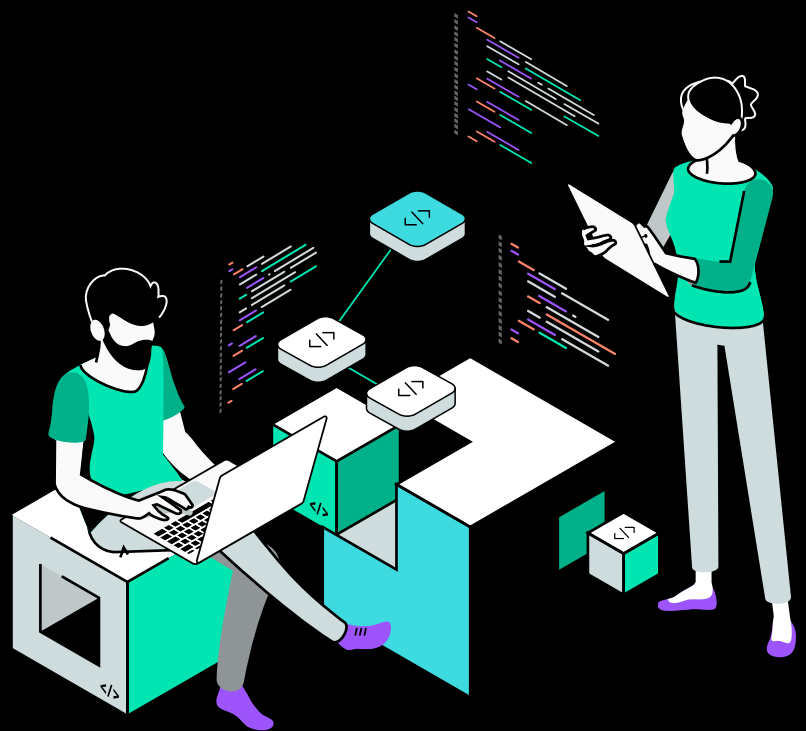


提升开发速度的八个建议

如何通过提升开发速度
驱动绩效变革



目录

软件正在已经占领世界	3
什么是开发速度?	4
什么是 DVI?	5
为什么需要开发速度?	5
开发速度的影响是什么?	6
提升开发速度的八个建议	7
建议 #1 - 学会快速失败	8
建议 #2 - 构建产品, 提升信心	9
建议 #3 - 原型	10
建议 #4 - 教堂/集市开发, 可行! 大泥球开发, 千万不行!	11
建议 #5 - 管理技术债务	12
建议 #6 - 管理代码分支	13
建议 #7 - 保持可追踪性	14
建议 #8 - 注意你的 KPI	15
关于 Incredibuild	16

软件正在 已经占领世界

十年前，企业家 Marc Andreessen 曾[说过](#)一句著名的话

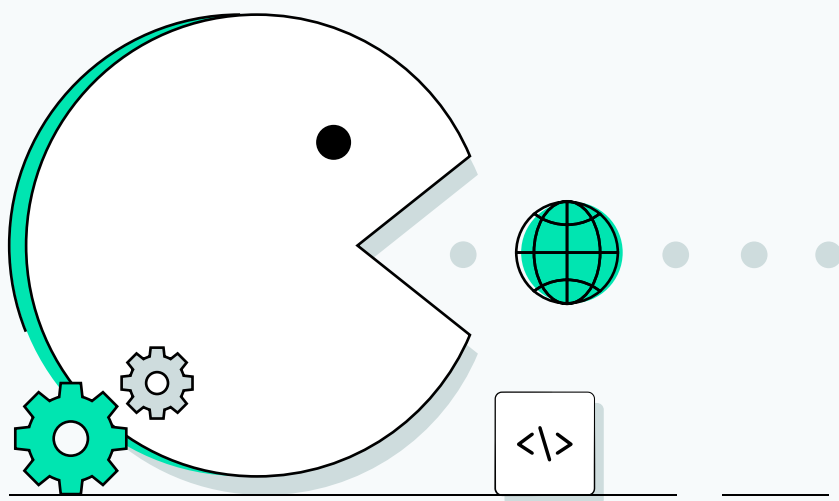
“软件正在占领世界”

如今，这个过程已经或多或少完成。

技术驱动着各行各业的企业，涉及到商业和客户关系的方方面面。

这意味着今天，除了各自的核心产业和行业竞争力之外，几乎每个组织都需要一定的软件开发能力。

即使是在公共交通（比如 Uber，航空公司）或酒店（比如 AirBnB，连锁酒店）等非科技行业，也必须具备整套全新的软件开发技能，才能生存下来。



什么是开发速度？

速度(Velocity):

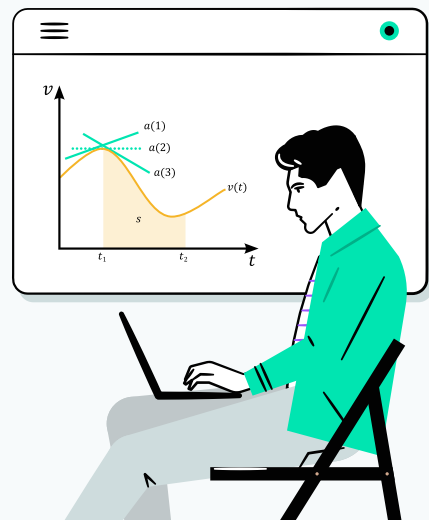
速度, 其大小为物体的速度, 其方向为物体的运动方向, 用方程表示 $v = \Delta s / \Delta t$

在物理学上, 速度的概念很常见。在以数字为中心的业务中, 敏捷性对任务至关重要, 保持软件的时效性是敏捷的关键, 所以我们需要讨论开发速度。

Microsoft 和 McKinsey 对开发速度的定义为

“通过提升软件开发速度驱动业务绩效变革。”

为了实现这一点, 组织需要授权内部开发团队, 打造一个鼓励创新的开发环境, 同时减少各种开发阻力。



什么是 DVI?

在 2020 年早期, [McKinsey](#) 调查了近 450 家大型企业, 量化分析了开发速度对企业的影响, 发现了驱动程序影响最大。其测量指数, 开发速度指数 (DVI), 体现了一个组织将数字理念投入生产的能力。

这个指数也成为了衡量开发者速度的最有效标准。

为什么需要开发速度?

Microsoft 发现, 在所有行业中, 拥有高 DVI 的公司具有以下特性:

X4-5

收益
增长
更快

60%

股东
回报
更高

20%

运营
收益
更高

55%

创新
能力
更强

McKinsey 的 DVI 指数以大约 50 个参数为标准, 覆盖了架构、工程实践、安全和开发工具等等, 他们发现 DVI 和关键业务绩效指标之间存在很强的相关性。

总结:


DVI 高的公司敏捷性更高, 对市场变化和客户需求
的反应更迅速, 最终更有利于盈利。

开发速度的影响因素是什么？



任何的结果，都是在各种技能、影响因素的多重作用下产生的。McKinsey 发现，希望提升开发速度的公司需要授权开发者，预测关键的推动因素，保证投资方向与客户价值一致，并清除生产障碍。

Github 创建了自己的开发速度模型，取名为 [SPACE](#)，也认为影响开发生产力的因素是复杂多样的。为了揭开开发速度的谜底，McKinsey 分析了近 50 个性能因素，探索高速开发的形成条件。他们发现，工具、文化、产品管理和人才管理对企业绩效的影响最大。





提升开发 速度的 八个建议



建议

#1

学会快速失败

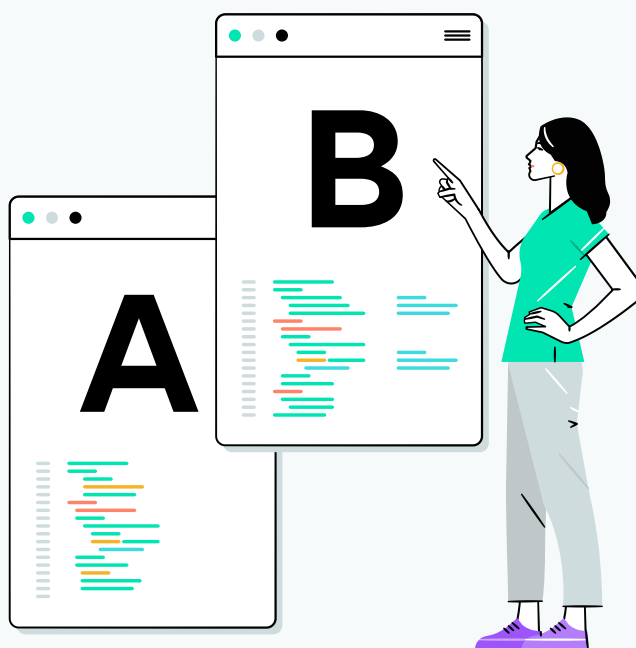
聪明的组织知道，失败并不是什么坏事。产品失败，意味着产品不符合标准，或者不能满足用户需求。

为什么不是坏事？因为失败越快，改进越快，离目标用户的需求也更近了一步。

在产品层面，AB 测试可以为产品的更改快速提供反馈。

在开发层面，[测试驱动开发](#)（TTD），[静态代码分析](#)，以及左移策略可以帮助开发者更早发现问题，修复问题，更快向前推进。

技术和方法上鼓励敏捷开发和小型迭代，可以为开发者提供安全保障——他们将明白，错误改进也是开发的一部分，因此更有信心加速开发。



建议

#2

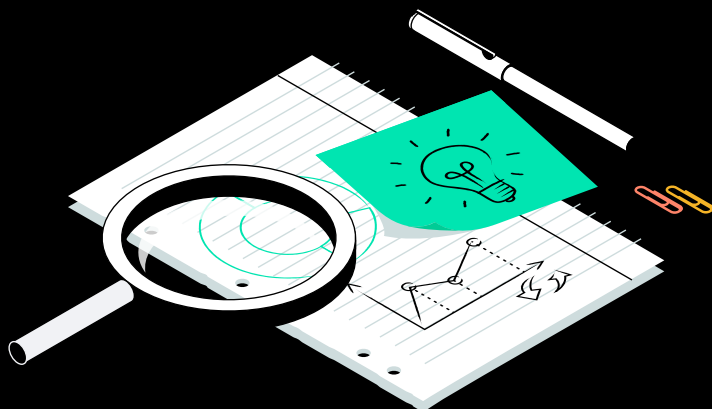
构建产品 和开发信心

快速失败，是组织创造最佳的产品、提升团队信心的前提。

为什么？和所有人一样，开发者在没有自信时工作会更慢。

对错误包容甚至鼓励的团队文化，可以让产品主管和开发者更有信心去尝试新的方法——其中许多方法最终可能让开发更好、更快。

从管理的角度来看，接受错误可以提高信心，鼓励试错（同时通过快速失败来控制犯错成本），并对新想法和非常规方案保持开放的态度。在开发层面，使用 [结对编程](#) 方法，在编码和设计时给予员工更多自由（必要时可以严格）。



建议

#3

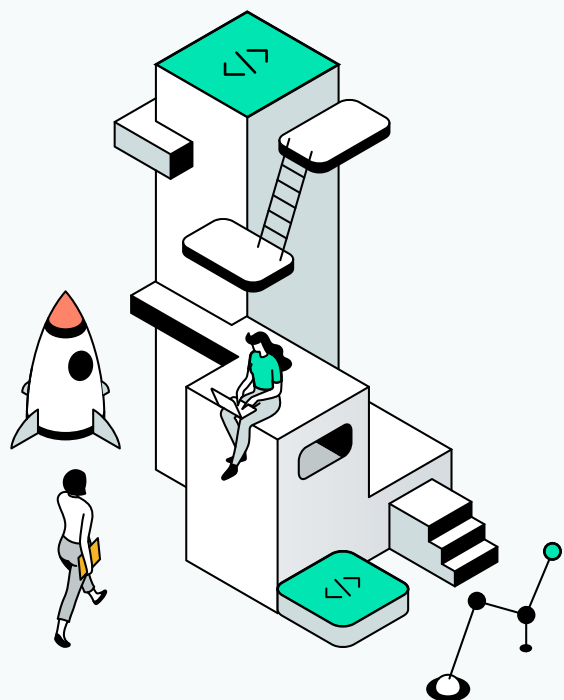
原型

尽管有学派认为原型是资源浪费，但事实是创建并放弃一个失败的原型比使用实际产品成本更低。

原型降低了风险，而且能帮助开发者积累经验——开发者可以在一个小沙箱中（而不是在实际的代码库中）安全地寻找并处理风险。

开发者更快地进行试验，获得结果，因此可以提升速度——即便这些试验可能最终并不可行。

开发原型鼓励打破常规思维，激发创新和突破——而所有这些操作都不会影响到实际产品。



建议

#4

大教堂/市集开发， 可行！

大泥球开发， 千万不行！

无论开发领袖选择[大教堂](#)或是[市集](#)开发策略，都可以维护概念的完整。但没有什么比[大泥球](#)（混乱的代码和架构）更影响开发速度了。

开发者需要清晰的架构、优秀的流程、有效且持续的内部沟通、完备的发布说明及文档，以及强调细节的工作环境。

准确定位代码中的新特性区域，追溯错误出现的根本原因，理解新代码的含义，更好地计划并了解需要测试什么、如何测试，这些方法都可以让开发加速进行。



建议

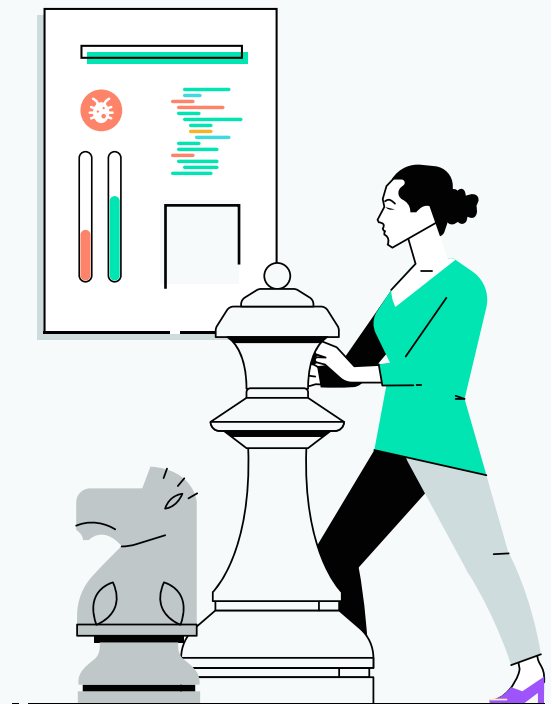
#5

管理技术债务

没人想要去构建一个“大泥球”。**技术债务** 源于长期在老旧的基础架构上进行开发，或者经常匆忙实施许多小改进，或者没有合适的计划、工具或库。这是一个持续存在的问题，是开发速度的敌人。

技术债务，意味着开发者必须投入更多的精力来解决每个新特性或修复 bug，这在一定程度上会导致开发者产生挫败感，影响他们的工作积极性。具有重大技术债务的产品甚至会赶走团队中的全能人才和重视效率的员工。此外，技术债务问题，也导致组织难以应用可加速开发、缩短上市时间、提升产品质量的新技术。

无论你的组织是选择在维护旧产品的同时构建新产品来处理技术债务，还是只是在旧代码库中小心翼翼地重建部分产品——为了实现更好的开发速度，解决技术债务是必须的。



建议

#6

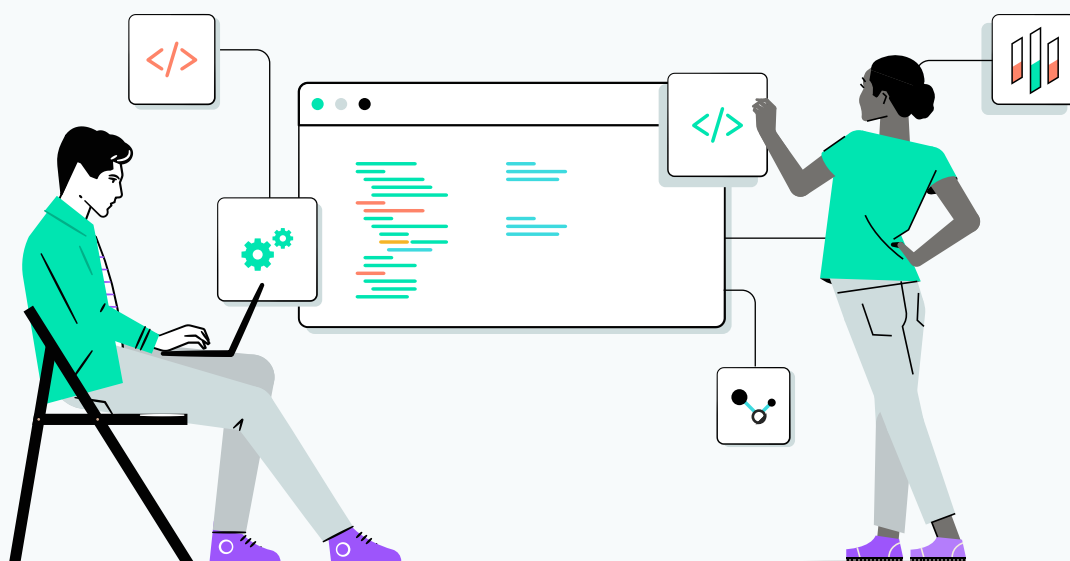
管理代码分支

分支可以帮助软件开发团队实现并行化。

然而，由于开放分支膨胀，或大量分支长期未进行合并，开发者需要花费很多时间和精力去追赶主分支，这对速度来说，是不利的因素。

有效的分支管理，通过降低开发团队需要维护的开放分支数量，也就是降低了后期维护的工作量（坦白地说，也减少了开发者的烦恼），从而提高开发速度。

为了避免“追逐主干分支”的问题，尽可能将大的变更分割成小块。对于看起来难以分割的大块变更，找到正确的重构方法（比如添加一个新的抽象级别），将变更分割成小块。这个操作不会影响代码，但会提高开发者的速度。



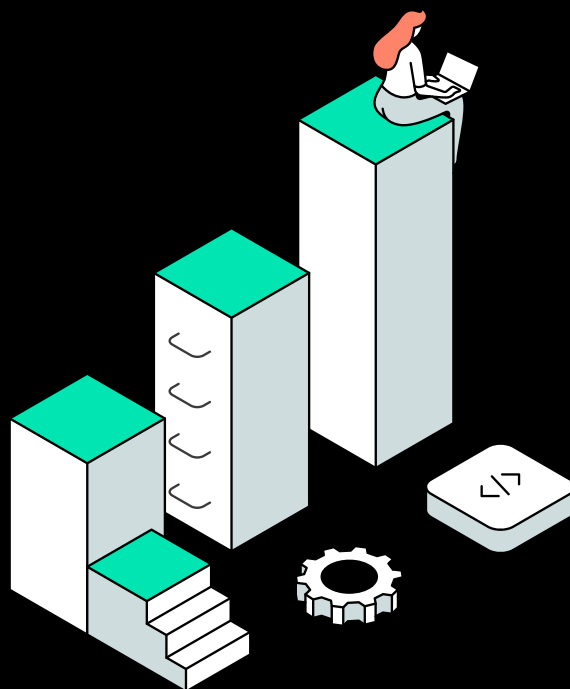
建议

#7

保持可追踪性

可追踪性 是一个非功能性需求，但不容忽视。可追踪性包括有效的日志记录和跨系统会话管理，同时在生产环境中公开所有相关数据，方便开发者访问。数据的追踪性极大地影响了速度。

当构建的可追踪性高时，开发者对系统的理解也更深，因此他们的行动更快，也更有信心。当然，可追踪性并不能取代测试和其他必要的检测。bug 出现时，如果开发者能信心满满地快速定位 bug，他们在快速发布新特性和 bug 修复时也将更加果断。



建议

#8

熟悉你的 KPI

产品和开发 KPI 的设定是帮助管理层跟踪开发进度。然而 KPI 也可能是错误的进展指标。过度依赖衡量指标，可能会无意中鼓励开发者朝着指标本身工作，指标成为目标，而不是实现目标的工具。

例如，测试中的代码覆盖率很重要，但是如果作为唯一的 KPI，可能会导致所有代码只执行了一些简单的测试，而不是真正需要的场景。

设定更接近实际目标的衡量指标，帮助团队更快地实现这些目标，而不只是设定 KPI，因为这可能会误导他们，让工作偏离正轨。



关于 Incredibuild

Incredibuild 是业界第一个混合加速平台，推进编译、CI/CD 构建、测试等开发进程。依靠强大的进程虚拟化技术以及独特的构建缓存加速解决方案，Incredibuild 可加速开发周期，提高迭代频率，从而改善产品质量，缩短上市时间，提高客户满意度。同时，显著降低预发布和云计算的成本。

Incredibuild 也是 Visual Studio 绑定的唯一商业工具。Incredibuild 拥有来自 2500 家全球企业的 25 万多个用户，其中包括 20 家《财富》100 强公司。了解更多 Incredibuild 的信息，请访问官网 incredibuild.cn