

RUANJIAN GONGCHENG
YU BIYE SHEJI GUIFAN



软件工程 与毕业设计规范

■ 张少刚 马海云 编著



国防工业出版社
National Defense Industry Press

软件工程与毕业设计规范

张少刚 马海云 编著



国防工业出版社

· 北京 ·

图书在版编目(CIP)数据

软件工程与毕业设计规范/张少刚,马海云编著. —北京:
国防工业出版社,2015.5
ISBN 978-7-118-10152-2

I. ①软... II. ①张... ②马... III. ①软件工程-毕
业设计-高等学校-教学参考资料 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2015)第 100599 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)

北京奥鑫印刷厂印刷

新华书店经售

*

开本 787 × 1092 1/16 印张 14½ 字数 329 千字

2015 年 5 月第 1 版第 1 次印刷 印数 1—2000 册 定价 39.50 元

(本书如有印装错误,我社负责调换)

国防书店:(010)88540777

发行邮购:(010)88540776

发行传真:(010)88540755

发行业务:(010)88540717

前 言

毕业设计是高等院校应届毕业生在毕业前接受课题任务,在教师的指导下独立进行科学研究或工程实践并取得成果的过程,是高等院校实现人才培养目标的综合性实践教学环节,是一次理论与实践相结合的极佳的锻炼机会。毕业设计质量的好坏是衡量高等院校教育、教学质量好坏的重要评价指标,反映了高等院校的办学水平。

软件工程是一门指导计算机软件开发和维护的工程学科,其出现的背景和根本目的是利用工程学和经济学的思想来指导软件开发的过程,从而解决软件开发领域日趋严重的软件危机问题。在软件工程中,把软件产品从形成概念开始,经过开发、使用和维护,直到最后退役的全过程称为软件生命周期。

高等院校毕业设计过程中的各个环节与软件生命周期的各个阶段有着一定的映射关系。我们可以把毕业设计环节作为一个大型的软件项目,以软件工程思想为指导思想,在毕业设计的过程中严格遵守软件工程学的各个原则,这样可以很好地解决目前毕业设计中存在的问题,提高毕业设计的质量。

本书共分7章,具体内容如下。

第1章为引言,分析介绍了毕业设计与软件工程的关系。

第2章为软件工程模型,介绍了软件开发过程中常用的几种模型。

第3章为软件工程过程,把毕业设计过程与软件开发周期的各个阶段联系起来,详细介绍了各个阶段的设计技巧。

第4章为几种常用开发工具介绍。毕业设计时常用开发工具包括 SQL Server 2008、C++、Struts 等,这一章对运用这些工具进行设计时的要点进行了详细的介绍。

第5章为书店网上售书系统的开发。

第6章为移动渠道酬金系统的研究与实现,通过这两个案例的设计,详细说明了如何运用软件工程思想进行毕业设计,以及每个阶段的设计方法。

第7章为设计中常用的数据库技术,介绍了毕业设计过程中常用的数据库技术。

本书的特色如下。

(1) 准确的案例。在第5章的书店网上售书系统的开发和第6章的移动渠道酬金系统的研究与实现中,每个案例都是精心选择的,第6章的案例是作者亲自设计并经过测试的,这样使得知识点更能得到体现,以达到融会贯通的目的。

(2) 与毕业设计紧密结合。书中的每个设计步骤都符合软件工程的设计规范,通过理论与实践相结合的原则,把毕业设计的每个阶段与软件工程的每个阶段对应起来,可以

帮助学生更好地完成毕业设计答辩。

在本书的编著过程中,除了总结自己在这方面的一些研究成果外,还参考了许多作者的书籍、论文、著作以及其他在互联网上公布的相关资料等,从中受到了不少帮助和启发,由于篇幅所限,恕无法一一列出,在此对他们表示衷心的感谢。

由于作者学识和水平有限,加之时间仓促,书中难免存在不足之处,恳请广大读者与同行专家、学者批评指正,让更多读者受益。

最后,感谢天水师范学院教务处、科研处、电信学院各位领导老师的关心支持!感谢每一位关心和支持本书编写工作的各方人士!国防工业出版社的责任编辑同志为本书的出版做出了大量工作,在此表示衷心的感谢!

编著者

2015年2月

目 录

第1章 引言	1
1.1 软件工程与毕业设计的对应关系	1
1.2 毕业设计中存在的问题	2
1.3 用软件工程思想指导毕业设计	3
1.3.1 以可行性研究指导选题,严把选题关	3
1.3.2 需求分析是毕业设计的关键	3
1.3.3 规范设计过程	4
1.3.4 重视文档的写作	4
1.3.5 重视测试工作,撰写毕业设计说明书	4
第2章 软件工程模型	6
2.1 软件及软件工程	6
2.1.1 软件概述	6
2.1.2 软件工程的定义及其研究内容	6
2.1.3 软件工程的作用	7
2.2 软件工程的基本原理	8
2.3 软件生命周期	10
2.4 软件开发过程模型	12
2.4.1 瀑布模型	12
2.4.2 原型模型	14
2.4.3 快速开发模型	15
2.4.4 螺旋模型	16
2.4.5 增量模型	18
第3章 软件工程过程	19
3.1 软件开发方法	19
3.1.1 问题分析	21
3.1.2 可行性分析	21
3.1.3 需求分析	23
3.1.4 系统设计	29

3.1.5	编写程序代码	31
3.1.6	软件测试	33
3.1.7	软件运行、维护与文档管理	36
3.2	面向对象方法与 UML	37
3.2.1	面向对象的基本概念	38
3.2.2	UML 概述	39
3.2.3	常用的 UML 图	41
3.2.4	面向对象的分析	47
3.2.5	面向对象的设计	50
3.2.6	数据库设计	53
3.2.7	面向对象的编程与测试	54
3.2.8	软件运行、维护与文档管理	55
3.3	软件工程实践中的项目管理	55
3.3.1	项目管理概述	55
3.3.2	人员的组织与管理	55
3.3.3	项目计划	56
3.3.4	风险管理	57
第 4 章	几种常用开发工具介绍	61
4.1	SQL Server 2008 特点	61
4.2	C++ 面向对象程序设计要点	69
4.3	MVC 模式	79
4.3.1	MVC 模式概述	79
4.3.2	J2EE	101
4.4	Struts 介绍	111
4.4.1	Struts 概述	111
4.4.2	Struts 组件	112
4.4.3	Struts 配置文件	115
4.4.4	Struts2 概述	120
4.4.5	Struts2 接口和组件	121
4.4.6	Struts2 配置文件	125
4.4.7	Struts2 标签	127
第 5 章	书店网上售书系统的开发	129
5.1	需求分析	129
5.1.1	所要完成的功能	129
5.1.2	定义	133

5.2	系统的总体分析与设计	137
5.2.1	实现的功能	138
5.2.2	系统的功能	138
5.2.3	系统框架	140
5.2.4	系统流程分析	141
5.2.5	网上书店的整体网页设计	145
5.2.6	数据库的设计	147
5.3	系统详细设计	148
5.4	系统测试	153
第6章	移动渠道酬金系统的研究与实现	155
6.1	概述	155
6.1.1	开发背景及目标	155
6.1.2	问题定义	156
6.1.3	可行性研究	156
6.1.4	系统流程图	157
6.1.5	数据字典	157
6.1.6	论文结构与主要工作	158
6.2	需求分析	158
6.2.1	现有系统的分析	158
6.2.2	对系统的综合要求	158
6.2.3	实体联系图	159
6.2.4	功能概述	160
6.2.5	软件的设计要求	161
6.2.6	设计相关技术介绍	162
6.3	系统设计	168
6.3.1	系统设计的任务与原则	168
6.3.2	系统的总体设计	169
6.3.3	系统设计概要	171
6.3.4	整体结构详细设计	172
6.4	系统实现	173
6.4.1	启动界面设计	173
6.4.2	登录窗口设计	173
6.4.3	主框架设计	174
6.4.4	系统实现的部分源代码	177
6.5	系统测试	184
6.5.1	软件测试概念	184

6.5.2	测试设计	186
6.5.3	测试中的问题	187
6.5.4	测试报告	188
第7章	设计中常用的数据库技术	189
7.1	数据库技术概述	189
7.1.1	数据库系统的基本概念	189
7.1.2	数据库系统的特点	190
7.1.3	数据模型	190
7.1.4	数据管理技术的产生和发展	192
7.2	关系数据库	195
7.2.1	关系数据库概述	195
7.2.2	流行的关系数据库	196
7.3	数据库的创建	197
7.3.1	Access 2003 应用概述	197
7.3.2	新建数据库	198
7.3.3	打开和保存数据库	200
7.4	表的创建与修改	200
7.4.1	创建表	200
7.4.2	修改表	203
7.5	表间关系	204
7.5.1	表之间的关系类型	205
7.5.2	创建表之间的关系	205
7.6	SQL 介绍	216
7.6.1	SQL 概述	216
7.6.2	SQL 的分类	216
7.6.3	SQL 语句	217
参考文献		221

第 1 章 引 言

软件工程是计算机学科中一个年轻并且充满活力的研究领域，其出现的背景和根本目的是利用工程学和经济学的思想来指导软件开发的过程，从而解决软件开发领域日趋严重的软件危机问题。它包括两个方面内容：软件开发技术和软件项目管理。其中，软件开发技术包括软件工程方法学、软件工具和软件工程环境，软件项目管理包括软件度量、项目估算、进度控制、人员组织、配置管理和项目计划。

毕业设计是大学生在校期间最后一个重要的综合性实践教学环节，这个教学环节着重培养学生综合运用所学知识与解决实际问题的能力，体现出了专业的培养目标和要求。对于计算机专业的学生来说，毕业设计是培养学生动手能力和进行工程技术人员必须具备的基本素质训练的重要手段。通过毕业设计，可以培养学生理论联系实际、实事求是的科学态度和严谨求实的工作作风，训练和提高学生方案设计、资料利用、实验研究、理论计算、数据处理、文字表达等方面的能力和技巧。把握好这个环节，对学生的就业有很大的帮助。

1.1 软件工程与毕业设计的对应关系

毕业设计本身就是个工程，因此要用工程的方法去处理。软件工程是每个学生都学习过的内容，便于学生应用。在软件工程中，把软件产品从形成概念开始，经过开发、使用和维护，直到最后退役的全过程称为软件生命周期。软件生命周期与毕业设计过程各个阶段的对应关系如图 1-1 所示。

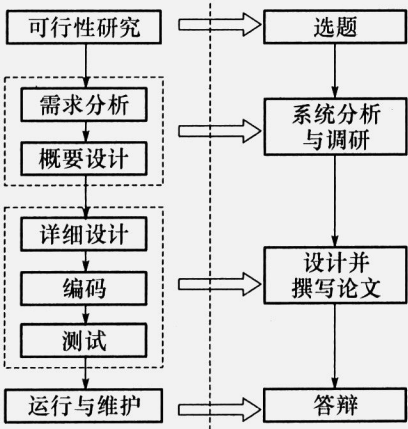


图 1-1 软件生命周期与毕业设计各阶段的对应关系

由图 1-1 的对应关系，可以把毕业设计环节作为一个大型的软件项目，以软件工程思想为指导思想，在毕业设计的过程中，只要严格遵守软件工程学的各个原则，就可以

很好地解决目前毕业设计中存在的问题，提高毕业设计的质量。

目前，许多高等院校计算机专业的学生在毕业设计环节都存在着各种各样的问题，比较突出的问题就是没有严格按照软件工程的方法来进行毕业设计。

1.2 毕业设计中存在的问题

1. 选题不当

选题恰当是做好毕业设计的前提，但选题不当的情况仍然时有发生，主要表现在两个方面。一是对于毕业设计的重视程度不够。目前，多数的大学生没有充分认识到毕业设计的重要性，他们只是把毕业设计环节作为大学生活的必经环节，纯粹地为了完成学业而去做毕业设计。在这种前提下，出现了一题多做，多人做一题的现象，甚至有些学生为了减少精力投入，别人做什么题目，他也做什么题目，其目的就是为了偷工减料，将别人的研究稍作变动就变成自己的“研究成果”。有些学生更加“精明”，干脆几个人做同一个题目，其内情也就不言自明了。二是选题没有一个确切的目标。不少同学对毕业设计的含义理解得不够深刻，只是为了做设计而选题，没有一个具体的目标。

2. 不重视可行性和需求分析

可行性研究阶段的任务不是具体解决某个问题，而是研究问题的范围，探索这个问题是否值得去解决，是否有可行的解决办法。在这个阶段应该导出系统的高层逻辑模型，并在此基础上更准确、更具体地确定工程规模和目标。

需求分析是软件开发的起始部分，也是软件开发中最关键的部分，是准确地回答“系统必须做什么”的问题的部分，对用户需求的理解是否正确直接决定了系统构建的正确与否。因此，必须要对用户需求作出全面、正确的分析，这对于软件开发的成败起着至关重要的作用。

在毕业设计过程中，学生往往认为编程是最重要的，因此，对软件工程中的可行性和需求分析阶段不重视，甚至在软件开发过程中，一边写程序，一边了解需求，程序完成了才清楚需求。

3. 设计过程混乱无序

软件设计的过程中，一般要求经过从概要设计到详细设计、模块设计后才能进行编码，而在毕业设计中，大部分的软件开发跨越了前期的设计过程，直接进行编码；或者在整个的设计过程中，没有具体的规划，往往是想起来哪个部分就做哪个部分，甚至有些部分做了一半，就又开始了另外一个阶段的工作，多个阶段同时进行导致思路混乱不清。这种无序的软件设计只能带来软件结构混乱、功能代码重复开发的问题，并导致系统集成性差，这样会极大地降低毕业设计的质量。

4. 开发过程中文档的不完备

软件工程要求系统开发过程中的文档必须和所开发的软件完全一致，文档要清楚地说明系统开发到某一阶段时已经做了什么，下一步应该做什么，以及算法的精确描述或出处等，以便毕业设计小组成员之间进行有效的交流。同时这些文档也是备忘录，对系统的维护和更新意义重大。不合格、不完备的文档会给开发人员内部的工作交流带来诸多不便，更严重的是，如果没有准确的文档，那么在进行诸如修改系统中的错误、改进

和增添新功能、移植到其他软件环境等这些工作时都将非常难做。

5. 不重视软件测试工作

软件工程要求系统开发过程中必须要进行测试,通过各种类型的测试使软件达到预定的要求。但在毕业设计过程中,由于多种因素,学生往往不重视测试工作,甚至省略该步骤。

1.3 用软件工程思想指导毕业设计

从软件工程的角度去指导学生进行毕业设计,使学生明白在毕业设计的过程中,要系统地将所学知识应用于实际问题,按软件工程思想展开工作。

1.3.1 以可行性研究指导选题,严把选题关

可行性研究的目的是不是解决问题,而是确定问题是否值得去解决。一般来说,至少应该从技术可行性、经济可行性、操作可行性三个方面去研究,必要时还应该从法律、社会效益等更广泛的方面去研究。

毕业设计题目的选择应立足于解决实际工程问题,注重提高学生发现问题、分析研究问题和开发创新的能力。

在选题原则上,首先,要从本专业的培养目标要求“重基础,强应用”出发,体现本专业基本训练的内容,对于需求分析、系统设计、代码实现、系统调试、系统运行和维护、资料查阅和外文阅读、翻译等内容应综合考虑,使学生得到比较全面的综合训练,而且毕业设计工作要有一定量的编程要求,以保证学生有明确的工作成果;其次,选题应具有独立性和完整性,原则上一个学生对应一个题目,但当几名选择选择一个课题时,必须明确每个学生应独立完成任务,保证其主要部分为独立进行的工作,使每个学生都受到比较全面的训练;最后,鼓励学生自己到社会、生产企业中寻求研究课题,培养他们自主学习的意识。

1.3.2 需求分析是毕业设计的关键

用户需求的获取是软件开发中最重要的工作,也是难度最大的工作。它不仅要涉及软件开发方面的知识、应用领域方面的知识,还会涉及与用户沟通的能力。

在软件开发时,需求的获取和分析由系统分析员负责。系统分析员可以通过多种形式获取用户需求,对用户提出的需求进行分析,以此来确定其中要实现的系统功能;对需求划分优先等级,找出核心需求和重要需求,以保证系统构建的正确性。由于系统分析员和系统程序员对于需求的理解不同,可能会出现程序员完成的系统与需求不一致的情况,容易导致最终开发完成的产品与最初的设计要求不符。因此在进行系统开发时,还需要对某些核心需求进行需求跟踪。需求跟踪是为了保证开发出的软件产品符合用户需求。使用需求跟踪可以随时了解用户需求的实际情况,可以有效避免在开发出的产品中遗漏用户需求或程序员即兴增加功能等情况的出现。

随着项目的进展,会有一些新的需求产生,也会有一些原来的需求产生变化,这种现象叫作需求变更。对于需求变更,首先要进行分类,判断出该变更是不影响总体设计

的变更和还是影响总体设计的变更。在进行项目开发时，要尽量避免影响总体设计的变更，因为影响总体设计的变更需要修改系统的构架，重新对系统功能进行设计和实现，这样会大大延长系统的开发时间和周期。

在进行毕业设计时，系统分析员一般由学生来担任，由于学生自身能力和经验的限制，不可能把获取用户需求做得十分完善。为了保证需求获取的准确性和全面性，可以让项目组的所有学生都参与到用户需求获取的工作中来，发挥集体的能力，并由指导教师进行最终确认。

1.3.3 规范设计过程

需求分析完成后的下一个阶段就是系统设计，系统设计包括概要设计和详细设计。概要设计阶段主要是确定方案，这个阶段要回答的问题是：概括地说，应该如何解决这个问题？在目标和成本间做出选择，选择一个合适的方案；然后设计软件的结构，确定程序的模块组成以及模块间的软件结构。在实现上可以是确定程序由哪些函数构成，每个函数的功能及参数，以及函数的调用关系，并根据模块划分及功能完成系统测试用例和系统测试计划。在这个阶段里应该完成的交付物是概要设计说明书和集成测试用例。

详细设计阶段的任务就是深入每个函数内部，以伪代码或注释的形式实现整个函数，但是本阶段不编写代码，而是设计出程序的详细规格说明，并根据函数语句的流向写出每个函数的单元测试用例和单元测试计划。

从系统设计开始，学生要按照软件工程的要求，画出 DFD 图，对输入、输出数据流进行详细的说明，建立符合软件工程规范的数据字典，根据 DFD 图设计出模块结构图。

1.3.4 重视文档的写作

软件工程要求在整个软件开发过程的每一阶段，都要有相应的文档，用来说明系统开发到该阶段时已经做了什么，下一步应该做什么，以及算法的精确描述或出处等。

在毕业设计的过程中，要严格遵照软件文档的思想，记载设计过程的每一阶段分别完成了哪些设计，以及下一阶段的任务、目标及技术手段，这样，把文档作为整个设计的指挥棒，使设计有条不紊地进行。另一方面，文档还可以方便毕业设计小组成员之间进行有效的交流，同时这些文档也是备忘录，它们对系统的维护和更新的意义重大。

1.3.5 重视测试工作，撰写毕业设计说明书

软件测试工作也是软件开发的重要组成部分。对于毕业设计来说，实际应用情况和测试数据是不可缺少的。通过实际应用效果和测试数据，毕业设计的成效便一目了然。

毕业设计主要应该进行三种形式的软件测试：单元测试、集成测试和系统测试。

单元测试由系统程序员自己来完成。系统程序员在完成一个模块的编码后，就要对这个模块进行相应的黑盒测试和白盒测试，测试通过后才标志着该模块编码完成。单元测试要达到两个目的：首先要验证自己的程序是否完成了相应的功能；另外要对程序的所有的分支进行测试，以确保每个分支是可以正确执行的。

集成测试由系统测试员来负责。系统测试员需要对系统程序员完成的模块进行集成。集成测试的主要目的是把通过了单元测试的程序模块拿来，构造成一个构件或子系统，

并对构造好的子系统进行测试，保证新的构件或子系统能够正确执行。

系统测试由系统测试员负责。系统测试主要用于测试整个系统的整体功能是否正确，是否与当初用户提出的要求相同。经过系统测试的软件才是最终的软件产品。系统测试的作用不光是能修正特定的错误，还能够进一步改进软件的质量。

毕业设计说明书的撰写是最后一个阶段。这一阶段要求学生回顾设计过程，总结劳动成果，这时学生就会对软件开发及必须经历的几个步骤都有更深刻的认识。最后，要求学生毕业设计进行系统的总结，写出毕业论文。

第2章 软件工程模型

本章主要介绍软件工程中的两个重要概念：软件生命周期与软件过程。在早期的软件工程中，这两个概念常常不加区分地使用，后来演变为既相互联系又有重要差异的不同概念。本章将对它们及相关概念加以说明。

2.1 软件及软件工程

2.1.1 软件概述

现在，绝大部分人都知道计算机系统由软件和硬件两大部分构成，他们也大概了解软件，但是，究竟什么是软件呢？

软件的定义是随着计算机技术的发展而逐步完善的。在 20 世纪 50 年代，人们认为软件就等于程序；60 年代人们认识到了软件的开发文档在软件中的作用，提出了软件等于程序加文档的说法，但这里的文档仅是指软件开发过程中所涉及的分析、设计、实现、测试、维护等，不包括管理文档；到了 70 年代人们又给软件的定义中加入了数据。因此，软件是计算机系统中与硬件相互依存的一部分，它包括以下几点。

- (1) 在运行中能提供所希望的功能与性能的程序。
- (2) 使程序能够正确运行的数据及其结构。
- (3) 描述软件研制过程和方法时所用的文档。

2.1.2 软件工程的定义及其研究内容

什么是软件工程呢？自从 1968 年第一次提出软件工程的定义以来，软件工程的定义也在一直不停地完善着。IEEE(IEE93)对软件工程的定义为：软件工程是将系统化的、严格约束的、可量化的方法应用于软件的开发、运行和维护中，即将工程化应用于软件。

通俗地说，软件工程是指导软件开发和维护的一门工程学科。它采用工程的概念、原理、技术和方法，把经过时间检验而证明的正确的管理技术和当前能够得到的最好的技术方法结合起来，用于开发和维护软件。

软件工程是一门综合性的交叉学科，它涉及哲学、计算机科学、工程科学、管理科学、数学及应用领域的知识。软件工程研究的内容主要集中在软件的开发技术与管理技术两大方面。开发技术包括软件的开发模型、开发过程、开发方法、工具与环境等；管理技术包括人员组织、项目计划、标准与配置、成本估算、质量评价等。

从另一方面来说，软件工程又是一种层次化的技术，如图 2-1 所示。因为任何工程方法都必须以质量控制为基础，因此质量控制是整个软件工程的基础。保证软件开发质量的前提条件是对软件工程中的各个过程进行有效的管理，因此必须为软件过程规定一

系列的关键过程域，以此作为软件项目管理控制的基础，通过人员组织管理、项目计划管理、质量管理等环节来保证软件开发按时按质量完成。软件工程中的“方法”提供了实现软件过程的技术，它涉及一系列的任务：需求分析、开发模型、设计、编码、测试和支持等。利用“工具”可以为软件工程的过程与方法提供自动或半自动的支持，在适当的软件工具辅助下，开发人员可以既快又好地做好软件开发工作，这些工具被称为 CASE(计算机辅助软件工程)工具。所以，一般将“过程”、“方法”和“工具”称为软件工程的三要素，这也是现代软件工程的研究内容。



图 2-1 软件工程的层次划分

2.1.3 软件工程的作用

软件工程的目的是提高软件的质量与生产率，最终实现软件的工程化管理和工业化生产。而质量与生产率往往是一对矛盾，软件的供需双方由于其利益的不同，其关心的焦点也不同。质量是软件需方最关心的问题，需方要求供方提供货真价实且满足需求的软件产品；而生产率则是供方最为关心的问题，供方追求的是高的生产率，以获得最大的利益。因此，如何在提高生产率的情况下开发出高质量的软件，必然就成为了软件工程的主要目标，好的软件工程方法可以同时提高质量和生产率。

由于软件工程一开始是为了应对“软件危机”而提出的，因此，如果软件在开发过程中能较好地利用软件工程的原理对软件开发的过程进行有效的管理，就可以充分保证软件开发的质量和生产率，反之就有可能导致项目的失败。

一般来说，软件企业的专业人员应该由以下几个层次构成。

(1) 高层管理人员。是软件企业的管理者，应具备软件专业的宏观知识、软件工程的管理知识和商业与资本的运作知识。他们是使用软件工程原理对企业进行管理的决策者。

(2) 项目经理与程序经理。是程序员的管理者，也是软件工程的拥有者与实践者。他们应具备系统分析与系统设计的能力和项目管理方面的知识。他们必须使用软件工程的理论与方法来对软件的开发过程实施管理。在整个团队中，项目经理与程序经理的技术水平、办事效率应该是最高的，而且他们具有较高的人格魅力。

(3) 程序员。是软件开发团队中的基础人员，应占整个企业员工的大多数。他们应具备阅读相关文档的技能和出色的程序设计与程序测试能力。他们要使用软件工程的理论与方法来实现软件项目的功能、接口与界面。

(4) 营销人员。是软件企业的形象代表，必须通过他们才能将软件产品推销到使用单位。他们必须具备商业营销的知识和软件工程的基本知识。他们既是某个行业领域的产品专家，又要成为该产品的实现顾问。

(5) 售后服务人员。代表企业直接与用户打交道，实施软件的安装、运行与维护。他们也需要用软件工程的方法来对软件进行实施维护。

在以上几类人员中，前三类人员必须掌握软件工程的原理，后两类人员只需要了解软件工程的相关知识并能将其应用于实践之中即可。所以，只要是在软件行业里工作，就必须重视软件工程、学好软件工程、用好软件工程，不断地将自己的实践经验上升到

软件工程的理论与方法层面，不断地用软件工程的理论与方法指导自己的实践，使自己得到升华与发展，这就是软件工程的作用。

实际上，软件工程的作用是多方面的：对一个软件项目团队来说，实施软件工程可以保证在规定的时间内，按照规定的成本来完成预期质量目标的软件；对软件企业来说，软件工程可以规范软件的开发过程和管理过程，不断地提高软件组织的个人素质和集体素质，从而逐渐提升软件企业的市场竞争力；对软件的整个发展进程来说，软件工程可以克服“软件危机”，控制软件的开发进度，节约开发成本，提高软件的质量。

2.2 软件工程的基本原理

既然是工程，就有许多相关的准则与基本原理，软件工程也不例外。自从1968年第一次提出软件工程的概念以来，全世界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与基本原理，1983年，B.Woehm对这100多条准则进行了总结归纳，提出了软件工程的七条基本原理，他认为这七条原理是保证软件产品质量和开发效率的最小且相当完备的集合。这七条基本原理如下。

1. 用分阶段的生命周期计划严格管理

统计表明，在不成功的软件项目中有一半左右是由于计划不周全造成的，由此可见，把建立完善的计划作为第一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的生命周期中，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，在软件的整个生命周期中应该制定并严格执行六类计划，它们分别是项目概要计划、里程碑计划、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

Boehm当时就已经认识到，软件的质量保证工作不能等到编码阶段结束之后再进行。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%，而编码错误仅占37%；第二，错误发现与改正得越晚，所需付出的代价也就越高。因此，在每个阶段都应该进行严格的评审，以便尽早发现在软件开发过程中所犯的错误，这是一条必须遵循的重要原则。

3. 实行严格的产品控制

在软件开发过程中不应随意改变需求，因为改变一项需求往往需要付出较高的代价。但是，在软件开发过程中改变需求又是难免的，由于外部环境的变化，相应地改变用户需求是一种客观需要，显然不能硬性地禁止客户提出改变需求的要求，因此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，为了保证软件各个配置成分的一致性，必须实行严格的产品控制，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：一切有关修改软件的建议，特别是涉及对基准