

软件方法、体系和过程的思考 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/141/2021\\_2022\\_\\_E8\\_BD\\_AF\\_E4\\_BB\\_B6\\_E6\\_96\\_B9\\_E6\\_c29\\_141248.htm](https://www.100test.com/kao_ti2020/141/2021_2022__E8_BD_AF_E4_BB_B6_E6_96_B9_E6_c29_141248.htm) 哲学研究的对象是物质的存在、联系和运动。软件工程研究的对象是软件技术、方法、过程和工具。近三十年来软件方法层出不穷，被实际开发所运用的软件方法曾达两三百种之巨。但我们通过对哲学研究的角度进行相关的类比，我们不难发现，这些软件方法归根结底不外乎下面三种角度。

1. 基于物质运动角度：着眼于物质本身，强调物质作为一个整体对外界作用的动态交互，在软件开发方法中体现为基于功能角度的观点。著名的方法有结构化分析方法，强调软件系统（或子系统）的输入和输出，内部对外不可见，处理时宜至上向下，逐层分解，如医学之解剖一般，化整为零。
2. 基于物质联系角度：着眼物质的存在与物质间的恒定关系，强调物质间的层次性和主体地位性，在软件开发方法中体现为基于实体（Entity）角度的观点，分析的重心为对实体的静态描述和恒定联系的界定，这种角度无视实体之间的运动交互，数据库设计的E-R方法即是该观点的典型方法。例如学生的选课系统，我们关心的是学生选的是哪门课程，而不是选课的过程如何进行的。
3. 基于物质存在状态角度：着眼物质系统的自身的存在状态，分析各种存在状态间的变迁缘由和变迁途径。在软件开发方法中常为实时领域所独领风骚，体现为状态迁移分析。常见的例子有十字路口的交通灯模型，我们通过分析灯组的状态变化来对其进行分析和仿真。近来风靡一时的面向对象方法，兼具上述的物质运动角度与联系角度的特色，诸如对象

( Object ) ， 类 ( Class ) ， 继承 ( Inherence ) 之类的概念，基于的是物质联系的角度；函数 ( Function ) 和方法 ( Method ) 之概念，基于的是物质运动的角度。我们随便举一个基于存在角度的例子，UML的状态图，它反映了单一对象的各种存在状态，因此广泛应用于实时系统的设计之中。接下来谈谈体系的问题。凡方法、体系，皆如哲学的内涵与外延。外延宽广则内涵浅，外延狭窄则内涵丰富。翻译成行业用语即：高效的体系适应范围比较窄，低效的体系适应范围广。由此断定，软件行业无一包治百病，立竿见影，药到病除的狗皮膏药体系和方法。诸多企业、项目应当考虑自身实际，借以标准，适当增删修正，以合自身病症，而不是一味照单全收。君不见如今中国的软件行业，利火攻心，ISO9000做烂了，CMM/CMMI也开始泛滥成灾。暗地高兴的只有那些兜售标准的认证企业，因为他们更关心的腰包里的钱袋。最后要谈软件过程的问题，过程离不开环境。软件开发更像是一个生态进化，我们应该把软件开发作为一个不断进化的生态体系来看待，强调各方面的和谐有序。一味追求软件过程而忽视相关的环境（行业环境，企业环境）最后的结果只能是侏罗纪的恐龙，在开发生态被破坏的同时自己亦随之消亡。所以我们常常会提到：软件过程和开发方法要结合企业自身的实际。过度的追求标准、规范最终的结果是从体力上和脑力上压倒了整个团队，继而压垮整个企业。在这里我们的意思并不是说标准和规范不重要，但不要让标准和规范成为一张白纸或是开发团队、企业的沉重负担。因此每个企业和项目团队有必要根据自身的环境、规模和资源配置选择合适的软件开发方法和过程。 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](http://www.100test.com)