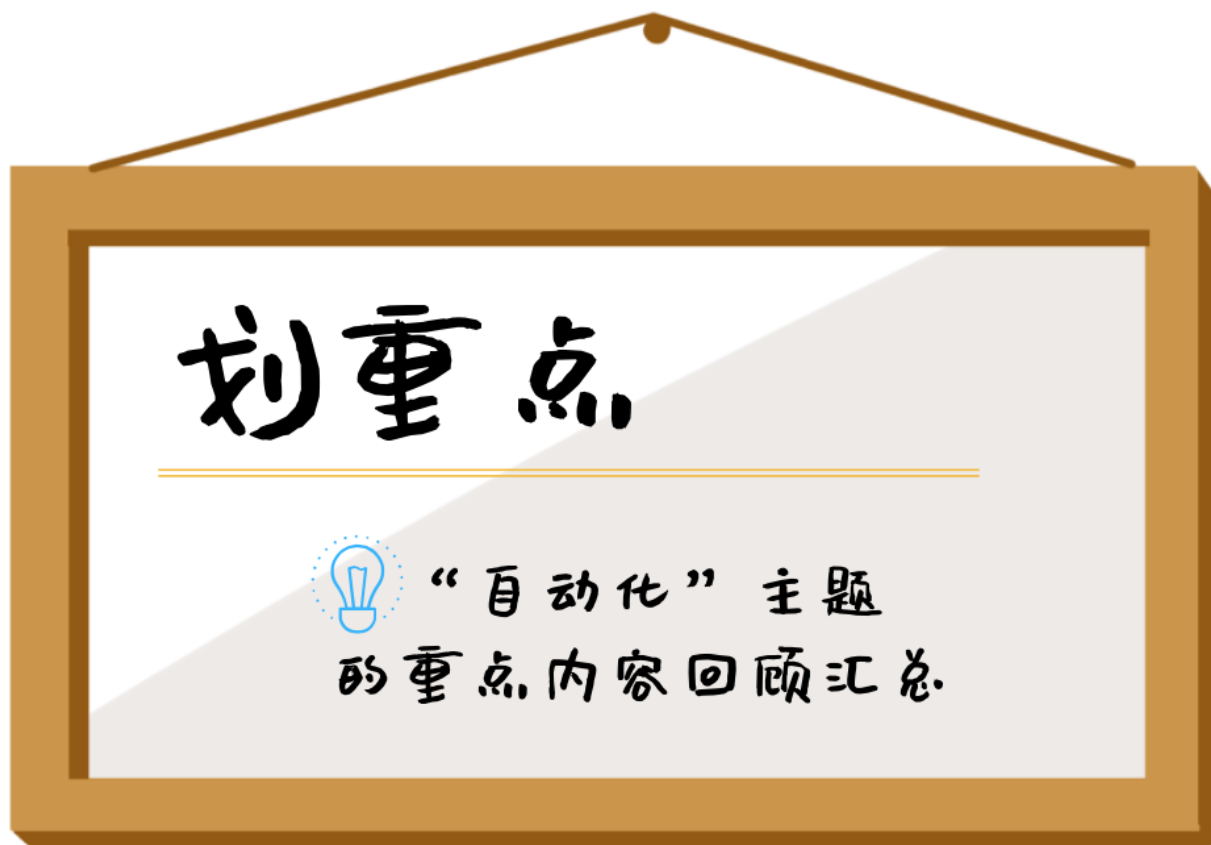


划重点 | “自动化”主题的重点内容回顾汇总

2019-04-10 郑晔

10x程序员工作法

[进入课程 >](#)



讲述：郑晔

时长 00:37 大小 585.62K



“自动化”模块终于全部更新完毕。至此，四个工作原则我已经给你全部介绍了一遍，相对而言，这个模块的内容比较“硬”，我也竭尽全力帮你串起更多知识的脉络，所以，信息量也是非常大的。希望你能够找到自己接下来努力的方向，不断提升自己的“硬实力”。

重点复习

在这个模块中，我们学习到了一些最佳实践。

持续交付

将生产部署纳入了开发的考量。

持续交付的基础设施通常包含持续集成环境、测试环境、预生产环境和生产环境。

构建流水线保证到了下游的交付物一定是通过上游验证的。

随着 Docker 的诞生，交付由发布包变成了 Docker 镜像。

DevOps

将开发和运维结合到一起。

环境配置工具上的进步，让基础设施即代码成了行业共识。

验收测试

验收测试要站在业务的角度编写。

BDD 是一种编写验收测试的方式。

Given...When...Then... 的描述给了一个描述业务的统一方式。

写好验收测试，需要构建测试模型。

SOLID 原则

设计模式背后的道理。

单一职责原则（Single responsibility principle, SRP）。

开放封闭原则（Open-closed principle, OCP）。

Liskov 替换原则（Liskov substitution principle, LSP）。

接口隔离原则（Interface segregation principle, ISP）。

依赖倒置原则（Dependency inversion principle, DIP）。

用好单一职责原则，前提条件是看待问题颗粒度要小。

DDD

它将思考的起点拉到了业务上。

DDD 分为战略设计和战术设计。

微服务

做好微服务的前提是划分好限界上下文。

微服务的第一步，不要划分微服务。

在这个模块中，我们还了解了一些重要的思路，让我们把工作做得更好。

程序员的三大美德：懒惰、急躁和傲慢（Laziness, Impatience and hubris）。

小心 NIH 综合症（Not Invented Here Syndrome）。

写好构建脚本，做好项目自动化。

参照 Java 知识体系，学习运维知识。

软件设计最基础的原则是“高内聚、低耦合”。

分层架构是一种设计上的分解。

不同业务量的系统本质上不是一个系统。

采用简单技术解决问题，直到问题变复杂。

实战指南

请谨慎地将工作自动化。

——《[29 | “懒惰” 应该是所有程序员的骄傲](#)》

将你的工作过程自动化。

——《[30 | 一个好的项目自动化应该是什么样子的？](#)》

有体系地学习运维知识。

——《[31 | 程序员怎么学习运维知识？](#)》

将部署纳入开发的考量。

——《[32 | 持续交付：有持续集成就够了吗？](#)》

将验收测试自动化。

——《[33 | 如何做好验收测试？](#)》

把函数写短。

——《[34 | 你的代码是怎么变混乱的？](#)》

构建好你的领域模型。

——《[35 | 总是在说 MVC 分层架构，但你真的理解分层吗？](#)》

用简单技术解决问题，直到问题变复杂。

——《[36 | 为什么总有人觉得 5 万块钱可以做一个淘宝？](#)》

学习领域驱动设计。

——《[37 | 先做好 DDD 再谈微服务吧，那只是一种部署形式](#)》

额外收获

在这个模块的最后，针对大家在学习过程中的一些问题，我也进行了回答，帮你梳理出一个思路，更好地理解学到的内容：

持续集成的延伸。

持续集成完成系统集成。

持续交付完成可部署上线。

“持续验证” 完成产品想法验证。

AB 测试，用一个软件的多个版本验证想法。

Selenium 用以完成浏览器的自动化。

熟练使用快捷键。

——《[答疑解惑 | 持续集成、持续交付，然后呢？](#)》

留言精选

在讲到 “懒惰” 应该是所有程序员的骄傲时，jxin 同学提到：

有价值的事并不局限性于事情本身。做自动化很重要，写代码很重要。但根据现有情况判断是否需要自动化，是否需要写代码也很重要。有的放矢，任务分解。权衡跟设计是件很艺术的事情，令人着迷。

另外，关于持续交付，Jxin 同学也提出了自己的理解：

分而治之是解决复杂问题的一大利器。持续交互就像重构中小步快走（每次微调后运行测试代码验证），都能保证大工程的稳步前进。同时由于单元小了，所以也灵活了，持续交互可以结合最小产品的理念，以小成本做 test，收集数据后，即时调整产品发展方向。

关于软件设计，毅 同学分享了自己的感悟：

我们常说任务到手不要着急去做，要从设计入手，把时间多花在前面。工作中发现大家都是思考了才动手的，那为什么越往后偏差越大呢？

共性原因有二：一是全局观不够，用咱们课里的话说就是上下文局限和反馈延迟（看到问题不提，直到代码写到那绕不过去了再沟通）；

二是没有领域的概念和有意识地去实践（纸上谈兵），尤其是做流程型任务，都喜欢先把表结构定义出来，再去生成实体，所以从领域层面来看这些实体就很不合适了。结果必然是用面向对象的工具写出了面向过程的代码，既然是面向过程那 OO 设计原则就鲜有用武之地了。这两点也是我个人理解要做好软件设计的两个必要条件。

讲到分层架构时，desmond 同学提到：

学了 REST 和 DDD，感觉两者有相通的地方：两者都以数据（一个是资源，另外一个领域对象）为中心，并制定一套标准的数据操作（一个是 HTTP Verb，另外一个我项目主要用 JPA 这一套）；而核心是业务建模。

对于微服务的理解，风翱 同学提到：

公司说我们的开发方式是敏捷开发，实际上只是使用了一些敏捷开发的方法，只有遵守敏捷开发的价值观和原则，才能算是敏捷开发。微服务也是一样，不是说拆分成多个服务去部署，就叫做微服务。也不是采用市面上常用的微服务框架，就是微服务了。

对于一个好的项目自动化应该是什么样子这个问题，西西弗与卡夫卡 同学提到：

设想过这样的情景（还没实现，打算实践一把）：我们新招一名比较熟练的程序员，从 TA 入职拿到机器，到开发示意代码，再提交 SCM，然后 CI/CD，再发布到线上交付给用户，整个过程可以在入职当天的午饭之前完成。

这不光要求构建和集成自动化，甚至要求从入职开始的各个环节都能提前准备好，包括机器、开发环境、线上环境等，甚至连示范的需求都要能及时传

递给 TA。理想情况下，程序员只需要开发好程序，保证质量，提交到 SCM 即可，其他事情都应该交给机器。

要知道程序员都很贵，越早给用户交付价值越好。

对于自动化验收测试，shniu 同学分享了他的学习感悟：

自动化验收测试确实是很好的东西，比如在回归测试，省去了很多的重复工作。但我理解 BDD 的初衷是驱动产品、业务、开发、测试等去深入讨论沟通需求，在还没有真的写代码的时候去实例化 story，并一起定义验收用例，让每个人对需求的理解都很透彻，当然特别注意的是要从统一的业务角度去描述，可见，真的做好 BDD 是需要不断的尝试和总结的。

对于“5 万块做淘宝”这个话题，enjoylearning 同学提到：

做一个淘宝那样的，客户指的是业务类似，但用户量多少，需要多少并发数，搜索性能等如何都是需要跟客户沟通后才能决定技术选型的。现实中我们的有些系统已经满足了业务需求，就没有必要为了追求技术复杂度而去拆分了，只有面向问题技术选型才会有成效。

关于运维知识，hua168 同学对文章内容进行了补充：

现在运维流行 DevOps，高级一点就是 AI，其中一篇文章《DevOps 详解》不错，链接如下：

<https://infoq.cn/article/detail-analysis-of-devops>

《DevOps 知识体系与标准化的构建》也不错，下载地址：

<https://yq.aliyun.com/download/778>

运维知识体系：

<https://www.unixhot.com/page/ops>

Web 缓存知识体系：

<https://www.unixhot.com/page/cache>

感谢同学们的精彩留言。在下一个模块中，我将结合具体的应用场景，将之前讲过的“思考框架”和“四个原则”进行综合应用的分析。

综合应用主题预告

新入职一家公司，怎么快速进入工作状态？

——找到关键点，快速上手

面对遗留系统，你应该这样做

——用思考框架应对遗留系统

我们应该如何保持竞争力？

——不断提升自己的核心优势

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。

 极客时间

10x 程序员工作法

掌握主动权，忙到点子上


郑晔

火币网首席架构师

前 ThoughtWorks 首席咨询师

TGO 鲲鹏会会员



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

上一篇 [答疑解惑 | 持续集成、持续交付，然后呢？](#)

下一篇 [38 | 新入职一家公司，怎么快速进入工作状态？](#)

精选留言 (6)

写留言



刘晓林

2019-04-19



郑老师在专栏中推荐了很多非常好的书籍作为参考，可否考虑在某一期中，将这些参数书籍整理成一个书单，按照专栏的主题做个小分类，然后每本书简单点评两句作为领读内容。希望在专栏的结束语之前可以看到这个书单，嘻嘻。

展开 ∨

作者回复: 马上就来!



enjoylear...

2019-04-15



我就奇怪了，用的intelliJ 用户gradlew编译成功，可每次用IDE自己build菜单执行总是报lombok解析错误，也按照作者说过的设置过了，还查了相关lombok plugin issue网上的回答,设置方案跟作者提到的一样。这个问题虽然很小白，但困扰了我一段时间了

作者回复: 这就只能怪人品了 :)



hua168

2019-04-10



老师，“高内聚、低耦合”听了很多了，但都是一个标题，能不能简单说一下，怎么能写出“高内聚、低耦合”的代码，我理解的就类似功能的写在一个模块中，这样就是“高内聚”，独立性强；模块之间用REST-ful API或public类调用就实现了“低耦合”这样理解会不会太简单？

展开 ∨

作者回复: 前面讲过的设计原则，就是帮你写出高内聚低耦合代码的方法啊！



小白菜

2019-04-10



哈哈，早上好

展开 ∨

编辑回复: 早上好 ☺



捞鱼的搬砖...

2019-04-10



坐等更新！ 不看睡不着

展开 ∨

作者回复: 休息好才能工作好。



西西弗与卡...

2019-04-10



强迫症一下，里面我的id为啥不是橙色的，哈哈

展开 ∨

编辑回复: 可能是我设置问题，我去改一下，然后面壁思过五分钟 ☹

