不定你所知道的东西已经过时了。

没有研发背景的产品经理就需要修炼了，其实也不需要去学习开发技术，但是要知道一些专业术语，比如要知道缓存、JS 脚本、Ajax、数据库、存储过程、BI 等等名词到底是什么东西，否则你会发现你在和开发人员沟通的时候会一愣一愣的，因为他们说的你听不懂。学习的时候要有针对性，比如公司产品都是采用 JAVA 开发的，那就去了解一下 JAVA 相关的基础知识，数据都是采用 MYSQL 的，那就去简单了解一下这个数据库相关的知识，我们的目标是能听懂开发人员说的话，以免陷于被动。如果让开发人员发现他说了半天，你都没有听明白，如果要他讲第二遍或者一一解释一下，估计首先会有点不耐烦，其次会有点嫌弃你了。

从我目前的经验来看，以下这些技术点是产品经理应该掌握的，这里不说“必须掌握”，确实是因为有例外存在。

SQL 的读和写

在大数据时代，产品经理几乎天天都要和数据打交道，数据的来源有很多，但大部分肯定都是存储在数据库里面的，这时要做分析数据的话，数据分析能力是体现在拿到数据之后的，前提是你要拿到数据，这时就和 SQL 有关系了。

大公司数据仓库建设比较完备，有较为完善的数据管理系统，且有专门的数据维护人员，俗称 BI 或者 DA，即做商业智能和数据处理加工的。在这种条件比较优越的公司，常规的取数需求都可以提交给这些专业人士处理，如日常的分析报表、项目开展所需要的基础数据等，只要说明清楚取数逻辑和所需要的字段就可以了。但平时的一些产品分析所需数据的获取，以及一些指标考核项的数据就需要你自己动手了。好一点的情况下，你可以央求 BI 给你写一个 SQL 语句出来，然后根据需要你自己改改；次一点的情况下，BI 会将一些表结构说明给到你，然后你自己去组织表关联取数；最差的情况就是啥都没有，你自己去取数系统里面摸索着取数。而在小公司，产品经理可以让开发人员帮你取数，但你要是频繁需要取数的，我想你也不好意思老开口，毕竟会影响开发人员的本职工作。自己动手丰衣足食，而且因为小公司数据系统建设不完善，最好的情况就是上面说的第二种，有表结构说明给你参考。

上面这些场景就要求产品经理需要懂 SQL 语言。这里首先是要能看懂 SQL，比方说你是求别人帮你取数的，但看了数据总感觉不对，这时你就需要去看看人家写的 SQL 对不对，毕竟你自己才最清楚取数需求是什么。看懂了发现有错误的地方，还要会改，特别是别人给你

一段有相似取数功能的 SQL 语句时，要能手动改改之后适合你自己的取数需求。其次是要知道怎么写 SQL，这就要求产品经理懂得 SQL 语言的语法和一些常用的函数，比如日期函数、格式转换函数、数学函数、字符串函数等等，最常见和最基本的都要掌握和灵活运用。

个人强烈建议产品经理同行们都学习一下 SQL，这样在日常工作当中会方便很多。需要注意的是，SQL 语言有 T-SQL (Transact-SQL) 和 PL/SQL (Procedural Language/SQL) 两种，需要根据不同的数据库类型，有针对性的去学习。比如说产品采用的是 SQLServer 数据库，那就需要学习 T-SQL；如果采用的是 Oracle 或者 Mysql 数据库，就需要学习 PL/SQL，两种语言在语法上差别比较大，大部分情况下都不能混用。另外 Oracle 和 Mysql 数据库在应用 PL/SQL 的时候，会有一些函数有使用上的差异，需要注意一下。有人会说现在还有一些非关系型数据库，如 MangoDB 之类的，这种数据库基本不支持 SQL 语言去查询，而且里面存储的都是非关系型的数据，也不需要查询出来做分析。

UML 和 E-R 图

早些年，产品经理这个名词还没有流行的时候，做类似行当的人一般称为“需求分析师”，传统的软件需求分析师的一项必备技能就是用 UML 画用例图，当然这个现在也适用，用例图对于说明需求来说，作用还是很明显的，至少开发能看的比较明白。现在随着思维导图和原型的兴起，用例图逐渐的有点被淘汰的意思，但很多开发人员还是

会用 UML 来画系统设计图，如活动图、状态图、协作图等，产品经理需要了解一下这些，以便能在设计评审上看懂这些图。

E-R 图也称实体-联系图(Entity Relationship Diagram)，提供了表示实体类型、属性和联系的方法，在数据表关系设计上应用比较多。一个产品的数据存储，肯定不会只有一张数据表，而是由多张数据表通过关系关联起来的，相互之间的这种关系就可以用 E-R 图来表示，可以从中看出各表之间关联的主键和外键分别是什么，哪些字段值唯一等。

UML 图和 E-R 图在系统设计当中经常出现，产品经理要能看懂其所表达的意思。你和开发人员讲需求设计的时候需要面面俱到，每个细节都会讲到，但开发人员跟你讲系统设计的时候，可是能简则简，你看不懂是你的事情，他们可是按这个开发的。

移动端的设计规范

这里包含产品设计规范和编码规范，无论是 Android 还是 IOS 都有自己特定的规范，移动端的产品经理在设计产品的时候就需要先了解清楚这些规范的内容，和一些技术实现的方式。移动端的技术实现相对比较可控和有限，多了解一些技术有助于进行有效的产品设计。盲目的设计要么实现不了，要么最终无法通过审核，是不可取的。移动端的一些特效也需要清楚，如拖拽、滑动、下拉、手势等，产品经理如果不了解这些，都设计不出操作体验很好的产品来。另外诸如不能获取用户的某些敏感数据等注意事项，都需要在产品设计时就考虑到。这块具体的还是要看场景来决定。

基础的技术名词

如前面最开头所提到的那些，如缓存、Cookie、Ajax、中间件、数据库等等名词，需要去了解一下其所表示的意思，开发人员在沟通的时候会经常提到这些，如果你听不懂，就会比较杯具。另外掌握如缓存、Cookie 的使用，还可以在产品后端逻辑中添加一些功能，比如让 Cookie 记录用户填写的用户名，下次再来登录时不要重复输入；用缓存记录用户的登录验证信息，在一段时间内不需要用户重新登录等等。

还有就是区分学习一下产品实现的开发语言，比如 Java、PHP、.Net 等，了解一些基本的知识点，如部署方式，运行载体等，比如常说的 LAMP 是指 Linux+Apache+Mysql+PHP 的组合；了解一些脚本语言，如 JS 脚本、CSS 等。这里需要提一下的是，产品经理最好懂 HTML，注意不是 HTML5，当然要是懂 H5 就更好了，HTML 是最基本的网页脚本语言，有时候看不懂某个页面的逻辑时，可以点击右键属性—“查看网页源代码”来看一看。

上面所提到的这些是个人觉得比较基础也比较容易掌握的技术点，特别是 SQL，强烈建议产品经理们都学习一下。不过这是一个持续学习的过程，只要你不转行，学习了之后会一直受益。