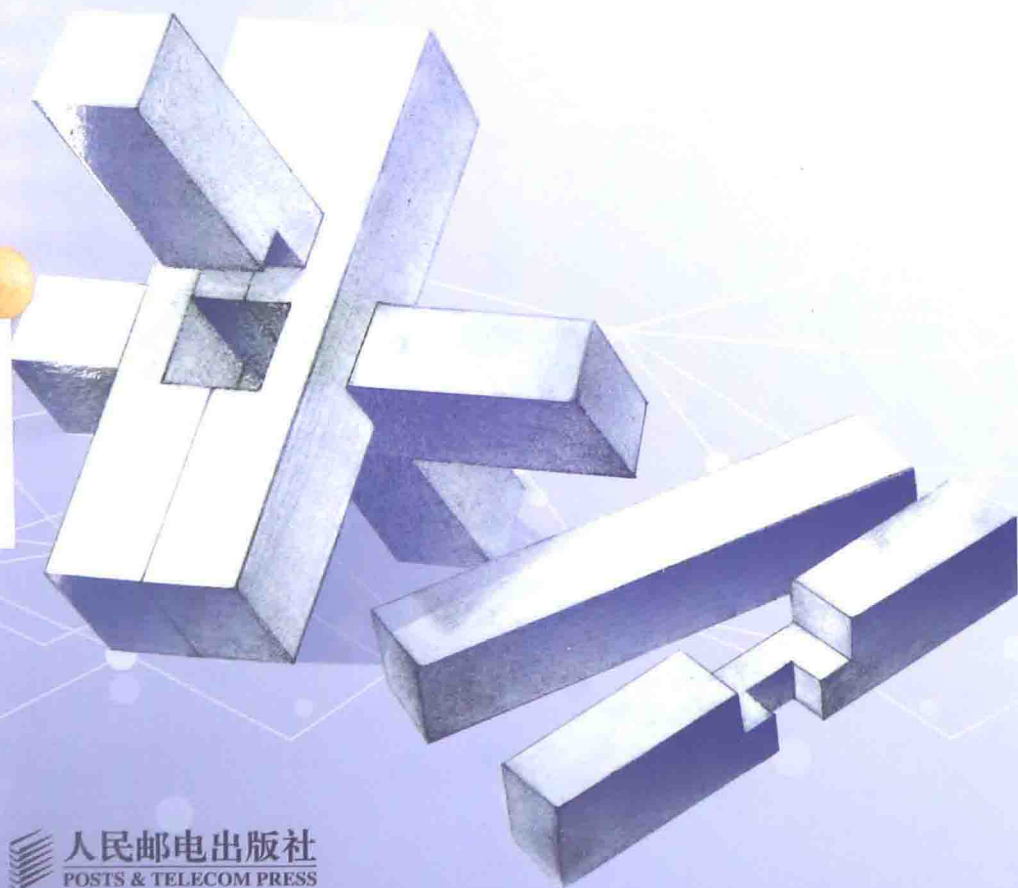


现代软件工程

构建之法

邹欣 著

JUST { PUB



人民邮电出版社
POSTS & TELECOM PRESS

JUST { PUB

构建之法

现代软件工程

Build To Win

Effective Software Engineering

邹欣 著

人民邮电出版社
北京

图书在版编目 (C I P) 数据

构建之法：现代软件工程 / 邹欣著. — 北京：人民邮电出版社，2014.9
ISBN 978-7-115-36916-1

I. ①构… II. ①邹… III. ①软件工程—教材 IV. ①TP311.5

中国版本图书馆CIP数据核字(2014)第189122号

内 容 提 要

软件工程牵涉的范围很广，同时也是一般院校的同学反映比较空洞乏味的课程。但是软件工程的技术对于投身 IT 产业的学生来说是非常重要的。作者邹欣有长达 20 年的一线软件开发经验，他利用业余时间为数所高校进行了长达 6 年的软件工程教学实践，总结出了在 16 周的时间内让同学们通过“做中学 (Learning By Doing)”掌握实用的软件工程技术的教学计划，并得到高校师生的积极反馈。在此基础上，作者对软件工程的各个知识点和技能要求进行了系统性整理，形成教材。

本书共分 17 章对照美国 ACM/IEEE2013 年新出版的计算机科学教学指导 (Computer Science Curricula 2013) 中的软件工程相关部分。这本教材覆盖了其中大多数 Core-Tier1 和 Core-Tier2 的内容。可以说，全书对软件工程内容的覆盖不逊于任何一本现行的教材，同时讲述了业界最新实践方法。



-
- ◆ 著 邹欣
 - 责任编辑 陈冀康
 - 审稿编辑 李琳骁
 - 版式编辑 胡文佳
 - 策划编辑 周 筠
 - 责任印制 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京铭成印刷有限公司印刷
 - ◆ 开本：720×960 1/16
印张：24.75
字数：428 千字 2014 年 9 月第 1 版
印数：1 - 4 000 册 2014 年 9 月北京第 1 次印刷

定价：49.00 元

读者服务热线：(010) 81055410 印装质量热线：(010) 81055316
反盗版热线：(010) 81055315

推荐序

我和邹欣大约是在 2009 年认识的。当时北航软件开发环境国家重点实验室的同学要学习各种软件开发环境，我们就通过微软亚洲研究院的高校关系部经理马歆找到邹欣，请他给我们介绍微软软件项目管理环境 Team Foundation Server，并给我们的学生讲一讲《现代软件工程》这门课。

2009 年秋季学期，邹欣如约在北航开讲《现代软件工程》，我们从北航计算机学院大三的学生中抽出一部分学生上这门课，其他学生继续上常规的《软件工程》课。为了更好地测评授课效果，我们还在两个班级收集了数据，对比同学们的学习时间、代码量，以及在专业技能、职业技能方面的收获。2012 年，在北航计算机学院高小鹏老师的支持下，邹欣老师和罗杰老师合作，再次讲授该课程。两个学期的数据表明，《现代软件工程》采用的“做中学”的教学方法和面向实战、超大量的项目实践给学生带来了明显的帮助，不但让基础好能力强的学生如虎添翼，基础一般的学生更是从中获益，能力得到了显著的提升。2013 年秋季学期，罗杰老师正式开始独立讲授这门课，继续北航的软件工程教学改革。邹欣老师则带着对软件工程教学极大的热情，针对教学体系中的薄弱环节，又开设了《现代程序设计》选修课，继续他对软件教育的探索。

最近十多年来，软件产业和互联网产业的迅猛发展，给众多计算机和软件专业的学生们提供了用武之地，同时也对大学软件工程的教育提出了很大的挑战。经典的软件工程理论和模式虽然仍有其价值，但是国外的教师和业界人士一直在探索新的理论和最佳实践。我们中国大陆的教育工作者要更接近实际，从工业界中汲取生动活泼、行之有效的工程技术和方法论；在教育方式上要多向实践、

实训靠拢，探索 MOOC、翻转课堂等新形式。改变通常会令人不悦，很多老师希望用非常“稳定”的教案教学，很多学生希望课程“好通过”。我也听说在上《现代软件工程》这门课的时候，很多学生抱怨作业比别的课程多，还要写博客……然而，他们中的很多人在课程结束后，却给了这门课很高的评价。看来，在评估教学效果时，我们应该“风物长宜放眼量”，看看学生在课程结束之后，特别是走上工作岗位后，是否从课程中获益。

我很高兴看到这本书的出版，希望这本生动活泼的教材能引起大家对高校软件工程教育的讨论和改进，希望更多的新方法、新思路能出现在计算机教学的讲台上。



2014年8月于北京

前言

我在高中一年级的時候（1984年）接觸到計算機語言和計算機，後來在大學本科和碩士期間讀的也是軟件專業。在二十年的職業軟件開發過程中，我參與和領導過很多項目，有些是幾個星期開發出來的內部演示項目，有些是大公司內部的專用工具，有些是長期的面向全世界用戶的產品。從產品平臺上看，我做过 Unix/PC/Web/Mobile 等類型的項目；從團隊類型來看，我經歷過國內大學的校辦軟件團隊、國內創業團隊、跨國公司企業內部工具開發團隊、商用軟件團隊、研究機構技術轉化和創新團隊以及互聯網產品團隊。不同項目的挑戰各不相同，結果也不盡如人意，但是我一直覺得軟件工程是挺有意思的事情。我從 2003 年加入 TFS¹ 項目之後，就萌發了向開發者社區介紹現代軟件工程思想和實踐的想法。2005 年回到微軟亞洲研究院之後，我參與了對實習生的培訓和對外的教學合作，在這些實踐的基礎上，我在 2007 年出版了《移山之道——VSTS 軟件開發指南》，它是國內第一部介紹微軟開發工具和理論的原創書籍。書籍出版後，我徵求了一些高校老師的看法，他們覺得這本書很獨特，很活潑，但是“不好教”。

軟件工程牽涉的範圍很廣，軟件工程的內容對於投身 IT 產業的學生來說是非常重要的。但是這門課是大學學生們普遍反映比較空洞乏味的課程²。造成這種結果的原因有不少，如教材過時且偏重理論，老師缺乏實際項目經驗，教學方法陳舊等。經過 2007 年以來的探索，我總結了在 16 周內讓同學們通過“做中學（Learning by Doing）³”掌握實用的軟件工程技術的教學計劃。這本書就是這幾年探索的匯報，也算對上面“不好教”的一個回答。

本書的內容在下面的學校正式課程中完整地運用過：

2007—2010 清华大学理论计算机科学研究中心，主要是大四上学期（学生 20—30 人）

2009，2012 北京航空航天大学计算机系，大三上学期（学生 60 人）

2010—2012 微软亚洲研究院创新人才班，大四上学期（学生 20—30 人）

本书内容有以下特点。

- 理论和实践相结合。讲现代理论，同时讲体现理论的工具。
- 结构紧凑。个人项目 / 结对项目 / 团队项目紧密配合，能在 16 周讲完。
- 面向实战，强调做中学（Learning by Doing）。学生项目都通过团队博客实时公布项目进展；工程项目都公开发布；用户数量和反馈是项目重要的评价标准。
- 讲人在软件工程中的不同角色和作用（团队的角色，不同角色的技术能力和职业能力）。
- 有丰富的材料给教师和助教使用⁴。
- 练习量大，学生工作量和国际一流大学相仿。补充内容多：参考教材（3 本），参考书（20 本）。

尽管本书介绍了不少业界正在使用的理论和技术，不过，本书的目标并不是介绍所有的新思想和新技术。上个世纪末，有人问软件工程专家戴维·帕纳斯（David Parnas）：将来会有什么令人兴奋的软件工程技术出现？他回答：最有用的技术不在将来，而是已经出现好些年了，只不过我们没好好用。本书就是想让学生在一个学期内把一些软件工程的技术好好地用一下。

书中人物介绍

软件开发是一件很愉快、很有意思的工作，为什么高校的同学觉得软件工程特别乏味呢？一个很重要的原因是教材只是干巴巴地讲述理论和原则，脱离了“人”这个重要因素，因此的确很乏味。我在《移山之道》这本书中，创造了一个虚拟的环境：王屋村软件学院、移山公司和一些人物（阿超、果冻、小飞、小李等），希望通过人物的对话和活动，把软件工程的丰富内容生动地展现出来。这本书也沿用了《移山之道》的一些人物，并且扩展了他们的故事。下面是人物介绍，他们都是大学生、研究生或者刚工作几年的技术人员，读者可以从他们身上看到各自生活中熟悉的形象。



阿超：有几年实战经验的项目带头人



国栋：外号叫“果冻”，喜欢引经据典，对知识有些消化不良



小飞：两年编程经验，对任何事都有自己的看法，爱唱歌，喜欢足球和军事



小李：有几年项目管理经验的产品经理

致谢

在教学、构思和写作的过程中，我得到了很多老师，同行和各方面专家的鼓励和帮助。我在写第一本书《移山之道》的时候，我告诉当时的领导沈向洋博士说我以后计划写《编程之美》，还可能写写软件开发，估计叫《构建之型》……我自己都觉得想法很缥缈，他倒是表示不妨一试，让我有更多信心。清华大学的姚期智老师在我没有任何大学教学经验的时候就支持我去教软件工程课。北航的李未老师也很信任我，鼓励我去北航实践我的教学方法。清华大学软件学院的刘强老师、北京大学的张铭老师和北京理工大学的金旭亮老师还请我去和他们的学生交流经验。在本书审读过程中，北航的吕云翔老师和罗杰老师、天津大学软件学院

的王赞老师、浙江大学计算机学院的陈越老师、复旦大学软件学院的黄萱菁老师、南通大学软件学院的鞠小林老师、中国科技大学软件学院的孟宁老师、哈尔滨工业大学计算机科学与技术学院的王忠杰老师等都提出了很好的反馈意见。在两次北航讲课的基础上，北航计算机学院的高小鹏老师和罗杰老师正基于本书的内容继续推动软件工程的教学改革，他们是真正身体力行的改革者。微软亚洲研究院高校关系部的经理马歆、吴国斌在过去的几年中一直大力支持我在实习生和学校中开展软件工程教学探索，在这里一并致谢。

这本书的有些实战故事和经验是来自我在微软的团队，有些案例经过了一些改编。任何团队都会有这样那样的问题，正视这些问题，不断改进，正是一个优秀团队应该做的。特别要指出的是，必应团队的用户体验设计师高霖和项目经理徐萌对本书的需求分析、交互设计和 PM 的内容提出了专业的意见。

这是我和本书的出版人、来自独立出版团队 JUSTPUB 的周筠老师合作的第三本书，她是我唯一的出版人，同时也是最好的一个。JUSTPUB 的特约审稿编辑李琳骁和设计师胡文佳为提高本书的质量和阅读体验做了很多工作，人民邮电出版社的陈冀康、刘涛、俞彬等多位编辑为本书的如期出版做了积极的协调配合。没有他们的努力，这一本书还是十几篇风格散乱的博客文章。胡文佳还为本书设计了封面和整体装帧风格。封面的素材（鲁班锁）和书名题字都来自设计师高霖的艺术之家。

最后，要感谢我的家人，他们容忍我一个人在厨房里对着电脑长时间发呆，有时还给我煮绿豆汤喝！

对我来说，这几年教书的过程也是一个学习的过程。同学们给了我很多反馈，我还学习了不少好老师的建议⁵，还有些教课的心得⁶，也对中国大学的 IT 教育有些反馈⁷。近两年高等教育有不少创新的尝试⁸，我希望这本书也能在 IT 教育改革中发挥一些作用。

邹欣

2014 年 8 月于北京

-
- 1 TFS: Team Foundation Server, 微软公司出品的项目管理软件。2006 年正式发布第一版, 是微软 Visual Studio 产品的一部分。
 - 2 请看大学生们在微博上对软件工程课程的意见:
<http://www.cnblogs.com/xinz/archive/2013/02/06/2908169.html>
 - 3 所谓“做中学”的办法也不是包治百病的, 这篇博客剖析了各种误区: <http://www.douban.com/note/344117673/>
 - 4 参见: <http://www.cnblogs.com/xinz/archive/2011/11/27/2265425.html>
 - 5 参见: <http://www.cnblogs.com/xinz/archive/2011/12/29/2306652.html>
 - 6 参见: <http://www.cnblogs.com/xinz/archive/2012/01/15/2322913.html>
 - 7 参见: <http://www.cnblogs.com/xinz/archive/2011/12/03/2274445.html>
 - 8 参见: <http://www.cnblogs.com/xinz/archive/2012/08/25/2656822.html>

给任课老师和助教的建议

To Teacher & TA

课程安排

表 1 16 周的课程安排

周	授课知识点	实践内容	个人项目	结对项目	团队项目	助教
1	软件工程相关概念； 软件工程和计算机科学的关系； 源代码管理	简单的源代码管理操作（选择：TFS、GitHub、SVN……）	个人项目			发布助教博客； 确定团队成员， 确定结对分组， 核对个人博客和团队博客
2	程序效能分析；单元测试； 个人软件流程	单元测试	每周一篇博客			批改个人项目
3	代码质量，代码规范， 代码复审，两人合作	代码复审练习	同上	结对项目 (1)		核对团队博客账户， 队员名单
4	软件开发的各模式， 团队类型，团队成熟度模型， 敏捷流程	软件项目的估计和 WBS 练习			团队项目启动， 展现项目目标、角色分配	审核结对项目并评分
5	软件需求，竞争性需求分析的框架， 软件规格说明书 (Spec)， 基于场景的设计，功能驱动的设计 (Feature Driven Design)	项目管理工具实践（选择：TFS、Trac、看板）			展现具体场景、工作项、燃尽图	
6	项目经理，软件业的创新， 创新的迷思，创新的招数	分析和评价当前热门软件的创新			每日例会（五天）	每天审核 Scrum 报告

续表

周	授课知识点	实践内容	个人项目	结对项目	团队项目	助教
7	测试的分类和工具, 软件的稳定阶段	测试当前热门软件, 撰写测试报告, 说明你的团队应该怎么做			每日例会 (五天)	每天审核 Scrum 报告
8	各种方法论, 微软软件解决方案框架	Alpha 版本发布			Alpha 发布	组织团队互相测试其他团队的产品
9	用户体验, 认知阻力 (Cognitive Friction), 用户调研, 绩效管理 (1)	测试并评价其他小组的 Alpha 版本; Alpha 版本的回顾与反思			收集用户反馈, 团队成员评分, 成员换组	公布团队和个人成绩
10	开发流程的管理, 风险管理, 软件的发布	对项目中的 Task/Bug 进行会诊, 决定取舍和优先级			Beta 阶段目标、角色、任务、燃尽图	
11	软件流程的质量, 绩效管理 (2)	CMMI 以及软件团队合作状态测评			每日例会 (五天)	每天审核 Scrum 报告
12	效能测试, 压力测试	结合团队项目进行效能测试和压力测试			每日例会 (五天)	每天审核 Scrum 报告
13	软件团队的成熟度, 团队总结和改进	Beta 版本发布			Beta 发布	
14	软件工程师的职业道德	Beta 版本的回顾与反思	(可选) 读书报告	(可选) 结对项目 (2)		
15		项目最终汇报 (要有用户量, 最终用户到现场分享), 项目复审	(可选) 读书报告	(可选) 结对项目 (2)		审核团队的汇报、代码、博客
16						总结各个项目的得分, 并决定所有同学的最终分数

上表是一个 16 周的课程安排，在我开课时（清华大学、北京航空航天大学），每周有两个小时用于课堂授课，一个小时用于助教的答疑和处理学生作业、打分等。学生每周要花 8 个小时在上课、各种作业和项目上（这是国际一流学校的工作量）。如果老师认真教，学生做出实际项目的话，这门课程相当于 4—5 个学分的课。考虑到各个学校的具体情况，老师也可以考虑把这门课当作两门课的结合（软件工程理论和软件工程实践）；也可以安排两个学期，一个学期做 Alpha 阶段，一个学期做 Beta 阶段。但是我坚决反对把这门课的理论 and 实践拆成两个学期的课（一个学期上理论，一个学期上实践），这种安排，将导致学生上理论课时没有实践帮助理解理论，而上实践课的时候则早把理论忘光了。

大马哈鱼洄游模型

软件学院的老师抱怨，软件工程这门课看似容易，实则太难教。老师反映的情况通常有以下这些。

我是按照经典的瀑布模型来讲课的，本以为会是高屋建瓴，一泻千里，但是实际情况是下面这样。

- 需求分析：学生们都不知道需求是啥，什么是业务活动，上课睡觉。
- 设计阶段：学生们画了许多 UML 图，用设计工具画了不少矩形、菱形，如此而已。
- 实现阶段：几个学生开始讨论非常细节的问题，UML 图早已经扔到一边，测试只存在于文档中。
- 稳定阶段：学生中十分之一的人开始写代码，其他人不知道在干什么。代码大部分情况下都不能工作，设计好的种种黑箱和白箱测试都无从开始。
- 发布阶段：这个只有一天时间，就是最后检查的那一天，同时还有人在调试程序，美化 PPT。
- 维护阶段：课程结束了，同学们对自己的产品没有任何维护，放假了！

最后，大部分同学都说这门课特别没用，自己根本没学到什么东西，老师特别烂。然后下个学期，新的一批学生会重复这一过程……

事实上，在现实世界中，软件工程师的职业发展好像与瀑布流程刚好相反。

1. 毕业进入公司（或者实习生），开始学习并维护一些已有的软件（**维护阶段**），主要由自己的师傅（Mentor）带领。
2. 能够在项目中改一些 Bug，然后发布小规模的更新版本（**稳定/发布阶段**），

练习重构，开始和其他同事打交道。

3. 有机会负责重写一个较小的模块，没有多少文档，自己要写很多代码（**实现阶段**）。
4. 表现好的员工，有机会设计比较大的模块，自己写一些文档（**设计阶段**），和更多成员发生工作联系，在一些情况下还能发挥领导作用。
5. 员工逐渐成长为团队的骨干，有机会计划新的项目（**需求分析**）。

那么软件工程的课程能否也像这样安排？可以试一试下面的流程。

Alpha 阶段

1. 开始维护以前的同学开发出来的程序，理解程序，理解用户的痛点。
2. 找 Bug，改 Bug，重构小部分代码；一部分同学可以开发测试用例。
3. 在现有版本的基础上做少部分增量开发，快速发布并收集用户反馈。

Beta 阶段

1. 根据 Alpha 版本的反馈，进一步分析需求，估计实现需求的难度（这个时候应该能理解客户需求是什么）。
2. 设计 → 开发（重构）。
3. 回归测试（用到上面开发的测试用例）。
4. 发布，收集用户反馈，看看新的版本是否真的解决了用户的问题。

这个教学模式看起来像是从瀑布下方一步一步上溯到源头，然后又从源头流下去，不妨叫“大马哈鱼洄游模型”。

师生关系

首先，要明确这门课要求的师生关系是什么样的。

餐馆 / 食客？

一些学生说，我既然交了学费来上学，就像进餐厅吃饭，想吃多少，想吃什么，都是我决定。如果不喜欢，就去另一个餐厅好了。上课能这样么？在饮食行业，顾客拍拍屁股就可以离开一个餐馆。在一些学校里，是有不同的老师上类似的课程，同学们可以根据老师的介绍和师兄师姐的提醒选择适合自己的老师，但学生必须要在一定时间内做出选择（必修课），老师掌握着最后给学生多少分，学校

掌握着毕业证。所以不能把餐馆 / 食客的关系照搬过来。学生们非但不能成为有主动权的顾客，反而会被人以分数 / 学位 / 毕业证相要挟，成为下一种关系中的弱者。

老板 / 雇员？

在学校里，很多学生把自己的指导老师叫做“老板”，学生变成打工仔或打工妹。不光有大老板，还有小老板，因为大老板太忙，平时都是小老板在管理。学生虽然是“雇员”，但是并没有相应的权利，倒像是一个“长工”。这样的关系并不利于教学相长。

保姆 / 幼儿？

还有一种情况，老师像保姆一样，为学生操办一切，把课程内容煮成婴儿食品，一勺一勺地喂食。同学们有什么问题，都去找老师搞定。学生把老师反复咀嚼过的东西再咀嚼一遍，这个模式与“做中学”（Learning by Doing）的模式有很大的区别。

哥们 / 哥们？

还有一种情况是，老师和学生心照不宣一起混，“你对我好，我就对你好。”这里有一条新闻¹。

授课没营养 部分大学课堂师生心照不宣一起混

2010-11-08 10:28:00 来源：中国青年报(北京) 原帖：李·手机看新闻

核心提示：“老师与学生一起应付”，这并非大学生们学习之余的调侃之语，而是不少大学课堂的真实写照。

图1 心照不宣一起混

路人甲 / 路人乙？

很多学校有巨大的新校区，老师对着百人左右的课堂宣讲幻灯片，下课后就开车回老校区或市区的家里。老师不认识学生，也未必有精力了解具体学生的情况。学生也极少接触到老师，双方形同陌路，学生甚至很少接触到本专业的高年级学生或研究生。这种情况会恶化成下面的关系——

狱警 / 犯人？

老师想方设法让学生来上课（点名、突击考试、指纹打卡），学生则想方设法逃课。学生视上课为坐牢，巴不得早一点解放。对于一些同学来说，老师就是自己和“自

由”之间的一道障碍。但是，没有任何约束的自由真的是大学生需要的么？

说了这么多，大学软件工程专业理想的师生关系是什么？是“健身教练/健身学员”的关系。



图2 健身教练和健身学员

大家可以从各种各样的健身房中看到这样的关系。在这种关系中，是谁想提高自己的水平？是那些学员，这些学员的想法得足够强烈，他/她才会花钱去参加这样的健身活动。在健身活动中，谁要做各种运动，谁要流汗呢？是学员。谁在这个活动中对别人进行批评指导？是教练。

那为什么教练可以这样做？因为教练有下面的资源。

1. 教练是很有经验的身体力行者，并有足够的理论知识。如果我只是得到了某老师的PPT，然后就照本宣科去教健身，虽然我讲的话和一个资深教练的话没什么区别，但是学员们一定受益不多。
2. 教练有一套训练计划和各种练习方法，教练（场馆）有仪器、工具、设备，不是每一个人都打算在家里放一套各种重量的哑铃和杠铃。
3. 教练可以随时指出学员的进步和不足，给予具体指导。
4. 教练能召集到一群有相似基础的队友，这对有些类型的锻炼是很重要的。

教练和学员的关系一旦确定，就很好办了。每一个来学习的学生，都是想学好软件工程专业这门技术才来的。各人的先天条件不同，目标也未必相同。有些同学想成

为世界一流的程序员，那老师就会以世界一流的标准来要求和评价学生。

- 谁要在这门课中写代码，做实验，找需求，修 Bug？
是学生，不是老师。
- 谁要看各种与软件工程相关的书籍、博客，并定期汇报？
是学生。
- 谁给各个学生设计练习，回答问题？
老师和助教。
- 如果学生的努力低于既定目标的要求，谁会批评这个学生？
老师和助教。

有些学生说：老师，你讲得特别好，我很想提高，但是我太忙了，没时间写程序，我就是来听听……

这种情况放在健身学员的类比中会是这样：

教练，你讲得特别好，我特别想减肥健美。但我太忙了，没时间练，所以我办了卡，放在钱包里面，有时候拿出来看看……

给授课老师和助教的建议

第一条：沟通

和领导沟通，获得各位领导的支持。学校想培养什么样的学生，是世界一流，中国一流，还是本省二流？有什么样的期望，就有什么样的课程设计。我上课的学校中，它们都把自己定位为世界一流或中国一流，那老师就要用相应的标准来要求学生，否则就是不称职的²。任课老师和助教要对课程达成一致意见。明确告诉利益相关者，这门课实际负担如何，估计有多少人会不及格。

和同学沟通：开门见山，在第一堂课上花时间讲述理想的师生关系是什么。如何打分，分数如何分布。软件工程的内容很多，学生们要做很多练习和项目，贯穿全课程的线索是什么？每一次作业或项目总结，老师都可以问同学们这些问题：

1. 我学到了什么软件工程的理论、技术和教训，能帮助我成为更好的软件工程师？
2. 我在项目中是否利用软件工程的理论和技术解决了用户的需求？
3. 我是否提高了项目的工程质量，让它更容易扩展和维护？