

国家示范性软件学院教材

软件项目实训

雷敏 姚志林 编著



国防工业出版社
National Defense Industry Press

软件项目实训

雷敏 姚志林 编著

国防工业出版社

·北京·

内 容 简 介

为缩短高校毕业生和企业岗位实际需求之间的差距,越来越多的高校开始建立校内和校外各种类型的实习实训基地,项目实训成为软件工程人才培养的一个重要途径。

本书不是介绍某一门具体的技术,而是根据在校学生和企业岗位之间的差距,通过一系列的实训环节和讲座内容,让参加项目实训的学员将所学的软件工程知识形成体系,更多地了解企业的工作流程和文化,掌握规范文档的撰写,同时介绍企业软件人才职业素质的培养、软件人才的职业规划和面试技巧。本书的主要内容是软件项目实训导论、项目组建、职业素质、项目概述及环境搭建、项目开发计划、需求分析、概要设计、详细设计、编码与实现、系统测试、项目总结、求职与面试、软件人才职业规划与发展。本书收集了软件项目开发中所需要撰写的各种文档模板。

本书可作为高等院校软件工程和计算机等相关专业的本科生和研究生的课程设计、项目实践、项目实训教材,同时也可作为各培训机构项目实训和实习培训的教材。

图书在版编目(CIP)数据

软件项目实训/雷敏,姚志林编著. —北京:国防工业出版社,2010.3
ISBN 978-7-118-06714-9

I. ①软... II. ①雷... ②姚... III. ①软件开发—项目管理 IV. ①TP311.52

中国版本图书馆 CIP 数据核字(2010)第 018191 号

※

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100048)

天利华印刷装订有限公司印刷

新华书店经售

*

开本 787×1092 1/16 印张 11 字数 243 千字

2010 年 3 月第 1 版第 1 次印刷 印数 1—3000 册 定价 20.00 元

(本书如有印装错误,我社负责调换)

国防书店:(010)68428422
发行传真:(010)68411535

发行邮购:(010)68414474
发行业务:(010)68472764

前 言

随着我国软件产业的不断发展壮大,软件在当今信息社会中占的地位越来越重要,市场对各领域软件人才的需求越来越大,但是,尽管高校毕业生数量一年比一年多,却因为缺乏实践工作经验而不能满足软件企业对实用型软件人才的大量需求。对于企业来讲,由于软件开发周期短,版本更新速度快,企业希望新进员工无需培训即能上岗,迅速展开实质性工作,为企业创造价值,因此要求企业和人才必须是“互动”的成长模式,高校培养人才时首先要考虑的就应该是企业的需要。这就要求学校在培养人才时更加注重实践,特别是模拟企业开发环境的项目实训,培养更多的 U 盘型人才,让学生毕业后能尽快地融入企业。

本书介绍的项目实训根据学生在学校和企业岗位之间的差距,通过一系列的实训环节和讲座内容,让软件学院学生将所学的软件工程知识形成体系,更多地了解企业的工作流程和文化,同时介绍企业软件人才职业素质的培养、软件人才的职业规划和面试技巧。帮助学生毕业后能够更好地适应工作岗位,发挥自己的潜力。学生在参加本项目实训之前,应具备一定的软件开发基础,掌握 C 和 C++ 编程语言和编程工具,熟悉面向对象程序设计、数据库、软件工程、软件测试等相关技术。

本书章节安排及学时数分配建议如下:

序号	章节内容	学时数
1	软件项目实训导论	2
2	项目组组建	2
3	职业素质	2
4	项目概述及环境搭建	2
5	项目开发计划	2
6	需求分析	4
7	概要设计	4
8	详细设计	4
9	编码与实现	2
10	系统测试	2
11	项目总结	2
12	求职与面试	2
13	软件人才职业规划与发展	2

本书第4章、第10章由姚志林编写,其他章节由雷敏编写。董培媛为本书整理了大量图表和模板,同时,本书编写过程中参考了东软实训先进的体系,在此一并表示感谢。

由于作者水平有限,不足之处恳请广大读者批评指正。作者电子邮件地址:leimin@bupt.edu.cn.

作者
2009年10月

目 录

第1章 软件项目实训导论	1
1.1 软件人才培养目标.....	1
1.2 软件人才如何适应企业需求.....	3
1.3 实训培养模式.....	4
1.4 软件工程基本原理.....	5
1.5 实训项目软件生命周期.....	7
1.6 实训体系和教学安排	10
第2章 项目组组建	14
2.1 人员组成	14
2.2 如何分组	16
2.3 人员分工	17
2.4 项目组长推选	18
2.5 项目技术负责人推选	19
2.5.1 笔试题目1	20
2.5.2 笔试题目2	22
2.6 项目组会议及日报表	24
2.7 项目组规章制度	25
第3章 职业素质	28
3.1 时间管理	28
3.1.1 时间特性.....	28
3.1.2 时间管理的四个象限.....	30
3.2 团队合作	31
3.2.1 如何快速融入团队.....	31
3.2.2 团队合作三大资格.....	32
3.3 团队沟通技巧	33
3.3.1 团队沟通的优缺点.....	34
3.3.2 影响团队沟通的因素.....	35
3.3.3 团队的行为规范.....	35

3.3.4	团队沟通的障碍	36
3.3.5	团队沟通原则	36
3.4	职场礼仪	37
3.4.1	办公室礼仪	37
3.4.2	握手礼仪	38
3.4.3	道歉礼仪	38
3.4.4	电梯礼仪	39
3.4.5	垃圾桶礼仪	39
3.4.6	电话礼仪	39
3.4.7	介绍礼仪	39
3.4.8	递名片礼仪	39
3.4.9	见面礼仪	40
3.4.10	迎送礼仪	40
3.5	软件工程师职业道德规范	40
第4章	项目概述及环境搭建	41
4.1	需求引出	41
4.2	项目需求说明	42
4.2.1	项目概要	44
4.2.2	软件详细需求	46
4.3	项目开发环境	49
4.3.1	开发环境的搭建	49
4.3.2	文档编写工具	49
4.3.3	群组通信软件	49
4.3.4	编程开发环境	49
4.3.5	配置管理工具	50
4.4	配置管理	50
4.4.1	配置管理含义	50
4.4.2	配置管理的重要性	50
4.4.3	配置管理的内容	52
4.4.4	软件配置管理过程	54
4.4.5	配置管理员的职责	55
4.4.6	配置管理工具 VSS	55
4.4.7	VSS 的使用	56
第5章	项目开发计划	59
5.1	制定开发进度计划	59

5.2	甘特图与时间管理	60
5.3	项目开发计划书	62
第6章	需求分析	65
6.1	需求重要性	66
6.2	需求来源	66
6.3	需求规格说明书	67
6.3.1	需求规格说明的结构和内容	67
6.3.2	需求规格说明的特性	71
6.3.3	需求规格说明文档的编写要求	72
6.4	需求评审	73
6.4.1	需求审查的关键点	74
6.4.2	审查形式与过程	75
6.4.3	评审员的职责	76
6.5	需求变更控制与管理	76
第7章	概要设计	78
7.1	概要设计的任务	78
7.2	模块化	79
7.2.1	模块	79
7.2.2	模块化与逐步求精	79
7.2.3	模块的分割方法与原则	80
7.3	概要设计说明书	81
7.4	概要设计文档评审	85
第8章	详细设计	86
8.1	详细设计目标和任务要求	86
8.1.1	详细设计与概要设计的区别	87
8.1.2	详细设计的任务	87
8.1.3	详细设计的需求	88
8.2	详细设计说明书	89
8.3	详细设计文档评审	90
第9章	编码与实现	91
9.1	编码规范	91
9.1.1	关于文件结构	92
9.1.2	编辑风格	92

9.1.3	注释	92
9.1.4	符号名的命名	93
9.1.5	函数定义	93
9.2	用户界面设计	94
9.2.1	用户界面设计的原则	94
9.2.2	用户界面设计工作流程	96
9.2.3	用户交互方式	96
9.3	用户操作手册编写	97
9.3.1	何时撰写用户操作手册	97
9.3.2	用户操作手册模板	98
9.4	我需要编码吗?	100
第 10 章	系统测试	102
10.1	测试目的	103
10.2	测试分类	103
10.3	单元测试	105
10.4	集成测试	106
10.5	测试计划	108
10.6	实训项目的测试	111
10.6.1	测试用例编写规范	111
10.6.2	测试中错误级别的定义	112
10.6.3	软件测试暂停及停止标准	113
10.6.4	软件测试中需注意的问题	114
10.6.5	软件测试报告	115
第 11 章	项目总结	117
11.1	实训答辩申请	117
11.2	个人总结和项目组总结	119
11.3	项目总结评审	120
11.4	项目组和个人成绩评定	121
11.5	实训反馈	123
第 12 章	求职与面试	124
12.1	简历准备	124
12.1.1	简历撰写前的准备	125
12.1.2	如何撰写简历	125
12.1.3	简历的基本格式	126

12.1.4	简历制作应当避免的问题	127
12.2	常见面试问题	127
12.2.1	面试测评主要内容	128
12.2.2	面试中经典问题	129
12.3	面试注意事项	131
12.3.1	着装礼仪	131
12.3.2	文明礼貌	131
12.3.3	言谈举止	132
12.4	如何快速融入职场	132
第 13 章	软件人才职业规划与发展	135
13.1	软件人才职业规划	135
13.2	软件人才的类型	137
13.3	大学生职业规划的方法——自我规划法	139
13.4	我适合从事哪种类型的工作	141
13.5	职业生涯规划的误区	143
13.6	软件人才职业发展	145
附录	VSS 安装与配置	148
参考文献		166

第1章 软件项目实训导论

随着我国软件产业的不断发展壮大,软件在当今信息社会中占的地位越来越重要,市场对各领域软件人才的需求越来越大,因此各高校都在相关专业培养大量的软件开发人才。但软件产业发展迅速,要求企业和人才必须是“互动”的成长模式,而目前各高校软件人才的培养没有考虑企业的实际需求。我国的软件人才市场现状是毕业生富余而企业却招不到合适的实用型软件人才,市场上出现了非常奇怪的现象,一方面很多高校毕业生找不到合适的工作,另一方面很多软件企业找不到满意的员工。目前高校毕业生找工作难的主要原因是实践能力弱,不能满足企业的要求。解决此矛盾的方法是学校在培养人才时应更加注重软件项目实训,特别是模拟企业开发环境的软件项目实训,培养更多的U盘型人才,让学生在毕业时能掌握软件开发的各种技能,满足企业的需要,找到称心如意的工作,并能快速地融入企业。

1.1 软件人才培养目标

1. 基础知识和技能

计划从事软件开发的毕业生应该熟练掌握软件工程知识与技能,并且具备作为软件工程师从事软件工程实践活动所需的各种专业能力。在教学计划中,每个学期应该有不同层次软件工程技能的教学和实践内容,经过正规的、循序渐进的强化学习和实践,学生的知识和技能将不断地积累和提高。同时,毕业生必须遵循职业的、经济的规范与社会伦理公德,要有强烈的职业责任感。

2. 团队工作的能力

任何一个软件项目都不可能由某一个个体单独来完成,它通常由一定规模的项目小组来完成,软件工程学科的毕业生不仅需要完成个人的工作任务,而且还需要与项目组其他成员协同工作,因此,软件人才必须具备个人工作与团队协作的能力,以便能够开发出高质量的软件产品。在各种实践和实训教学环节中,有意识地安排学生参加实际的软件开发项目小组的工作,让学生清楚地了解开发小组的性质、活动或角色,在每个项目小组中尽量能担任不同的角色。同时强调专门化训练的重要性,包括提交各项文档和成果的最后期限,同时注重沟通、个人与团队的绩效评估等方面的训练。

3. 分析与解决问题的能力

从事软件项目开发的毕业生应当能够协调各种相互冲突的项目目标,在成本、时间、知识、现有系统与组织环境等条件约束下找到适当的折衷方案。教学计划应当提供一些相关练习,要求学生面对相互矛盾和不断变化的需求时,能够完成高质量的需求分析和灵活的软件设计。这些练习应该具有真实环境的成分,以便学生获得真正的经验。

在学习过程中,学生必须了解各种通用的工程设计方法以及解决软件应用领域中特定问题的方法,能够理解各种可选方案的优缺点以及如何在特定的情况下进行适当的选择,并能利用软件工程的方法,考虑伦理、社会、法律、安全和经济等因素,为一个或多个应用领域设计适当的解决方案。

在软件人才培养计划的末期,工程实践和项目实训是十分重要的环节。学生以项目组形式完成一个大项目,展示自己的软件工程知识面以及应用所学技术解决真实问题的能力,从而达到综合应用各种课程所学内容的目的。

4. 谈判与沟通的能力

在一个比较典型的软件开发环境中,软件工程专业人员应当具备谈判能力、高效工作习惯、管理能力以及与项目组同事之间良好沟通的能力。因此,计划从事软件开发的学生在校期间有必要参与至少一次实际为客户提供解决方案的活动,从整个活动中认识到软件工程师必须提供真正有用的软件产品。

在条件允许的情况下,应该安排一段时间的企业实习的教学计划,也可以邀请有经验的软件工程师进行讲座,甚至可以参加外界的软件竞赛。所有这一切将使学生积累更加丰富的经验,并有助于建立一个培养高质量软件人才的环境。

5. 适应发展的能力

软件技术在不断发展和变化,各种新技术层出不穷。因此,在整个学习过程结束时,学生应具备自我终身学习的能力。自学能力的培养应贯穿于整个学习计划的不同阶段,学生到达高年级时(如项目实训环节),能够自觉学习随时涌现的新概念、新模型、新技术和新方法,以适应学科未来的发展和变化。

6. 认知能力和技能

知识与理解:掌握软件工程学科的各种基础知识,包括基本概念、原理和理论,具备在实践应用中综合运用这些知识的能力,特别是能够认知和理解抽象事物的能力。

需求:理解软件需求的概念和重要性,熟悉需求开发和需求管理的基本过程和主要活动,明确定义和分析各种软件系统的需求,形成需求规格说明书并进行验证。

方法与工具:将适当的理论、实践知识和工具应用到软件系统的开发过程中,包括需求分析、设计、实现、评审和测试以及演化。

7. 实践能力和技能

设计与实现:基于需求规格说明书进行软件系统的概要设计、详细设计和编码实现。

评审与测试:在软件开发过程中,实施有效的绩效评审和检查,并对软件产品的质量进行测试和评价。

过程与管理:使用标准进行质量管理、过程管理和项目管理,度量和评价软件过程和项目进展,实施风险分析和变更管理。

人机交互:将人机交互理论应用到评价和构建包括用户界面、网页界面和多媒体系统在内的各种实践活动中,设计满足用户需求的用户界面。

工具:有效地使用软件工程工具,设计和开发各种软件系统,建立良好的系统文档。

8. 其他技能

判断能力:运用所学的知识、技能和方法,对系统的各种解决方案进行合理的判断和选择,能够根据各工程的实际情况选择合理的方案,同时具备批判性思考的能力。

创新能力:充分认识到软件工程知识在当前的局限性和未来的不断发展可变性,积极探索新概念、新技术和新方法。

沟通能力:运用口头、书面和图形等形式进行有效的沟通,清楚地表达技术问题及其解决方法,具备倾听、说服和协商的技能。

团队协作能力:充分理解团队合作的重要性,在实际项目的开发过程中,能够与其他成员一起协同工作,认真听取和判断其他成员的观点,并提出自己的建议,在团队中扮演积极而不是消极的角色。

专业发展:坚持不断地学习,使自己在该专业的能力与学科的发展保持同步。

1.2 软件人才如何适应企业需求

不少学习软件开发的学生,在低年级时,重点学习编程语言和面向对象的设计方法,很多学生常常会形成这样的认识:所谓学习软件,就是重点掌握好一两门计算机语言(如C++、Java),然后编写几个小程序,完成几个项目实践就可以了。甚至有一些计划从事软件开发的学生,本身并不是计算机、软件工程等相关专业的,编程知识都是自学的。他们认为只要编程能力强,成为所谓的“编程高手”,在软件公司里就是所谓的“关键人物”,薪资待遇、职业前途都不在话下。其实,这种观点是不正确,至少不全面。随着软件产品规模不断扩大,软件开发涉及的人员和范围越来越广,靠一个人单打独斗很难完成一个大型项目的开发。

传统教育模式培养的毕业生往往不能符合当今软件企业对软件人才的要求,因此软件工程教育应该根据它自身的生命周期过程和软件工程管理需求,在充分考虑学生的接受程度和理解能力的基础上,更多地考虑分布在各课程中的知识点,应如何相互综合、沟通、衔接、融合和组织。

同时,在教育过程中,更应该加强软件工程的实践环节——项目实践和实训的组织。从事软件开发是一门实践性非常强的学习过程,在每一个教学过程中都安排了相应的实践环节,比如C语言程序设计、数据结构的作业,学生可以根据老师专门编排的小例程,完成一段小程序,学习编码过程。但是,如果完成各项目的过程仍然采用按单一课程考察(需求做需求、设计做设计、测试做测试),而不考虑整个软件过程上下、横向各环节的综合因素,忽略了这些真实要素的“模拟”,这样的项目实践培养出来的学生缺乏整体的概念。这就像一个企业,销售、市场、产品、开发和项目实施部门都是单兵作战,他们之间既没有沟通,也没有接口,更没有统一指挥和协同,遇到项目招标时,每个部门都去一个人,只负责自己部门的业务,那么,这个企业的命运就可想而知了。对于学生来说,要想成为合格的软件人才,就必须掌握软件开发整个生命周期各阶段的技能,否则也就是单兵作战,不能发挥更大的作用。因此在软件人才培养环节的实践教学,需要通过实训来完成多个知识领域的综合,完成一个大型、完整的项目与项目团队开发过程的现代软件工程教学。通过软件工程实训,促进学生有针对性、主动地学习和查阅有关软件工程的基本教学内容和相关资料,实现如下目标:

(1) 深化所学知识,完成从理论到实践的转化。学生通过项目实训的实践过程,进一

步加深对软件工程方法和技术的了解,将软件工程的理论、软件开发所需的技术和方法运用到整个项目实训中,并在实训中掌握这些软件工具的使用方法。

(2) 提高分析和解决实际问题的能力。软件项目实训是一次对软件研发企业整体研发过程的模拟训练,学生通过实训项目过程的学习,积累经验、提高分析问题和解决问题的能力。

(3) 培养撰写良好文档的习惯。撰写良好的文档在正规的研发过程中非常重要。作为程序员,大约有 30% 的工作时间是在撰写各种技术文档,而作为高级程序员或系统分析人员,这个比例会高达 70% 以上。正规 IT 企业,对软件工程各阶段文档都有着严格的要求。通过项目实训,让学生了解文档的重要性,并学会如何撰写规范化的文档。

开发高质量的软件需要高素质的专业人才,学校应当认识到专业实践和项目实训对于培养软件工程人才职业化素质的重要性,并且清楚地了解缺乏这方面训练所引起的不良后果。软件工程专业实践和实训包括诸如问题解决、管理、沟通、书面与口头交流、团队协作以及跟踪快速变化的学科发展等多方面的问题及其相关活动。

软件工程专业培养的人才应该具有综合素质,因此计划从事软件开发的学生在校期间不仅仅是要学习如何编程,还应该了解和掌握整个软件开发各生命周期所需的技能,如:需求分析、概要设计、详细设计、编码实现、软件测试等多道工序,以及如何生产出合格的软件产品。在市场上最受欢迎的软件人才是懂得软件开发各环节要求的系统设计师、架构师、需求分析师、技术经理和软件项目经理等。

因此高校的教师和学生都应该了解到,软件工程专业学习不仅是学习如何用语言实现一个具体的算法、如何设计好一个数据库结构、如何仔细推敲网络协议或接口的细节,更应该从整体的角度来把握软件产品开发和生产的全过程,把软件设计和开发看成是一个生产过程的完整工程性问题。

1.3 实训培养模式

软件项目实训有别于传统的课堂教学,实训以企业的真实项目作为案例,以“重现项目流程”为宗旨,让学生亲身体验软件开发全部流程,并参与软件开发所需的各项工作,了解软件企业如何组织一个项目开发团队、项目组成员之间的角色如何分配以及各角色之间如何配合。

参加实训的学生在指导教师的带领下,探索、发现、更加接近企业岗位需求,通过项目实训来完成某一个大型软件的开发。

项目实训的特点是采用完全企业化的教学监督管理机制,视参训学员为企业员工,通过角色和环境的转换让学员如同置身于真实的企业环境、投身于团队之中,从而积累规范化、工程化的软件开发实战经验,通过职业素质训练课程掌握时间管理、压力管理、团队协作、沟通能力等各种基本的职业素质。同时在整个实训过程中为参训学员提供软件人才职业生涯的规划和各种面试的技巧等。实训项目根据学生在学校和企业岗位之间的差距,定制一系列的培训环节及相应的讲座内容,学校可以根据学生的需要选择相应的内容弥补学生在知识点和能力方面的欠缺,帮助学生提升就业竞争力,并在今后的职场中寻求更大的发展空间。

软件项目实训和传统的课堂教学在方法上存在较大的差异,项目实训和传统课堂教学的比较如图 1-1 所示。

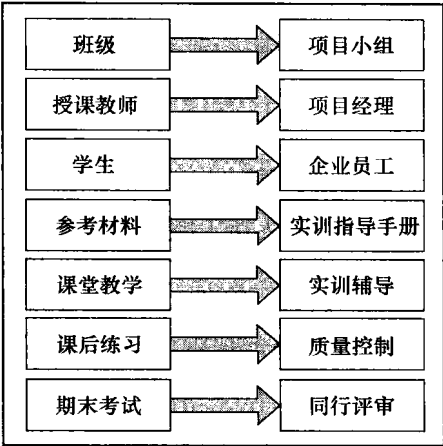


图 1-1 软件项目实训和传统课堂教学比较示意图

在传统课堂教学过程中,学生都是以班级为单位进行组织,在实训过程中,所有的参训学员被分成多个项目小组,这些项目小组可以跨班级,甚至跨学校。实训过程中,指导教师担任项目经理的角色,参训学员被当作企业员工,需要接受整个实训过程中各种规章制度的约束和管理。除课堂教学外,指导教师主要是在机房和实验室对学生手把手进行实训指导,解决学生在实训过程中遇到的各种问题。为确保整个实训过程的质量,采用质量控制的方法来控制整个实训过程的各教学环节,同时当每一个环节结束后需要进行同行评审,同行评审通过后才能进入下一环节。

1.4 软件工程基本原理

自 1968 年在联邦德国召开的国际会议上正式提出并使用了“软件工程”这个术语以来,研究软件工程的专家学者们陆续提出了 100 多条关于软件工程的准则或“信条”。著名的软件工程专家 B. W. Boehm 综合这些学者们的意见并总结了 TRW 公司多年开发软件的经验,于 1983 年在一篇论文中提出了软件工程的 7 条基本原理。他认为这 7 条原理是确保软件产品质量和开发效率原理的最小集合。这 7 条原理是互相独立的,其中任意 6 条原理的组合都不能代替另一条原理,因此,它们是缺一不可的最小集合。并且这 7 条原理又是相当完备的,人们虽然不能用数学方法严格证明它们是一个完备的集合,但是,可以证明在此之前已经提出的 100 多条软件工程原理都可以由 7 条原理的任意组合蕴含或派生。下面简要介绍软件工程的 7 条基本原理。

1. 用分阶段的生命周期计划严格管理

经统计发现,在不成功的软件项目中有一半左右是由于计划不周造成的,可见把建立完善的计划作为第 1 条基本原理是吸取了前人的教训而提出来的。

在软件开发与维护的漫长生命周期中,需要完成许多性质各异的工作。这条基本原理意味着,应该把软件生命周期划分成若干个阶段,并相应地制定出切实可行的计划,然

后严格按照计划对软件的开发与维护工作进行管理。Boehm 认为,在软件的整个生命期中应该制定并严格执行 6 类计划,它们是项目概要计划、里程碑计划、项目控制计划、项目控制计划、产品控制计划,验证计划和运行维护计划。不同层次的管理人员都必须严格按照计划各尽其职地管理软件开发与维护工作,绝不能受客户或上级人员的影响而擅自背离预定计划。

2. 坚持进行阶段评审

当上一个阶段评审不能通过时,就不能进入下一个阶段开发。而且在软件开发过程中,必须意识到这一点,软件的质量保证工作不能等编码阶段结束后再进行。这样说至少有两个理由:第一,大部分错误是在编码之前造成的,例如,根据 Boehm 等人的统计,设计错误占软件错误的 63%,编码错误仅占 37%;第二,错误发现与改正得越晚,所需付出的代价越高。因此,在每个阶段都进行严格的评审,评审不通过就不能进入到下一个阶段,这样就可以尽早发现在软件开发过程中所犯的错误,是一条必须遵循的重要原则。

3. 实行严格的配置控制

在软件开发过程中不应该随意改变需求,因为改变一项需求往往需要付出较高的代价。但是,没有一个产品在开发过程中是不改变需求的,由于外部环境的变化,也有可能是在概要设计阶段慢慢地了解到需求的不合理性,相应地改变用户需求是一种客观需要,显然不能硬性禁止客户提出改变需求的要求,而只能依靠科学的产品控制技术来顺应这种要求。也就是说,当改变需求时,为了保持软件各个配置成分的一致性,必须实行严格的配置控制,其中主要是实行基准配置管理。所谓基准配置又称为基线配置,它们是经过阶段评审后的软件配置成分(各个阶段产生的文档或程序代码)。基准配置管理也称为变动控制,即一切有关修改软件的建议,特别是涉及对基准配置的修改建议,都必须按照严格的规程进行评审,获得批准以后才能实施修改。绝对不能谁想修改软件(包括尚在开发过程中的软件)就随意进行修改。在软件开发过程中,配置管理越来越重要,在项目组中,必须指定一个专门的配置管理人员来负责整个项目的配置管理工作,以保证项目文档的一致性。

4. 采用现代程序设计技术

从提出软件工程的观念开始,人们一直把主要精力用于研究各种新的程序设计技术。20 世纪 60 年代末提出的结构程序设计技术,已经成为大多数人公认的先进程序设计技术。以后又进一步发展出各种结构分析(SA)与结构设计(SD)技术。近年来,面向对象技术已经在许多领域中迅速地取代了传统的结构化开发方法。实践表明,采用先进的技术不仅可以提高软件开发和维护的效率,而且可以提高软件产品的质量。在软件工程领域,各种技术和开发手段层出不穷,软件工程的系统设计和开发人员应当尽量采用现代和先进的技术来设计和开发整个系统。

5. 结果应能清楚地审查

软件产品不同于一般的物理产品,它是看不见摸不着的逻辑产品。软件开发人员(或开发小组)的工作进展情况可见性差,软件产品的代码质量和运行质量都无法准确度量,从而使得软件产品的开发过程比一般产品的开发过程更难于评价和管理。为提高软件开发过程的可见性,更好地进行管理,应该根据软件开发项目的总目标及完成期限规定开发组织的责任和产品质量标准,从而使得所得到的结果能够清楚地审查。因此,每个阶段对

文档都必须严格审查。

6. 开发小组的人员应该少而精

这条基本原理的含义是,软件开发小组的组成人员的素质应该较高,而人数则不宜过多。开发小组人员的素质和数量是影响软件产品质量和开发效率的重要因素。素质高的人员开发效率比素质低人员的开发效率可能高几倍至几十倍,而且素质高的人员所开发的软件中的错误明显少于素质低的人员所开发的软件中的错误。此外,随着开发小组人员数目的增加,管理和沟通的成本也会越来越高,因为交流情况讨论问题而造成的通信开销也急剧增加。当开发小组人数为 N 时,可能的通信路径有 $N(N-1)/2$ 条,可见随着人数 N 的增大,通信开销将急剧增加。因此,组成少而精的开发小组是软件工程的一条基本原理。

7. 承认不断改进软件工程实践的必要性

遵循上述 6 条基本原理,就能够按照当代软件工程基本原理实现软件的工程化生产,但是,仅有上述 6 条原理并不能保证软件开发与维护的过程能赶上时代前进的步伐。因此,Boehm 提出应把承认不断改进软件工程实践的必要性作为软件工程的第 7 条基本原理。按照这条原理,不仅要积极主动地采纳新的软件技术,而且要注意不断总结经验,例如,收集进度和资源耗费数据,收集出错类型和问题报告数据等。这些数据都可以用来评价新的软件技术的效果,而且可以用来指明必须着重开发的软件工具和应该优先研究的技术。因此,现代软件企业都采用 CMMI(能力成熟度模型集成)来改进软件企业的开发过程。CMMI 为软件企业的过程能力提供了一个阶梯式的改进框架,基于过去所有软件工程过程的成果,吸取了以往软件工程的经验教训,提供了一个基于过程改进的框架。它指明了一个软件组织在软件开发方面需要管理哪些主要工作、这些工作之间的关系、以及以怎样的先后次序,一步一步地做好这些工作使得软件组织走向成熟。

1.5 实训项目软件生命周期

项目的实训过程基本上是按照软件工程生命周期来规划整个的教学内容和教学安排,除技术内容之外,在整个的实训教学内容和教学安排上融入了职业素养、职业规划、面试技巧等方面内容。

不同软件的生命周期与开发模型有关,不同的开发模型,可能对应着不同的生命周期。生命周期不同,该软件的开发阶段划分、评审次数、基线标准都有所不同。软件公司的项目组在开发一个大型项目或产品时,首先在技术上必须选择一个开发模型,使该开发模型非常适合这个项目或产品的生命周期。

软件开发模型是软件开发的全部过程、活动和任务的结构框架。软件开发模型能清晰、直观地表达软件开发全过程,明确规定要完成的活动和任务,用来作为软件项目开发的基础。软件工程模型有:瀑布模型、原型模型、演化模型、螺旋模型和喷泉模型等。

由于项目实训的特殊性及学生对于软件工程的了解程度,实训时建议采用生命周期为“瀑布模型”。瀑布模型依据软件生命周期方法学开发软件,各阶段的工作自顶向下从抽象到具体的顺序进行。利用瀑布模型开发软件系统时,每一个阶段完成的任务得以确定后才能进入下一个阶段,前一阶段的成果作为下一阶段的输入。否则就要返回到前