

软件工程学科建设的若干问题

余金山 (华侨大学电脑系, 福建泉州)

摘 要

The importance and urgency of software engineering education are pointed out in [1]. This paper further discusses several fundamental issues about the construction of software engineering discipline. It includes the definition and qualification of a software engineer, the process-oriented and the product-oriented perspectives on software engineering, and the identification, organization and division of software engineering knowledge body.

一、引 言

在文[1]中,我们指出了软件工程教育的重要性、紧迫性和任务的繁重性;给出了软件工程的若干定义和重要特征。其中有待解决的重要问题之一是软件工程教育的课程建设问题。目前国内外尚无成功的课程模式和成熟的教材,其原因之一是软件工程领域尚处于幼年时期^[2],它的变化与发展之快令人难于预料,以致于在编写有关书籍时墨迹未干,许多内容就已过时了^[3];原因之二在于软件工程所涉及的范围太广了,它不但包含了计算机科学的基础知识,还包含了管理科学、经济学、通讯技术、人类工程学以及一般工程技术等各方面的知识,而且各个主题之间还互相交错地联系在一起^[4]。也就是说,软件工程的知识体十分庞大和复杂,其具体内容至今仍然是模糊的、高度动态的。

针对以上问题,文[3]提出了所谓的“抽象”方法,即首先给出“软件工程师”的定义,进而确定软件工程的教育范围。美国软件工程研究所(SEI)经过五年的努力,提出了一种可称之为“结构化”的方法。该方法不但能较系统地反映软件工程知识体的各个主题领域,有利于按不同的需要和教育目标组织教材,而且还有利于教学内容的更新,

跟上软件工程技术的发展并及时反映最新成果。此外,目前软件工程领域中存在两种观点:过程性观点和产品观点,是软件工程的重要概念性基础^[6],它们对软件工程教育和研究均具有重要的指导意义。本文将围绕以上问题进行介绍和探讨,希望能对我国软件工程的发展和学科建设有一定帮助。

二、软件工程教育的目标

同其它技术领域一样,软件工程的性质也是面向产品的,它的主要目标是:在合理的时间、成本约束条件下生产高质量的软件系统。或按文[7]的说法,即高效能、高质量、大规模、高灵活与低成本地生产软件。因此软件工程教育的目标应该是培养能够实现这个目标的软件开发(或生产)人员。这样的开发人员显然还有不同层次的差别,本文的内容对各个不同层次的人才培养也许有所裨益。

近年来,国外对具有一定水平的软件开发人员较习惯使用软件工程师这一称谓,并认为大学以上的软件工程教育的培养对象是软件工程师^[2-5]。下面就此做一些有关论述。

2.1 软件工程师的定义

(1) 软件工程师是具有实践软件工程所必需的各种技能、经验和训练的人员。软

件工程是指规范地应用各种工程的、科学的和数学的原理与方法能经济地生产高质量的软件^[5]。

(2) 说一个人是软件工程师,是指他/她的工作与软件开发过程的语法和语用方面有关。这里的语法与语用方面包括:各种非功能性问题,当前的技术限制,外部设计(约束)要求的记录,开发文档编写,建立并维护规格说明、设计和编码等文档的版本,从各种不同版本配置成各种产品,确认各种非功能性要求,等等^[7]。

(3) 软件工程师涉及软件系统的分析、设计、检验和测试,以及各种文档的编制,软件维护和项目管理等问题。软件工程师经常与软件开发者交替使用^[2]。

(4) 软件工程师与程序员的差别:

- 程序员通常指执行、组装和修改算法细节以及用某种程序语言编写数据结构的人。软件工程师应比程序员有更丰富的技术和经验,能了解软件工程问题的范围、目标和目的^[2]。

- 软件工程师“利用”自然法则,而程序员利用数学、计算机和计算科学的法则;软件工程师与当前的技术打交道,而程序员与总是固定不变的可计算性法则打交道。同一个人有时是一个程序员,有时是一个软件工程师,但一个好的程序员应把握这两种活动过渡的正确时机并知道何时请求计算机科学家的帮助,保证所描述对象有牢固的基础^[7]。

2.2 软件工程师应具备的知识和技能

(1) 软件工程师应能确定用户的实际需要;能选择通用的系统开发方法;分析需求后能找出和解决各种冲突;在费用、时间和工作环境的约束条件下能制定出满足各种需求的设计方案;能提出各种新的技术解法;能管理一组具有不同个性、不同训练程度、不同目标的人员,共同完成软件项目的最终总目标^[3]。

(2) 软件工程在实践中需要各种各样的科学基础知识和技能^[2]。计算机科学为软

件工程提供科学基础;管理科学为软件项目管理提供基础;经济学为资源估算、成本控制和有关的决策活动提供基础。工程师应以实用的方法平衡科学原理、经济问题和社会问题;同其它工程领域一样,软件工程师应具有熟练的工程解题技术和较强的创造力;应能针对不同的应用领域选择适当的符号、工具和技术;软件工程师在一个小组机构之内进行工作,要编写各种文件资料,因而良好的口述、书写和个人通讯技巧也是至关重要的。

(3) 软件工程师依赖并应用计算机科学的知识体,他们必须对小型的程序设计十分熟练,而且还必须受过下列内容的教育:软件开发过程、软件项目管理、需求分析、技术通讯、计算机工程、系统工程、各种嵌入式实时系统、配置管理、质量保证、形式化方法、性能分析、软件度量学、软件标准、验证与确认、安全性、人类工程和专门的应用领域^[5]。

(4) 软件工程师应熟悉下列主要内容:软件工程概貌、软件需求分析、软件设计、编码、软件测试、配置管理、软件维护,软件工具和一定的实践经验。此外还必须对软件计划、软件管理等内容有所了解^[10]。

(5) 根据软件工程的狭义定义^[1],文[8]认为软件开发可分为四个子学科:管理、工程、程序设计和理论验证,从而软件开发者可相应地划分为四类:管理者、软件工程师、程序员和计算机科学家。而各类人员的职责如下:

(a) 管理者:负责对开发的全局作出计划、估算所需的资源、预算和财务保证,分配资源(人员、机器、工具、时间)、监视和控制资源;组建开发工作所需的各种组织。管理是建立在社会法规之上的(如计量经济学、管理科学等)。

(b) 软件工程师:负责使程序设计适应现有的硬件和软件技术,调整开发工具以

适应现有的计算机系统；形成软件版本，并按版本配置成不同的产品；提供各种演变的日志记录；建立测试点、验证设计和实现，等等；此外还需建造各种所需的工具。软件工程是建立在软件科学和自然科学的法则之上的。

(c) 程序员：负责定义需求，设计结构的组成部件，描述函数功能，把描述转换成代码，以及证明以上步骤的正确性；程序员应遵循程序设计方法学的规则来构造理论，依靠计算机和计算设计的法规进行工作。

(d) 计算机科学家：负责检验程序是否表达了恰当的理论，计算机科学家构造新的计算模型和理论。理论检验是基于数学定律之上的。

文[8]还指出：“经常是同一个人来完成这四项目标”。因此，作为一个受过高等教育的软件开发人员应该对以上的内容有一个较为全面和深入的了解。

三、两种观点

当前对软件工程存在两种不同观点：过程观点和产品观点，这是软件工程的重要概念性基础，也是指导软件的科研、教学和实践的基本思想和观念^[6]。一项成功的软件工程，应该在这两个方面同时取得成功：一是成功的软件产品，二是成功的软件开发过程^[11]。

3.1 产品观点

软件工程的产品观点是一种静态观点，即把软件看成是一种独立的，由一组程序及其相关的定义文本所组成的产品，其应用的上下文是固定的且已被充分理解。产品观点从软件产品的类型（并行的、分布式的等等）和产品的一般质量要求（如可靠性、可维护性、容错等）出发考虑问题和解决问题。所采取的主要立场如下：

(1) 程序是一种形式化的数学对象，是从抽象的规格说明出发，经若干形式化的推导过程得到的。它的正确性由数学证明来

确定。

(2) 软件开发可按某种以时间顺序分阶段的线性模型来描述。软件的功能需求是可预先确定的。软件开发由最顶层的需求说明开始，然后依次对上一开发阶段的结果进行变换，直到得出所要的软件系统。软件维护是软件开发的后续工作。

(3) 软件的质量与产品的特色有关，质量特征是指程序的可靠性、效率等等，它们可能因程序的改变而受影响；质量通过验证（测试和证明相结合）确定；质量根据程序-用户的观点来定义（如：“用户友好的”，可接受性等）。

(4) 程序应该仅仅依靠有关的文件就能被理解；程序的含义是由其规格说明所给出的形式语义定义的；规格说明的重要部分是由程序所具体体现的模型，它必须是形式化的，因此不宜用自然语言来解释；知识是通过形式规则的理解而得到的，因此各种文件资料必须是完备的、相容的和无二义性的；文件资料应描述开发过程各阶段所得到的结果。

(5) 软件方法本身也是一种产品，方法的正当使用可导致统一的结果；方法用于约束通讯；方法可减轻软件开发对人的依赖；方法是静态的、定义良好的和通用的。

3.2 过程观点

过程观点是一种动态观点。它把软件的开发与使用看成与人类的学习、工作和通讯过程密切相关。过程观点从软件生命期中软件开发人员执行的各种活动出发考虑问题和解决问题。所采取的主要立场是：

(1) 程序是人们的工具和工作环境，是为了满足人类的需要，在学习和通讯的过程中设计出来的；程序只有在有控制的使用过程中通过程序的各种不同后继版本才是充分的。

(2) 软件开发是分析、设计、实现、评价和反馈等阶段之间的一个环状模型，软件是这些交错过程之总和的产物，而每次循

环重复的结果都产生产品的一个新版本；软件系统的功能需求是无法事先完全确定的，只有在开发和使用过程中才能逐步明了。

(3) 软件质量与产品的使用过程有关；质量特征是指程序使用的可靠性、效率等等，它们可能因工作环境的改变而受影响；质量通过评价（论证、试用）确定；质量根据用户-程序的观点来定义（如适合程序、充分性等）。

(4) 人员间的讨论和人员试用对理解程序是必不可少的，文件资料必须能方便这些活动；程序的含义与作者的意图和程序的各种可能用途有关；文件资料的基础部分围绕所定义的模型引入有关的概念，给出有关的定义和解释，并用自然语言来书写，形式化方法应该在需要时使用，但它对提高质量没有帮助；知识在很大程度上是通过实例、类比和已有的背景知识获得的，如果选取得当，它们将有助于增强人们对二义性的容忍程度；文件资料不但应描述开发过程中得到的各种结果，而且还应该说明这些结果是如何得到的（如经验教训、各种决策的依据等等）。

(5) 方法本身不存在，只有方法的开发和使用过程，这些过程会影响方法使用的结果，方法用于支持通讯；方法仅当它作为人们所采取的步骤时才是有意义的；方法是动态的，它与具体的应用领域有关，并随着人们的经验而演化，因此在使用时必须根据各种具体需要和具体情况进行裁剪。

四、软件工程的知识体

找出软件工程的知识体并加以描述记载是制定软件工程教育课程的必要条件。但是要验明软件工程庞大而复杂的知识体是不容易的，这需要进行大量深入的研究；需要合适的系统性方法。现在大多数方法是从分析软件工程所跨越的学科入手^[2,3]。《Computing Review》的分类系统把软件工程划分为如下的一些子领域：一般问题（概论）、需求/规格说明、工具与技术、编码、程序验

证、测试与排错、编程环境、分布式系统和维护、度量学、管理、其它。从教育的角度看，以上的方法至少存在两个问题：一是不利于课程内容的组织；二是没有足够的灵活性以适应软件工程技术的高速发展和变化。

美国SEI最近所提出的我们称之为“结构化”的方法则较好地解决了以上问题。该方法以软件工程的过程观点和产品观点为主线，力图把软件工程的知识体分解成一些相对独立的、足够小的但包含有某个中心论题的单位模块。该方法如下：

(1) 把软件工程过程划分为开发、控制、管理和操作等四种不同的活动。

- 开发：包括需求分析、规格说明、设计、实现和测试。

- 控制：指各种对软件开发起指导和约束作用的活动。如质量保证、配置管理、验证与确认、软件维护、复审、性能评价等。

- 管理：包括项目计划、资源分配、开发小组的组织、成本估算、法律问题（如同工、营业执照、软件保护等）。

- 操作：包括系统使用人员的培训、系统的移交与安装、新旧系统的交接、系统的操作、系统的退役，以及这些活动对系统开发各种决策的影响。

以上的各种活动又可认为共同由抽象、表现技术、方法、工具、评价、通讯等六个方面组成。

- 抽象：指各种基本原理和形式模型。例如：软件设计的模块化和信息隐藏原理，生命期模型，成本估算模型等等。

- 表现：指各种记号和语言。如项目计划的PERT网记号，设计的DFD记号，实现用的各种程序设计语言。

- 方法：包括各种形式化方法、当前流行的各种实践和方法学。例如，正确性证明的形式验证方法，面向对象的设计方法等等。

- 评价：包括对软件产品、软件过程以及软件对有关部门产生的影响等方面的测定、分析和评价，软件度量与软件标准等。

• 工具, 包括各种单项工具和集成化工具集。

• 通讯, 包括各种口头的和书面的通讯技术、各种文件资料的格式和书写技巧。

(2) 从软件工程的产品观点出发, 按产品的类型和一般质量要求考虑问题。

(a) 不同类型的产品所需要的开发技术、记号等等可能是不同的。因此, 可以以产品类型为主线组织有关的知识内容。软件产品类型可按以下方式划分:

• 按软件系统与应用环境的关系来划分, 如批处理的、实时的、交互式的、嵌入式的, 等等。

• 按处理方式划分, 如集中式的、分布式的、串行的、并行的, 等等。

• 按内部特性来划分, 如表驱动的、进程驱动的、基于知识的, 等等。

• 按应用领域来划分, 如商用系统、导弹控制系统、操作系统、数据库系统、专家系统, 等等。

(b) 不同产品对可靠性、正确性、可维护性、安全性等质量要素的要求可能是不同的, 而为了达到特定的质量指标, 可能必须采取一些特定的行动和手段。因此, 也可以以某些特定的质量要求为主线组织有关的知识内容。

(3) 以活动、活动的侧面、产品类型、质量要求为轴构成一个四维空间。空间中的每一点, 如果非空的话 (即至少包含某一论题的知识体), 就可构成软件工程知识体的一个单位模块。图1直观地给出了明确的说明性例子。

文[5]指出, 整个软件工程的知识体最终可能包含50到100个这样的单位模块。

SEI的方法不但能系统全面地标识、描述和记载软件工程的知识内容, 而且至少还具有如下几个优点:

(1) 把庞大、复杂的软件工程知识体系统地分解成这样的单位模块后, 可以就各个单位模块的主要论题组织有关专家和教育工作者对该论题作出全面、透彻和最好的论述。

(2) 对教育和培训工作来说, 能够根据不同的目的和目标, 把有关的单位模块组织成合适的教材。

(3) 有利于知识的更新。能够及时地修改有关的单位模块的内容, 及时反映软件工程的最新、最好原理、方

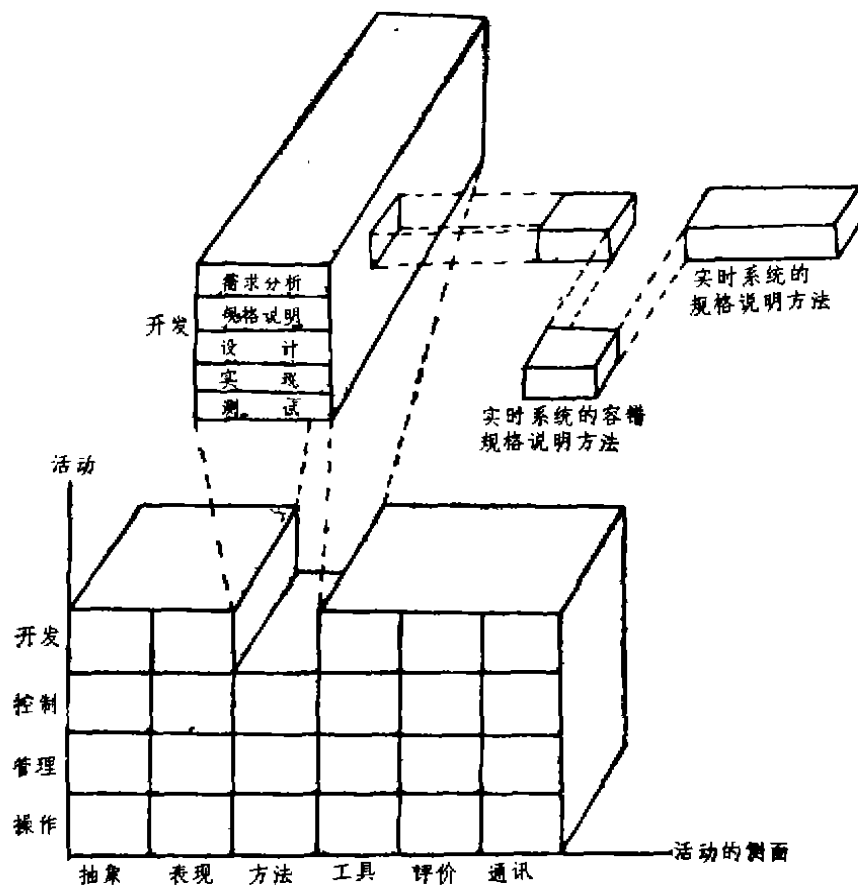


图1 SE知识体的组织结构

法、工具以及各种实践技术和经验。对于尚处于高速发展之中的软件工程学科来说,这一点显然是极为重要的。

(4)使得各类有关人员对软件工程的知
识体有一个全面的了解,这对软件工程的研究与发展将起积极的促进作用。

五、结 论

本文讨论的内容对整个软件工程学科的教学、科研以及实际的软件开发都有十分重要的指导意义和参考价值,但教学内容的确定和课程的组织还与教育形式、具体的培养目标、学制的长短、学生负担等等问题紧密相关,这些有待进一步探讨。

参考文献

- [1] 余金山,“软件工程教育——一个必须引起充分重视的问题”,全国高等学校计算机教育研究会1990年年会论文集,重庆,1990年11月
- [2] 柯中(编译),“软件工程的概念”,软件工
程动态, No.1, 1986
- [3] R.W.Jensen and C.Tonies, Software
Engineering, Prentice-Hall, 1979
- [4] G.Ford and N.Gibbs, “A master of soft-
ware engineering curriculum”, Compu-
ter, Vol.22, No.9, 1989
- [5] N.Gibbs, “The SEI education progra-
m, The challenge of teaching future
software engineering”, CACM, Vol. 32,
No.5, 1989
- [6] C.Floyd, “Outline of a paradigm chan-
ge in software engineering”, ACM
SIGSOFT SEN, Vol.13, No.2, 1988
- [7] D.Bjorner, “On the use of formal me-
thods in Software devepoment”, in P-
roc. 9th Int. Conf. Software Eng., I-
EEE Comput. Soc., 1987
- [8] D.Bjorner, “软件开发的各个方面”, 软件
产业, No.9, No.10, 1989
- [9] J.Musa, “软件工程职业的前景”, 计算机
科学, No.4, 1986
- [10] 朱三元等, 软件工程指南, 上海翻译出版
公司, 1985
- [11] 李树芬, “软件工程的方法学”, 计算
机应用与软件, No.1, 1987

计 算 机 学 科

第4期(双月刊)

1991年8月23日出版

国内统一刊号: CN51—1239

代号: 78—68

定价: 2.15元 国外定价: 5美元

编 辑 者: 中国科学技术情报研究所重庆分所

顾 问: <计算机科学> 审编委员会

出 版 者: 中国科学技术情报研究所重庆分所

重庆市市中区胜利路132号

邮政编码: 630013

印 刷 者: 中国科学技术情报研究所重庆分所印刷厂

总发行处: 四 川 省 重 庆 市 邮 政 局

订 购 处: 全 国 各 地 邮 政 局