

19 | 如何用最小的代价做产品？

郑晔 2019-02-11



00:00

讲述：郑晔

大小：10.33M

时长：11:16

1.0x

你好，我是郑晔。

前面我们讲了开发任务的分解和需求管理的分解，这些都是针对“已经确定好要做的事情”的分解策略，今天我们再上一个台阶，聊聊面对那些不确定的产品功能该如何分解。

产品经理的想法层出不穷，但是，如果我们一味闷着头实现产品经理的想法，无论你有多大的开发团队都是不够用的。我们要学会用最小的代价做产品。

谈到产品这个话题，在 [“精益创业：产品经理不靠谱，你该怎么办？”](#) 这篇文章中，我

给你分享了精益创业的理念，任何的想法都要放到真实世界中检验。

我们的直觉当然是把所有的东西都实现了再去检验，但是世界不会停下来等着我们。事实一次又一次教育我们，“憋大招”的瀑布式软件开发已经成为不合时宜的“老古董”。那我们的理想怎么实现呢？唯有分解。

我们前面提到，精益创业就是通过不断地尝试在真实世界中验证产品想法，其中一个重要的实践是最小可行产品（Minimum Viable Product, MVP），我们这次就把这个实践展开讨论一下。

什么叫最小可行产品？就是“刚刚好”满足客户需求的产品。客户需求好理解，怎么算“刚刚好”呢？其中的关键在于理解“最小”和“可行”。

最小的代价

先说“最小”。这里的“最小”，指的是最小的代价。什么叫最小的代价，就是能不做的事情就不做，能简化的事情就简化。

首先，我们必须清楚一件事，**我们要做的是验证一个想法的可行性，甚至不是为了开发一个软件，开发软件只是一种验证手段。**

很多程序员都会有一个认识上的误区，容易把解决方案当成问题。我们开发软件的目的是为了解决问题，如果不写软件就把问题解决了，岂不是更好。

我先讲一个自己的经历，帮你理解一下什么叫“最小”。有一次，有一个朋友找我帮忙，他手头有一些制造业的客户，想做一个物联网相关的项目，帮助这些客户改造设备，实现物联网功能。

该怎么着手呢？把软件写好，给客户试用吗？这样时间太长，成本太高。那么，我们是怎么做的呢？

第一步，我们要验证这样一个想法是否可行。我们做了一个产品文档，就好像我们已经有了这个产品一样，让负责销售的同事拿着这个文档给客户讲讲，看看客户对这个想法的反映。

在这个过程中，我们验证了基本的想法，已有设备进行物联网化改造的需求存在，客户看到了这样的一个东西，各种各样的想法和要求就会冒出来。

此外，我们还获得了一个额外的收获，我们知道了客户对于这样一个产品能够接受的价格区间，这可以帮助团队给产品进行适当的定价。

验证了方向上的想法，我们开始进入到具体的产品设计阶段。这个阶段我们想验证的是，我们给出的产品设计用户是否可以接受。于是，我们决定把这个产品的交互做出来。

得益于原型工具的快速发展，我们用一个原型工具做出了相对完整的用户界面，而且把各种交互流都做出来了。在用户看来，这几乎就是完整的软件了。

他们甚至可以在自己的设备上体验一下这个产品用起来是什么感觉的。一旦上手用起来，他们就会抛出各种细节的问题：如果这样就好了，如果能做到这个就太棒了。当然，他们也会说，这个东西我不需要。

这个时候，我们就可以知道，我们在产品上的假设哪些是好的，哪些是不流畅的。团队拿到这些反馈，就可以再调整产品设计，然后，再给到用户去测试，如此反复进行。有的时候，产品会在一天之内改好几个版本。

经过多轮测试下来，团队有了一大堆的用户反馈，而且是来自真实用户的反馈。接下来，就是整理这些用户反馈，决定哪些可以真正的开发出来，这时候，团队才真正进入到开发阶段。

不知道你注意到了没有，迄今为止，**这个团队验证了一大堆的想法，而代码却是一行都没有写，所有花费的工作量都是有针对性的验证。**

我们经常听到一个段子，叫“就差一个程序员了”。这说的是，一个创业者把前期的准备都做好，就差程序员把产品开发出来了。

按照 MVP 的思想，这个创业者做的就是对的，前提是他真的把前期准备都做好了。

开发软件是一件成本很高的事情。如果只是验证想法，无论是创业方向，还是产品设计，我们可以找到各种各样的手段，不用写代码。

即便我们不是在做一个新产品，我们依然可以运用这个“最小代价”的理念在日常工作中做事。比如，怎么来衡量产品经理的产品设计是不是好的。我会问，这个功能不做，用户会怎么样？有没有什么替代方案等等。以此来帮助产品经理想清楚自己的产品设计是否真的有价值。

可行的路径

说完了“最小”，我们再来看“可行”。可行是要找到一条路径，给用户一个完整的体验。做程序员出身的人，对软件系统的认识总是一个模块一个模块的，相对比较弱的方面是缺少一个完整的图景。

但从产品可行的角度，我们需要转换一下思路，**不是一个模块做得有多完整，而一条用户路径是否通畅。**

我再给你分享一个我当年做 P2P 项目经历，这里的 P2P 指的是个人对个人的互联网借贷平台。

这是一个从头开始的项目，项目方和所有的项目方一样，希望昨天这个项目就上线了，如果不能，那就尽快上线一个版本。他们给我们一个时间线，第一个上线的版本是一个月之后。

摆在我们面前的问题是，无论如何，在一个“一穷二白”的基础上，要在一个月内完成一个完整的借贷平台是不太可能的。

时间有限，我们只能做最基本的东西，许多运营上的想法，比如，发红包代金券之类的，第一期一律不做。即便如此，我们仍然认为完成完整的借贷循环是不现实的。

于是，我们就开始从需求完整性的角度动脑筋。这是一个借贷系统，其最基本的模型是：贷款方贷款之后，一次性拿到所有的钱，然后用等额本息的方式每个月还款，最后一个月剩多

少钱一次性全还了。

我们在这个模型中找到了一个关键点，每个月还款。换句话说，第一笔贷款发生之后，最早的一笔还款是发生在一个月之后的。

于是，我们做了一个决定，第一个版本只包含贷款能力。是的，这个版本只能贷款，不能还款。因为用户一个月之内不会用到这个功能，你从页面上，完全看不出这样的能力缺失，因为一个月内，根本没有任何用户有可还的款项。

因为缩减了项目规模，我们在预期的一个月內完成所有开发，成功地把项目送上了线。第一批早期用户就开始了使用。从用户的视角看，这是一个功能完整的项目，虽然简单了点，但它是完整的。

当然，我们把还款排到了下一期。按照我们两周一迭代的节奏，在第一期上线两周之后，我们就会上线还款功能，届时贷款方将拥有一个真正的还款功能。

不过，这个还款功能只是每期的等额本息还款，最后的一次性还剩余所有贷款的功能，我们依然是不支持的。因为根据需求设计，最后一次还款最早发生在一年之后。

在我们把基本的功能全部送上线之后，这个系统就是一个真正的、完整的借贷平台了。但是，相对于其他提供相同能力的平台而言，这个系统依然还是很简单。比如，常见的运营功能、短期借贷计划，这个平台都没有。

但我们有了基础，接下来，就是在基础上叠加，而且随着项目方自己团队的构建，我们拥有了够大的团队，可以同时做几个大需求了。

就这样，几个月之后，我们就逐步上线了一个功能相对完整的 P2P 平台。在这个过程中，我们每个阶段都会上线新功能，从用户可见的角度，他看到的始终是一个完整的平台，其中的变化只有站在内部实现者的角度才能看得清楚。

和大家分享这个例子，主要是想破除大家对于一个“完整”系统概念的认识。**当时间有限时，我们需要学会找到一条可行的路径，在完整用户体验和完整系统之间，找到一个平衡。**

站在开发团队的角度，我们怎样把 MVP 理念运用在自己的工作中呢？当产品经理有一大堆要实现的功能时，我们就可以根据 MVP 理念，从这些产品功能中找出一条最小的可行路径，重新安排一个合理的开发计划。

总结时刻

产品同样需要分解，目前在探索产品的不确定性上的最佳实践是精益创业，而精益创业就包含了将庞大的产品分而治之的方式：最小可行产品（Minimum Viable Product, MVP）。最小可行产品就是“刚刚好”满足客户需求的产品。

想要在实践中运用好最小可行产品的理念，就是要用最小的代价找到一条可行的路径。最小的代价就是能不做的事就不做，能简化的事情就简化。

程序员通常愿意用自己的代码解决问题，而写代码通常是代价非常高的解决方案，它应该成为最后的产品解决方案。

可行的路径，是一条完整的用户体验路径，至少在用户眼中是这样的。我们常常会想给客户一个完整的系统，但在时间有限的情况下，我们必须学会分解。

如果今天的内容你只能记住一件事，那请记住：**做好产品开发，最可行的方式是采用 MVP。**

最后，我想请你分享一下，你遇到或听说过采用 MVP 或类似方法解决问题的案例吗？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



Geek_73143a

由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。

Ctrl + Enter 发表

0/2000字

提交留言

精选留言(23)



行与修

这篇可以说是精益创业的姊妹篇，现在在做的项目也基本是按照MVP的思路来做的，只是任务分解方面还不够细，有点贪全~~

总的来说，小步快跑。先弄清客户想要什么，整理后再用自己的语言复述一遍，后面拿着原型+业务交互给客户确认，同时把自己的设计理念植入进去和客户探讨，有些点对方之前也没有想清楚，尤其是实现方面需要专业的软件意见去参考，而我们从探讨中更新自己的认知，如此往复几次，大方向上就不会有偏差，后面在实施过程中再局部调整。个人认为尤其是对于不确定性很高或者从零开始的工作，这是个可操作的方式，也希望能从大家的分享中获得灵感！

作者回复: 你理解了！

2019-02-11

2

14



Kăfkă²⁰²⁰

补充一个和MVP有关的话题。前阵子有个关于产品技术的段子，讲的是技术问产品，开发某个功能预计

会有多少人用；而产品说，不上线不知道会有多少人用。其实产品要有一种意识，即开发程序是代价高昂的解决方案，特别是开发某个不确定的特性。所以他需要花许多力气琢磨，找出方案的关键点，构建 MVP 去验证，而不是先做一个完整的产品方案让开发去完成。虽然这会导致产品经理多做很多工作，但对于团队来说整体代价反而是小的。更进一步说，产品经理的职责本来就是找到最经济有效的解决方案，而不是画一个原型让开发去完成

作者回复: 糊涂的产品经理到处都是，程序员不反抗，他们就意识不到自己的糊涂。

2019-02-11



10



Ryoma

之前做了一个类似 TodoList 的项目，有个功能是“历史任务”。当时由于时间紧急，与产品经理商量，在第一版上线时，其实并没有上线“历史任务”这个功能——因为历史任务的定义是一个月之前的任务。在接下来的版本中再实现“历史任务”这个功能，与老师处理问题的思路如出一辙。

作者回复: 多谢分享!

2019-02-17



7



丁丁历险记

如果全篇就记住一句话。那么就是开发的成本很高很高，不要烂用。

作者回复: 你的一句话 :)

2019-11-12



4



Henry Liao

文章说，得益于原型工具的快速发展，我们用一个原型工具做出了相对完整的... 这样的原型工具是指什么啊？有推荐的吗

作者回复: 举个例子, 墨刀。

2019-10-14

1

2



陈斯佳

最小可行性产品 也就是用最小的代价去找到一条给用户完整体验的路径, 这种完整不是指模块做得有多完整, 而是这条路径, 用户的体验是否通畅。

2019-05-22

1

2



geoxs

公司比较老, 并且是to b的那种, 目前对于新项目主要还是瀑布式的开发一个基线, 后续再迭代。看了本文受益匪浅。不过有一个问题: 像文中举的p2p的例子, 这样开发会不会风险太大? 利益与风险的权衡是由老板做决定吗? 谢谢

作者回复: 好问题, 当时老板给的要求是尽快上线, 这是我们找到的一条通往目标的路径, 而且我们给出了一个合理的推理, 老板并不反对。

所谓风险是不确定的, 当你看到一个几乎是确定的路径, 你就不会那么担心了。很多团队是用胆子大来做事, 这才是风险很大的。

2019-02-12

1

2



秦奋

当时做敏捷开发去开发产品就是前期做一个最小可交付版本, 后面的逐步增加功能。

作者回复: 这种做法没问题

2020-12-26

1

1



Bravery168

MVP确实是一个比较好的理念，小步前进，快速验证。

2020-11-08



1



一个帅哥

最小指的是能不加的功能就坚决不加。可行指的是刚好可以用

作者回复: 很好的总结!

2020-06-16



1



王维

以前做过一个政府部门的绩效考核项目，客户交来一叠厚厚的需求，然后我们就埋头“研究”需求，既没有分解需求，也没有识别出核心需求，项目一开始，开发人员天天打酱油，只有两位项目经理忙得不亦乐乎，结果他们做出了两个不同版本的开发需求。过了一两个月，公司要做需求评审，结果我们匆匆做出了一个所谓的原型才敷衍过去。接下来就开始做开发，好不容易项目做完了，在后期客户又提出要加上工作流审批的需求，结果我们又是一顿好忙。

所以学习了这篇文章，觉得需求一定一定要分解，要识别出核心需求，一定要在MVP的基础上不断重构，直到满足客户需求。

作者回复: 唉，多做了多少不需要的需求啊。

2020-03-18



1



246小言

总结如下：1.MVP的核心是最小可行产品。是指团队在Deadline前推出一个客户最小可行的产品。

作者回复: MVP 就是最小可行产品，Deadline 是形式，MVP 是内核。

2019-03-24



1



Geek9467

太棒了

2020-12-07



孙瑜

听到借贷系统的例子，想起了五年前去的创业公司，做的理财功能也是一样，只有一个月，先只有购买功能，因为最早的兑付功能要在三个月后，没有任何营销功能，不断迭代。

作者回复: 异曲同工

2020-05-06



Jerry Wu

看完“任务分解”模块，发现我们公司还在“憋大招”的时代，太容易出问题了。

有一次，我改进一个项目，分几次提交代码，每次都和他说，XX功能做好了，正式环境先上线。这样即使出事，也是小问题，对客户的影响不大。老板每次都回复了个：1。

到了周末，我微信狂弹消息，说系统不行啦，客户很生气，后果很严重之类的。最后，我发现，他大招没放好，一次性上线了好几个版本的代码，但忘记执行一个SQL文件~~

我心中一万只羊驼奔腾而过。。。。

作者回复: 大招伤敌一千，但更可能是自损八百，甚至一千二。

2020-04-22





chenzesam

举例太精辟了

2019-07-23



刘冲

如果做的产品是面向大众的怎么办，不同客户理解和需求不一样，怎么做这个需求呢

作者回复: 前行到26

2019-06-27



郑大大的粉丝

嗯.....郑大大每当你讲到需求管理时候，我总是没那么满意😞😞😞

作者回复: 这是程序员视角。

2019-05-09



郑大大的粉丝

我们第一期还做了提前还款！
用户不可能一个月还款，但是能提前还款呀哈哈哈哈哈

作者回复: 你居然还记得。

2019-05-09



enjoylearning

好多互联网公司都是做的创新型产品，不断去尝试新想法，to B还好说，to C恐怕无最终实现还是不好

说，此时就因该如作者举的最后例子，寻找用户立即使用的和一个月后才会用的去排档期。

作者回复: 越是 to C 这种不确定性的产品，越要在前期多试。

2019-03-29



有渔@蔡

学得很受用，不愧是thoughtwork出来的。

2019-03-28



行者

原来还可以这样。谢谢老师分享

2019-03-24



彩色的沙漠

MVP最小可行的产品，之前对"最小"理解成了功能最小，原来是"最小代价"。

作者回复: 道理挺简单，就怕走错路。

2019-02-13

