

在产品接触用户的过程中，我们要做哪些事情？

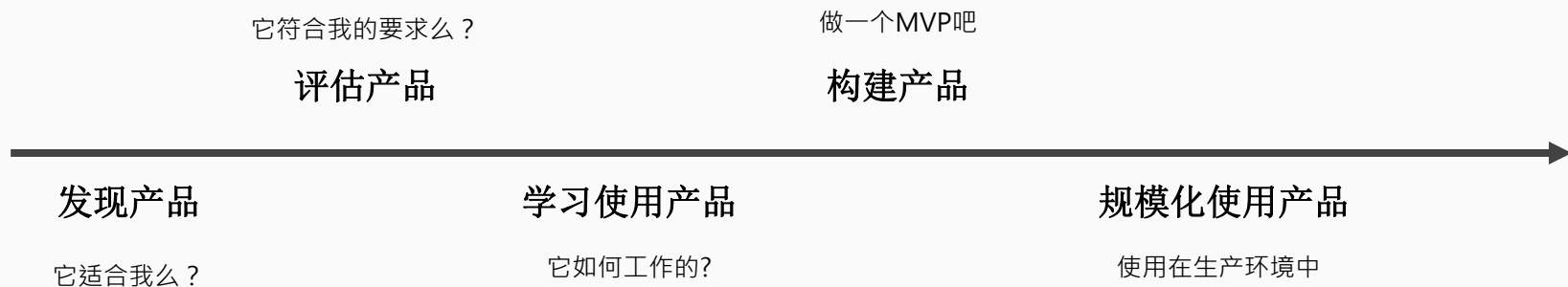
wanghaisheng

<https://github.com/wanghaisheng>



介绍

用户使用产品的五个步骤



发现产品时的 首要问题

1. 这是什么东西？
2. 这个东西能否解决我遇见的问题？
3. 这个东西是否可信赖？



什么样的东西可以回答用户的问题？

- **Landing Page:** 一个简洁明了的落地页，可以解答用户心中的所有疑惑
- **SEO/SEM:** 让用户带着问题在搜索的时候，能够看到你的解决方案
- **社交网络传播:** 让用户在社交媒体中发现自己信服的人也在使用这个产品
- **大会传播:** 参加线下、线上会议，用会议为产品背书
- **社区/博客:** 在社区/博客中介绍场景化的内容，解答用户对于能否解决问题的疑惑
- **案例学习:** 通过案例学习，一方面增进信任，一方面为类似问题找到解决方案
- **邮件列表:** 定期邮件 PR，推广产品

还有什么可以解决这个问题

- 线上媒体/线下媒体：用户可以在自己看新闻的地方了解到你的产品。
- 播客/音频媒体：让用户在听东西的时候，听到你的产品
- 线上 & 线下群组：社交关系，让用户增强对你的产品的信任
 - 民间以及官方qq微信群：用户之间的相互讨论，用户向官方抛出问题
 - discord/gitter/Ros discourse
- 推荐返利：通过社交关系转化来解决上述的三个问题

评估产品，用户想要知道的问题

1. 它是否解决我的问题？
2. 它看起来好用么？
3. 它看起来完成度高么（完整么）？
4. 价格灵活么？



什么样的东西可以回答用户的问题？

- 文档
- 产品/工具页面
- 论坛
- **FAQs**
- 用户案例
- 价格页面
- 线上会议

还有什么可以解决这个问题

- Github
- StackOverflow/SegmentFault/掘金/电子发烧友?

学习产品，用户需要知道的

1. 搞定 Hello World!
2. 文档的阅读体验如何？我能找到我想要的东西么？
3. 我愿意相信这个产品么？
4. 这个产品有社区么？



什么样的东西可以回答用户的问题？

- 快速开始教程
- 代码案例
- 使用教程
- 在线客服

还有什么可以解决这个问题

- 社区论坛/社区生态
- 合作伙伴机制

构建产品，用户需要的

1. 快速搞定 MVP
2. 产品使用体验好么？
3. 支持响应及时吗？
4. 值得我花钱使用么？



什么样的东西可以回答用户的问题？

- 产品页面
- 论坛
- SDKs
- API 文档
- 沙盒环境
- 变更日志
- 线上/线下 Workshop

规模化使用产品， 用户需要知道

1. 我还能做什么？
2. 我能否给予反馈？
3. 我如何参与并贡献其中？



什么样的东西可以回答用户的问题？

- 案例展示
- 多平台的联动，内外系统的打通，内建工单/用户反馈机制
- 布道师计划
- 合作伙伴计划

桥梁-正向的产品改进运营链条-数据驱动

● 从内向外的输出

- 源代码的多平台同步、自动更新
- 官方文档、**sdk**、教程、课程、专题培训、认证
- 合作伙伴贡献的教程、课程、专题培训、**up**的招募
- 嘉楠学院/**university**的建设 **awesome day**、**hackthon day**等
- 博文、视频素材的半自动化生产
- 多社交媒体的信息传播 自动提交和发布

● 从外向内的反馈

- 社交媒体的留言、私信汇总和联动
- **github gitee**的**issue discussion pr**的联动
- 官方以及其他社区命中关键词的**post**、**comment**监测和联动
- **qq**微信**discord**讨论群组的讨论检测,机器人的联动, 值守工程师的联动
- 反馈的智能分级和联动满意度的评价机制
- 知识库的沉淀和转化

画像

- 分析github仓库的star、fork的现有用户
 - 分析这些用户的基本信息
 - 推测用户所在公司 利用域名邮箱等信息
 - 获取用户的博客链接并分析博文技术栈
 - 分析他们共同关注的其他repo和技术栈
 - 寻求和这些技术栈相关项目社区意见领袖的合作
- 抓取b站上官方视频、关键词检索视频的播放量、评论、用户基本信息
 - 分析用户共同关注的up 寻找领域交叉型up
 - 分析标题、播放量反推文档、教程的侧重点
 - 分析评论 找到改进的点 比如
 - “k210牛逼是牛逼，但是没有openmv好上手。”
 - “阿婆主，用哪个固件能移植openmv的例程”

潜在客户/竞对/达人的监测

- 招聘需求的异动
- 专利的异动
- 学术论文的监测
- 社交媒体评论的监测
- 微信/淘宝销量/google/youtube关键词及搜索指数的监测
- 针对关键词的搜索引擎结果的监测

用得上、用得好

- 用的安心
- 用的放心
- 用的顺心
- 用的开心

滚雪球

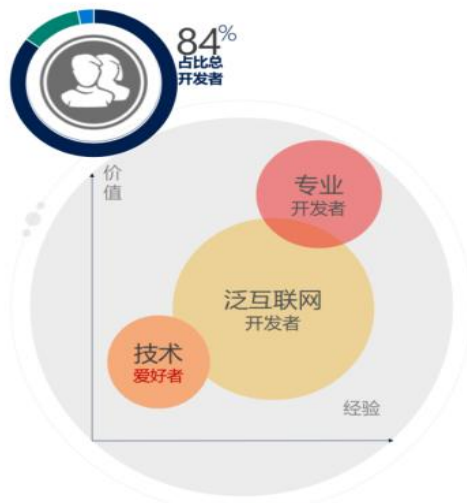
- 第一是“使用”，要先有人把这个项目用起来，形成一定的用户群体，进行讨论和交流；github 过1w star的四位数而已
- 第二是“创作”，如果没有人用这个东西来做创作，它的解决方案和发展性就会很少。案例集很重要
- 第三是“互助”，在社区运作的过程中，不懂、不会和不知道的人，通过互联网找到可能的解决方案，与社区成员进行交流，
- 第四是“分享”，社区成员分享信息，雪球就滚起来了

一位华为在做生态的朋友那里得到的材料

- 策略和运营目标如何选择，泛开发者到底是给业务带来营收，还是与开发者形成信任关系
- 我们产品在整个业务发展周期过程中需要哪类开发者，或者触达哪类开发者
- 从产品研发到规模化增长再到商业化前不同周期，开发者生态建设运营上需要有哪些方面的侧重

CSDN的调研

中国开发者趋于年轻化，重视学习培训、平台开放、技术支持和社区



图像来源: 2019-2020 中国开发者调查报告

81%

开发者的年龄在30岁以下

50%

近半数的开发者工作在一线城市

90后主流



80%

将编程是作为业余爱好

56%

认同自己属于某一个技术社区

业余爱好



80%

视文档为重要的学习资源，尤其新兴技术

> 10yr
< 3 yr

关注技术准确性
关注文档易用性

技术文档

81%

认为平台的开放很重要

56%

开发者参与贡献过开源项目

平台开放



83%

认为高质及时有效的技术支持很重要

技术支持



85%

一周内会多次访问社区

40%

参与社区贡献和讨论 Participate

技术社区



78%

学习新技能或是提高现有技能

71%

认为平台应提供培训认证

培训认证

57%

对组织购买工具软件组件起到一定的影响作用

影响决策



64%

参与生态的目的是为了盈利

商业盈利

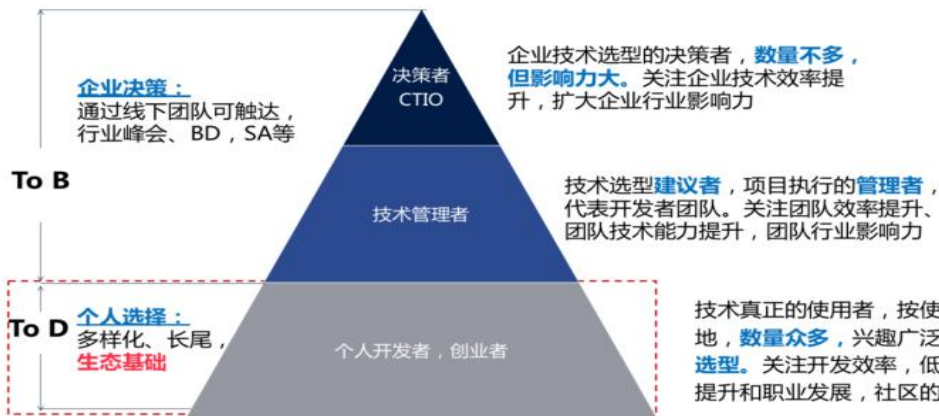


数据来源: Stack Overflow Developer Survey Results 报告 N= 100,000

开发者洞察

开发者影响技术选型，做好对个人开发者的支持，是构建生态的关键

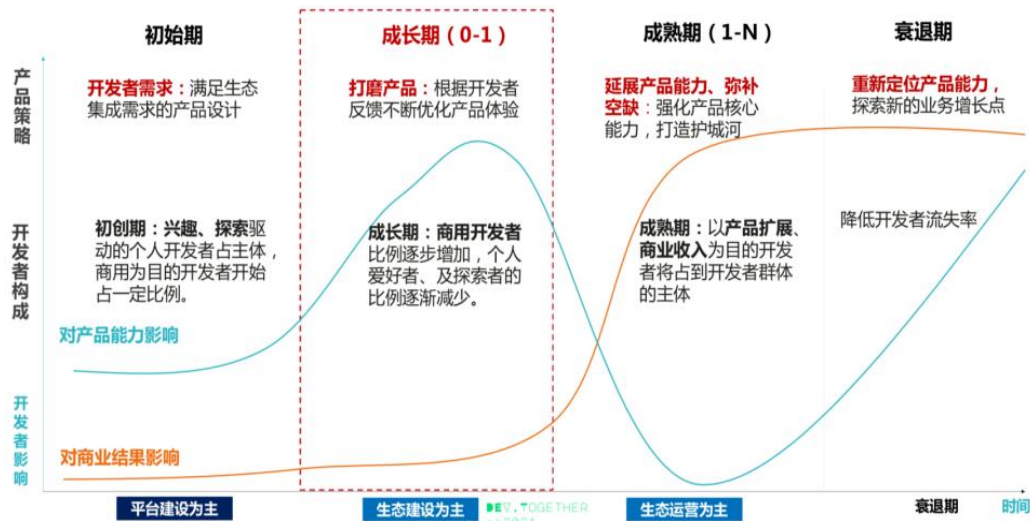
- CTO和中层管理者代表企业参与生态，可以通过线下团队触达；**个人开发者和创业者以个人身份参与生态，需要更多样化的方式触达。**
- 在企业技术选型过程中，通常开发者的意见会影响最终的技术选型，所以**做好对个人开发者和创业者的支持，是构建技术生态的关键。**



技术真正的使用者，按使用习惯和兴趣选择技术，可影响最终的项目落地，**数量众多**，兴趣广泛，**对具体技术的选择会影响上层管理者的技术选型。**关注开发效率，低学习成本，丰富的开发工具和套件、个人能力提升和职业发展，社区的影响力

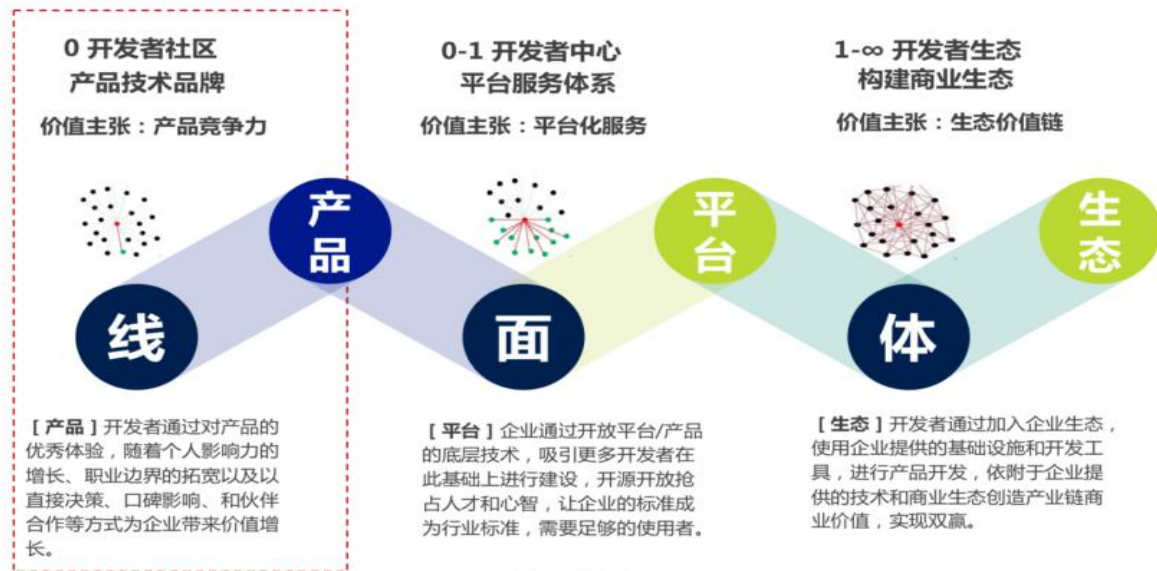
开发者生态运营的最终目标

开发者的构成对产品和商业产生影响，其价值贯穿产品的生命周期



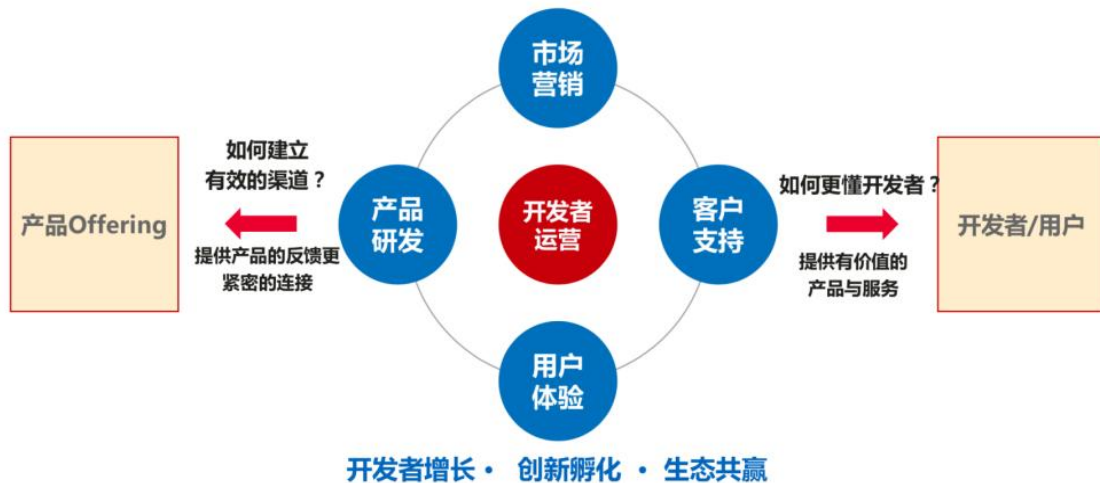
开发者0-1运营体系

产品形态演进的商业逻辑以及开发者生态构建各阶段运营重点



开发者运营的职能

运营是连接产品和用户的桥梁，目标是提升产品和用户终生价值



开发者运营的三个原则

围绕开发者技术成长、开发构建与技术交流构建开发者运营平台



多渠道全流程

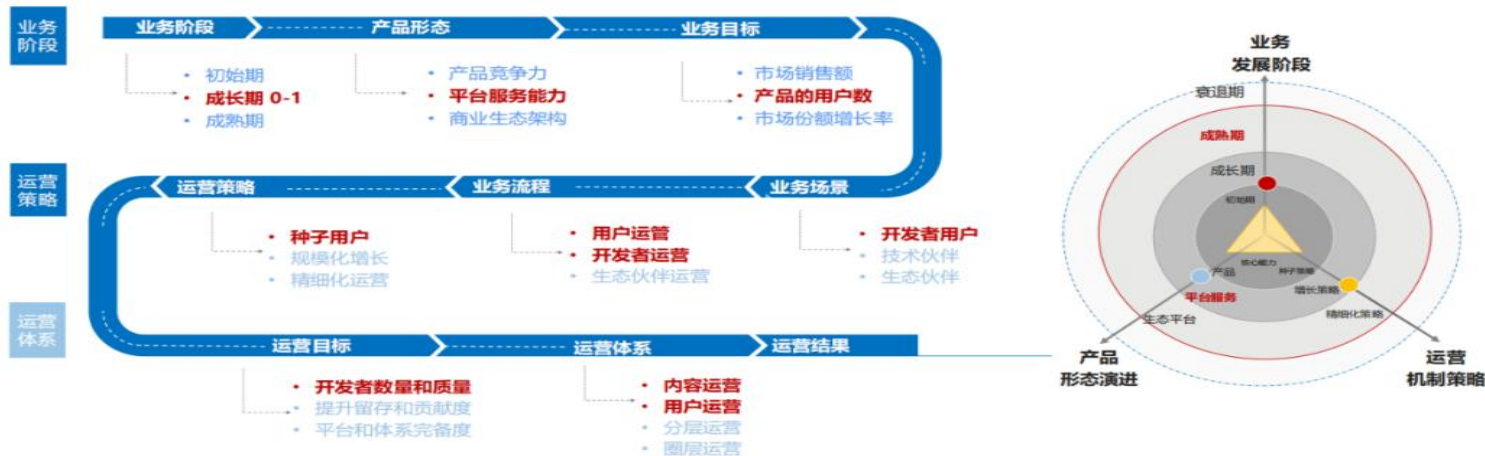
打通开发者全旅程数字化触点，提升开发者体验，建立用户信任和忠诚



运营地图

沿着旧地图找不到新大陆，0-1阶段需匹配相应运营策略

平台企业应围绕其所处的业务阶段和发展目标，提供与目标开发者相匹配的运营体系和支撑服务



主阵地

开发者生态运营的主阵地，一站式学习、开发、交流的自服务平台

	开发者中心			开发者社区
模块	开发资源	学习赋能	技术支持	内容/活动/生态等
定位	提供基础能力	提供学习内容	收集用户声音	一站式问题解决和交流互动平台
设置目的	提供开发者必须的开发工具和资源	满足开发者学习提升/产品使用的需求	提供产品问题、意见建议的反馈入口	通过开发者社区，提供学习、支持、互动、生态合作等开发者所需要的功能和内容。
细分项	API SDK 开发文档 示例代码 产品FAQ 开源项目	新手入门 进阶课程 视频教程 沙箱实验 培训认证	用户声音 工单入口 客服电话	博客 论坛 活动 学院 大赛 商业等
面向人群	使用产品的开发者			泛开发者

三大抓手

- 学、练、赛、考
持续提升能力
- 积分、等级、徽章
成长/激励体系→自驱力
- 面向决策者的内容
- 面向个人的内容
- 面向国内社区
- 面向国外社区

从0 到 1 构建开发者社区运营体系需要具备哪些基础运营能力

初始期以平台建设为主，通过内容与用户运营实现增长和留存，社区平台完善自服务体系和交互的运营功能



内容运营

Content Operation

- 提升内容产出质量，加大社区内容(UGC)产出数量，加强UGC占比和内容体验
- 施行数据驱动的内容流程，专业/社区内容产出计划，产品教程，构建学习路径、创建技术专辑和知识体系，



用户运营

User Ops & Growth

- 搭建用户分层管理的运营体系，建立用户激励机制，提高用户互动率和贡献率
- 招募布道师，培养社区贡献者，专家激励机制和影响力计划。
- 建立用户成长体系，活跃机制，留存和转化机制。



社区运营

Community Operation

- 建设社区运营平台，增加UGC贡献机制，社区平台完善搭建社区信用积分体系，引入专家成长体系，
- 建立社区的总体调性和文化，增强社区用户互动功能，实现增加用户活跃度的功能落地，

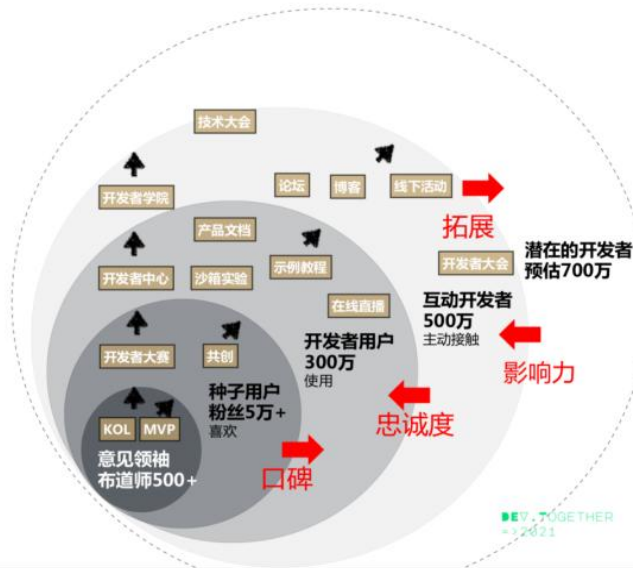
两种策略push vs pull

● 扶持计划/激励机制

- 针对学生的
- 针对中小企业的
- 针对企业开发人员
- 针对合作伙伴

圈层运营促进开发者互动传播，提升口碑影响力，放大产品传播效率

目标：扩大意见领袖（KOL），粉丝，用户群体，提升口碑，影响力，忠诚度；提升产品认知，形成自服务，培育人才



产品传播生命周期

