

[提问](#)[实验台边...](#)[首](#) [话题](#) [发现](#) [消息](#)

1202

[« 去我的收藏](#)

Office skill

[添加评论](#) • [编辑](#) • [删除](#)

停止学习框架

方应杭， 我司小姐姐的微信hungervalley， 我司公众...

1202 人赞

这是一篇译文，[原文](#) 在 Hacker News 上获得接近 500 个点赞。

每过几年都有类似的文章出现，然而程序员却依然疲于学习新的框架，看完此文希望对你有所启示。

那么，译文开始。

我们是程序员，每天都在了解最新的技术，每天都在学习编程语言、框架和库。

因为我们知道的现代编程工具越多越好，对吧？

不停地追随 Angular、React、Vue、Riot、Ember、Knockout 的脚步还真是一件有意思的事情呢。（译注：反话）

但这其实是在浪费时间！

时间是人类最宝贵的资源。时间是有限的、不可再生的，你可以用钱买任何东西，却买不了时间。

技术，就像时尚，在以光速在变化着。为了赶上它，我们需要跑的非常快。

但是这个跑道上没有终点，所以没有赢家。



华尔街之狼

我的导师曾经这样教我：

导师：艾德，你在做什么？

1202

我（自豪地说）：我在读一本关于如何使用 GWT 构建现代 Java 应用的书呢。

导师：你读它做什么？

我：作为一名 Java 开发者，我需要跟上潮流。GWT 就是现在的潮流。

导师：你在读这本书之前还读过什么书？

我：我读了一本关于 Apache Tapestry 的书，那本书有 500 页。Apache Tapestry 是之前的潮流。

导师：Apache Tapestry 现在还是潮流吗？

我：不是了，GWT 才是。

导师：你之前从 Tapestry 学到的技能现在还能用吗？

我：不能用了呀。

导师：Tapestry 能帮助你更好地理解 GWT 吗？

我：不能。不过两者都用到了一些设计模式。

导师：那就是设计模式了，设计模式能帮你解决你遇到的问题吗？

我：可以，而且帮助很大。

导师：新事物来了又走，其实有很多共同点。你应该学你该学的。你应该把你 80% 的学习时间用在学习基础上，剩下 20% 的时间才是用来学习框架、库和工具的。

我：哦……只留 20% 的时间学习框架、库和工具？

导师：是的。你在工作中解决问题时自然会学会框架、库和工具。

我：谢谢指导。

导师：你之后还会谢我的。

导师的建议改变了我的生活。我把书架上关于框架的书全部都扔了，五十本书一本不剩，扔得我很开心。

我买了一些不会过时的书，并用 80% 的学习时间来读这些书：

- 程序员修炼之道 The Pragmatic Programmer
- 代码整洁之道 Clean Code
- 程序员的职业素养 The Clean Code
- 领域驱动设计和实践 Domain-Driven Design
- 测试驱动的面向对象软件开发 Growing Object-Oriented Software, Guided by Tests
- 持续交付 Continuous Delivery

我只买了一本关于最新技术的书，是关于 Spring 的。因为根据林迪效应，学习 Spring 是一项不错的投资

林迪效应认为，对于不会自然消亡的事物，如一项技术或一个想法，其预期寿命与其当前的生命成正比，只要这一事物多存活一天，就意味着其预期寿命会更长一些。

一项技术在市场上存活得越久，就越值得我们投资（学习）它。

不要急着学习新技术，因为这些技术很可能会死。

时间会告诉你答案，你要学会等待。

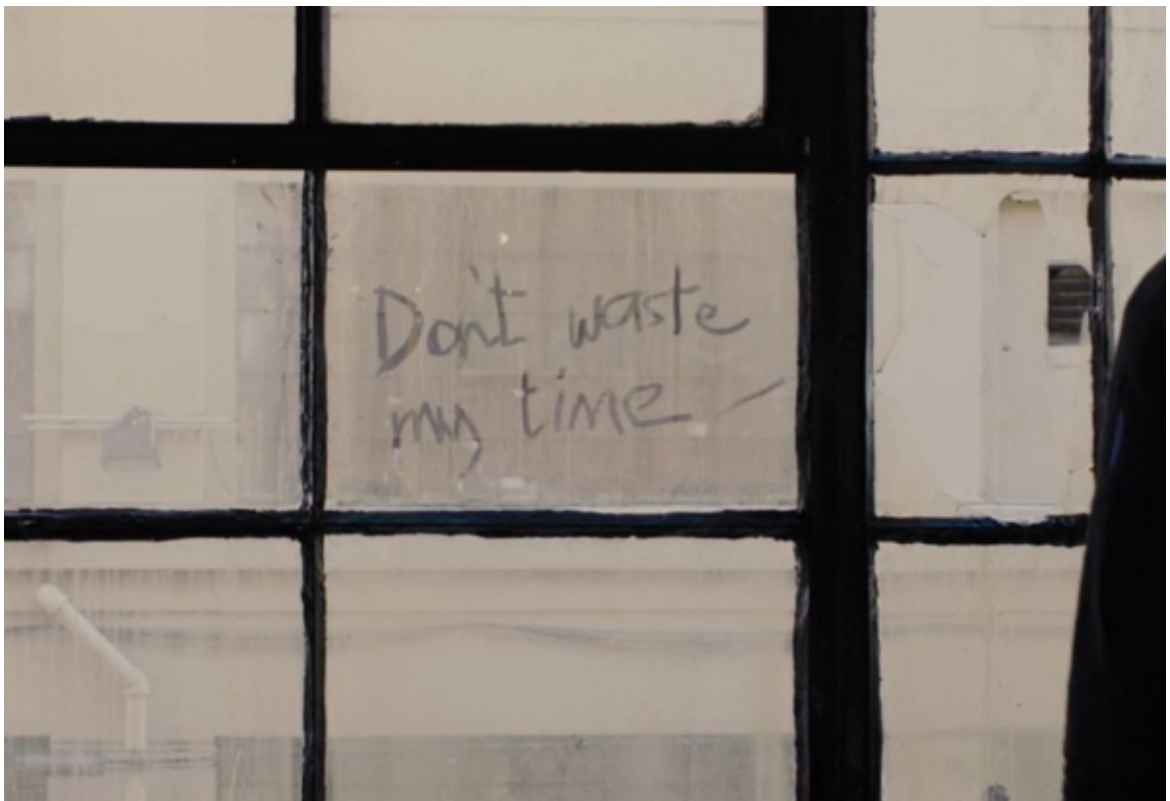
1202

十年来，我参与开发过 50 个不同的软件项目。得益于我导师的建议，我学的所有东西都适用于不同的公司、团队和领域。我的知识今天仍然有用。我没有浪费我的时间。

如果你看得更深入些，你会发现所有的软件项目都是类似的：

- 用的编程语言虽然不一样，但是设计方法是类似的。
- 用的框架虽然是不一样的，但是设计模式是类似的。
- 参与的开发者是不一样的，但是如何和这些人打交道是不变的。

记住，框架、库和工具来了又走。时间才是珍贵的。



© In Time (2011) by Andrew Niccol

将你的黄金时间用于学习通用技能，那些不会过时的技能。

- 不要学习微服务框架，学习演进式架构（Evolutionary Architecture）。
- 不要学习新的编程语言，学习代码整洁之道、设计模式、领域驱动设计（DDD）。
- 不要学习 LeSS 和规模化敏捷框架（SAFe），学习精益生产原则（Lean manufacturing principles）。
- 不要学习 Hystrix，学习容错模式（Fault Tolerance Patterns）。
- 不要学习 Docker，学成持续交付。
- 不要学习 Angular、React 和 Vue，学习 Web、HTTP 和 REST。

热门评论：

我同意你的大部分观点，但是我觉得你不用这么坚决地不学习一些东西。

「学习工具」与「学习它所蕴含的设计模式」并不互斥。

2007 年的时候我曾经试图搞清楚到底什么是「数据 ¹²⁰² 及怎么使用它，这是当时流行的 ORM 概念。我向别人问了一堆关于 NHibernate（译注：一个面向.NET框架的对象关系映射解决方案。主要用来把对象模型表示的对象映射到基于SQL的关系模型数据结构中去）的问题，很多人都回复我说「你应该先搞清楚原理，而不是学习这个工具」。但我心里想的是，shit，不行啊，因为我需要通过大量的实践才能理解这些原理啊。这是我学习的重要途径。

所以我觉得学习这些蕴含了丰富原理的工具其实是非常有用的。

同样的道理对很多工具都适用。比如 React，如果没有 React 谁能理解虚拟 DOM 呢？

不过我基本同意你的论点，但是过分强调不要学习工具就有一点何不食肉糜的意味了。

另外，Docker 也不仅仅是持续交付，「学习新的编程语言」和「学习设计模式和 DDD」也不是互斥的，Angular 最难的部分也不是 Web 和 HTTP，最难的是学习 Angular 提供的这些傻傻的工具和工作流（我不是很喜欢这些玩意）。

作者的回复：

看来我们达成了共识——学习基础常常意味着深挖某个框架、库或者工具。框架和基础都要学习，但是优先级必须是基础高于框架。

我的观点：

假设你面前有两个应聘者，一个对框架特别熟，但是对基础知识一点都不懂；另一个对框架一点都不熟，但是基础知识特别懂。你会雇佣谁？

小公司雇佣前者，能用就行。大公司雇佣后者，能堪重任。

154 条评论 分享 · 举报 · 去往文章页

怎么养成一张干净的脸？

33K minno， 公众号/微博：是minno 微信：minno8（日...



脸算干净，从不用粉底液以及bb霜之类的，几乎出门都是素颜状态（防晒为最后一层）最近皮肤状态也挺好（刚脱敏，万分感激）所以来分享下一些平时的护肤习惯还有心得～（阳光下的皮肤状态，iPhone前置拍的没有磨皮，鼻子也有点黑头，整体皮肤没什么其他问题，... [显示全部](#)

关注问题 1976 条评论 · 作者保留权利

取消收藏

java如何搭稳基础？

1548 Java3y， 原创技术公众号：Java3y

很简单，没事去看看校招考什么题目，如果你能大部分答上。那说明你Java基础已经很不错了。Java基础有好几个重点的地方，下面我给你列一下我做过的原创笔记。集合集合Collection总览 List集合就这么简单【源码】 Map集合、散列表、红黑树介绍 HashMap就是... [显示全部](#)



▲
1202

关注问题 35 条评论 · 作者保留权利

取消收藏

关于创建者



实验台边吃早饭

#来自墨尔本的追梦者#五颗柠檬...

收藏夹状态

最近活动于 2019-03-01

举报收藏夹

热门收藏

换一换

有营养的下饭菜

105 人关注 · 38 条内容

育儿、生活经验感悟

117 人关注 · 235 条内容

收藏了我觉得有意思的东西

321 人关注 · 118 条内容

五叔购物回答辑〔公众号：拾无缺〕

40757 人关注 · 153 条内容

更大的世界

3939 人关注 · 188 条内容

