发扬EJB，Spring思想将组件化进行到底 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/231/2021_2022__E5_8F_91_E6_89_ACEJB_EF_c104_231728.htm EJB、Spring，这不是Java界最有名的两大冤家，何以把它们扯在一起。其实Spring乃是EJB1.x、2.x的继承者，正如EJB之前的COM、CORBA。他们的思想一脉相承，那就是企业级的组件化思想，也可称之为理想！ 一、非组件化的国内软件行业 各个行业的企业总有一些核心业务，长久保持不变，新时期的新业务基本上都是围绕核心业务展开。很长时间以来，IT技术的变化与企业业务的扩展存在着很大的矛盾。当企业的新业务开展之后，如何保证原有业务稳定运行的同时，新业务能够得到IT的支持与扩展？当IT技术有重大进展后，如何保证原有业务的同时进行新技术改造？在以上两种运动中，如何重用原有的技术成果？这是每一位负责任的系统管理员、CIO与及开发商所关心的事情。遗憾的是，组件化思想及实践产生以前，这个矛盾基本上是极难解开的死结。绝大多数的做法就是重写。譬如DOS时代，很多单位都使用了单机foxbase版的财务系统，界面虽简但稳定实用；到了Windows时代，流行VB、PB，于是系统重写；再到B/S时代，系统再次重写；到最近热炒的RIA，系统是不是要再次重写？对于很多小产商的作品而言，答案肯定是Yes。 很多同道可能会说，这样正好啊？我们才可以不断地赚钱。错！ 这样的状况叫“ 低水平重复 ”，这个术语经常被国人用来痛斥社会经济领域的很多不合理现象。可惜我们这个自认为高智力的行业，很多时候就是在干这种愚事。每到技术革新，各企业要重花一次钱、重学一次操作

、重转一次数据，折腾得半死；而适应不了新技术的产商，随被淘汰的代码一同退出市场；适应不了新技术的程序员，只能转行。要想不被淘汰，就必须紧跟时代风潮，不停地把精力放在新技术上，在领会业务上花的时间太少，最后导致我们的系统与企业的业务总是差半拍。因为原先快把业务搞清楚的程序员大多升官或离职了。 笔者以为这是软件界，尤其是国内软件界混乱的根本技术原因。由于技术长期得不到积累，我们不得不一次又一次吃外国的人剩饭。 那我们终究需要怎样的软件才能解决这些问题。 其实答案早就在我们身边晃了太多年，偏偏我们视而不见。大家接个新外设，要不要换主机？加根内存条、换块显卡、声卡要不要换主板？OS是不是只能用几个特定的硬件，跑几个特定的程序？而大家在OS下写的程序，是不是在系统新版本运行不了？答案基本上是否定的。 OS可以适应全世界数以万计的程序及其发展，为什么我们的应用程序不能适应哪怕一个特定单位的变化和发展。为什么我们的应用系统到了新环境下就要重写？ 原因就在于我们大多数应用程序规范性太低、耦合度太高。 要提高规范性、降低耦合度，就要不断地设计、不断地分层、不断地抽象、不断地重构。当我们终有一日把有用和成熟的代码封装成jar或是dll，不仅自己能重用，别人也能重用的时候，代码其实才算合格。 现在大家都习惯用开源产品了，外国人热衷的就是不断地制造这样的零件（组件）或技术。而我们中国人，热衷的却是组装人家的零件和技术（一如其它涉及技术的产业）。很多外国人十多岁就做出了了不起的组件，而我们中国人却把“ 不重复发明轮子 ”这种搬来的话挂在嘴在，结果是既不制造旧轮子，更没有能力发明新轮子。很

多人从业十多年都写着乱糟糟代码。 项目，不是说东西扔到客户手上套足了钱，拍拍屁股就走人。成功的项目，要对客户负责任。就算自己退了，也该把工作交接好。前面的工作成果后面用不上，只能说前面的工作不合格。 国内应用软件界一直在走RAD的道路，一开始吃着爽，越到后面越不是滋味。不是说RAD不能用，而是说，一上来就RAD，注定被养成懒汉，早晚沦落成编码机器。RAD诱使我们逃避思考，诱使我们逃避设计，最终让我们被早早淘汰！ 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com