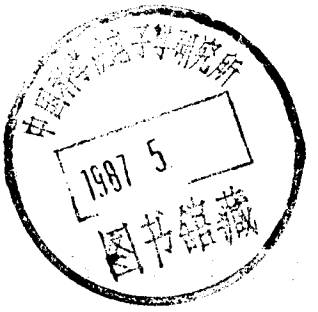
软件工程指南

RUANJIAN GONGCHENG ZHINAN

软件工程指南

朱三元等 编译



上海翻译出版公司

软 件 工 程 指 南

朱三元等　编译

上海翻译出版公司
（上海武定西路 1251 弄 20 号）

新华书店上海发行所发行　　上海东方印刷厂印刷

开本 787×1092　1/16　印张 11.5　字数 260,000
1985 年 9 月第 1 版　1987 年 1 月第 2 次印刷
印数 20,001—23,000

统一书号：13311·17　　定价：2.54元

前　言

六十年代末期产生的“软件危机”，主要表现为：大型软件错误很难消除、软件生产进度无法预测、成本增长速率失去控制、程序员的人数增长要求难以满足等等。面临这种情势，人们意识到：依靠传统的方法再也无法驾驭极为复杂的软件生产。于是，有人提出了用系统工程学的原理和方法来管理软件生产过程的设想。到1968年，在北大西洋公约组织(NATO)主办的会议上，才正式提出了称为“软件工程”的术语，目的是想凭借传统工程设计的一些基本思想，突出软件生产中的科学方法，以此来减少软件结构、软件管理的复杂性和提高软件的品质。

经过廿年的悉心研究，使我们获得了许多软件生产过程中的重要知识，而这些知识已逐步趋向传统工程学科中的重要知识了。人们基于这些知识，创造了许多能适用于软件生产中几个阶段的新的方法和技术，从而确立了这些阶段的生产规范和组织管理制度。如果把这些组合起来，就成为生产软件所必须使用的方法和技术、工具和环境、管理和规范，并使整个生产过程中各个阶段之间的转移简化和有章可循，推动了整个生产过程逐步走向自动化的进程。

过去，一提到软件生产或者软件开发，往往容易理解为“那就是编制程序”。但随着软件工程的发展，使人们清楚地认识到软件产品和其他工业产品一样，也具有设计、测试、检查等不可缺少的过程，从而逐渐形成了“软件产品生存期”的概念。一个软件产品的生存期，可以划分成若干个彼此既有区别又有联系的阶段。每个阶段中的工作都以上一阶段工作的结果作为依据，同时又为下一阶段的工作提供前提。工作中造成的差错越是发生在生存期的前期，则造成的影响和损失就越大，因而发现越早越好。若发现越迟，那末为了纠正错误所花费的代价就越高。由此使人们懂得，上一阶段的工作在没有做好之前，决不要草率地进入下一阶段，这样将有助于显著提高软件质量和降低软件生产成本。

关于软件产品生存期的划分方法，至今尚未有统一的认识，可是各种方法之间都有着许多相同之处。在本指南中，我们考虑将生存期划分为定义、开发、维护三个阶段。其中定义阶段又分为系统定义、软件计划、软件需求分析、计划修改等步；开发阶段分为设计、编码、模块测试、集成测试、确认测试等步。

在软件产品的生存期各阶段中，可以采用许多方法和工具，以及开发和维护的环境。目前，国际上流行两类方法，而这两类方法是从生存期开始到结束都要采用的，它们是系列方法。第一种方法是面向数据流的方法，即结构化方法，也称之为Yourdon方法。第二种方法是面向数据结构的方法，也称为Jackson方法。各种方法都有其优缺点，对方法的评价主要看它所应用的领域。虽然所有的软件都能表示成数据流图，而且在理论上，使用数据流图

可以适合于任何软件产品的开发工作。但也未必尽然，当系统具有良好定义的层次数据结构时，采用 Jackson 方法倒是恰当的。第一种方法在美国甚为流行，第二种方法在西欧比较流行。我们在本指南中采用的是第一种方法，对 Jackson 方法就不作介绍了。

“软件工程指南”的编译出版，相信会对我国从事软件研制、计算机应用开发的科技人员有所裨益。同时，由于本指南主要介绍了软件项目的计划、设计、编码、测试以及随后的软件维护等各方面的管理和技术方法，内容丰富齐全，具体实用，所以对于高等院校的高年级学生和研究生来说，也是一本较好的入门书。

本指南是由张作民、周之英、潘锦平、陆春江、何守才、林斌贵、石晓虹、张芳等同志和我一起共同努力、愉快合作，顺利完成编译工作的。倘若能对广大读者确有助益，乃是我们的最大愿望。

朱三元

1985.7.15

目　录

第十一章　管理问题

第十二章　一个实例

附录A　文档格式

附录B　软件方法和工具

第一章 导论

软件工程是一种开发和维护软件的规范化方法。它是从过去一般称之为“计算机程序设计”的活动中演化而来的。这本《软件工程指南》将帮助管理人员和技术人员深入理解软件工程处理中的每一步。

《软件工程指南》主要介绍了软件工程的计划、规范、设计、编码、测试，以及在项目活动结束之后的软件维护等各方面的管理和技术方法。详细描述了软件项目获得成功所必需的各种文档格式及复查过程。还规定了跟踪、报告和控制软件开发各部分的各种方法。

本章对软件工程作了一般介绍，并提出了软件工程指南（SEG）使用方法建议。以后各章详细讨论了软件工程处理中的每一步。

1.1 各种问题

在过去的十年里，计算机软件已变为以计算机为基础的各种系统的核心部件，正是软件决定了整个系统的成败与否，软件通过挖掘硬件设备的“潜力”提供各种功能，从而导致以计算机为基础的各种产品的“智能化”。

可惋的是，计算机软件的开发以及软件的获取有它自身的问题。在六十年代初期，软件开发费用在整个系统的成本中只占很小的百分比。硬件却非常昂贵，因而自然把管理工作和规程应用于控制硬件的成本。

随着微电子技术的迅速发展，硬件成本骤降。在五年之内，成本下跌二、三个数量级已经不是罕见之事。而另一方面，由于劳动密集，软件成本却象通货膨胀那样持续上涨。到1980年，软件在以计算机为基础的系统的开发中已经变为最主要的成本因素。

1.1.1 软件开发

和软件开发相关的各类问题的根源在于计算技术发展初期程序设计过程中缺乏适当的质量控制。在1960年，因为软件是个相对低成本的系统成分，所以未能得到管理部门的足够重视。由于管理方法常常不足，技术开发常有缺陷。在许多组织中这些过时的做法仍然存在，从而引起下列问题：

1. 软件成本及进度估计往往相当不准确。大大挫伤了开发者的信心，便导致企图走捷径而使质量下降，引起用户不满。

2. 即使在计划、规范书、设计、编码和测试中使用了经过证明的方法，但仍没有系统地开发软件。

3. 不适当的软件文档。计算机程序应该有一整套的文档。这套文档不是在软件开发之后而是在软件开发过程中编制出来的。它们作为软件开发的里程碑，能帮助管理人员控制并核查进度。

4. 软件质量值得怀疑。由于不可能一致地使用经过证实的软件质量保证技术（复查、审查及测试），导致了质量保证方面的各种问题。

5. 软件经常是“不可维护”的。要在许多程序中纠正潜在的错误是很困难的，要使这些程序适用于新的计算机或是增强用户所要求的功能，实际上也是不可能的。可重复使用的、通用性较强的软件，仍然是一个急于寻求而又尚未达到的目标。

上面提到的开发问题都可以通过应用本指南中所讨论的各种软件工程技术加以解决。

1.1.2 软件获取

软件作为一个有它自身权利的产品，已经成为计算机市场上的主要商品。许多管理人员因不熟悉计算机软件的获取技术，当购置了一个不能满足所述需求的程序之后将大失所望。同软件获取相关的这些问题和从外部购买产品的问题没有什么两样。然而，应该特别注意下列问题。

1. 获取的软件产品应符合所述的需求。买主必须恰当地理解并规定自己的需求和由买主提供的产品性能。完成这件事的方法有二种情况。

2. 如果软件产品为“适合”买主的环境而必须加以修改的话，则其费用必须是合理的，并且应评价其成本的有效性。通常修改软件的费用要比修改买主的使用环境便宜得多。

3. 卖主必须为软件产品提供适当的支持。象开发问题一样，软件获取的问题也能避免。和软件工程相关的规程亦可应用于计算机程序或系统的获取。

1.2 解决办法

解决上节中所描述的各种问题的一个办法是基于计算机软件“生存期”的观点推论出来的。象计算机硬件一样，计算机软件应该在一系列仔细地控制和系统化地执行过程中研制而成的。

1.2.1 生存期概念

在计算机软件生存期中有三个阶段：定义阶段、开发阶段和维护阶段。在定义阶段（也称计划阶段），要为软件项目做出计划、预算资金和进度，分析并且规定详细的需求。在开发阶段，用经过验证的各种设计、编码和测试方法把软件需求转变为一个可执行的程序，最后在维护阶段，纠正所遇到的各种问题，修改软件使它适合于不同的工作环境，以及增强功能要求。在软件生存期的每一个阶段都有一系列的工程步骤，每一步都以能加以复查并可移交才作为结束。

1.2.2 工程化方法

用于解决任何产品开发的一种工程化方法是：

- 要求在定义、开发和维护阶段的每一步中都采用经过验证的方法。
- 要求一系列的复查，以便在产品开发中保证质量。
- 规定在每一步中要产生的特定的文档。
- 鼓励能够加速开发的各种工具和方法的使用与研制。
- 提供从原始产品概念到最后产品制造的一个可追溯的途径。

任何规程的发展都经历了许多年。在经典领域中，诸如机械工程、电子工程，已经发展了通常可接受的工程科学。软件工程则是计算机软件走向工程科学的途径。

1.2.3 软件工程

软件工程是应用于计算机软件的定义，开发和维护的一整套方法、工具、文档、实践标准和工序。软件工程处理和它在硬件工程中相对应的部分十分相似。它显示出在第1.2.2节中所讨论的相同特性。

本书的下面各章将描述下列各步骤：

1. 计划——确定项目范围，规定资源预算，并且确定进度。

2. 需求分析——确定详细的功能和性能需求，设计制约条件及确认标准。

3. 设计——把需求转变为软件的体系结构表示和软件的具体工序表示的有系统的

过程。

4. 编码——用程序设计语言把设计的表示转变为一种机器“可理解”的形式。

5. 测试——软件工程的测试是依次实现下列三个步骤：

i)企图发现程序中的潜在错误，

ii)将软件装配成一个可工作的软件包，

iii)根据需求验收软件。

6. 维护——应用上述五个步骤(再加上另外的控制过程)管理现有的计算机程序。

软件工程是一组规范化的方法，通过应用这些规范，就能解决软件开发中产生的问题。

1.3 软件工程指南中引用的若干关键概念

本指南为软件工程的管理和实践描述了许多重要的概念。本节将综述这些概念。

1.3.1 生存期

在软件工程指南中贯穿着“软件生存期”概念的讨论。软件生存期建立软件工程事件的顺序表。

软件生存期是从软件作为以计算机为基础的系统的一个部分而开始的。当系统工程师给软件加上一定的功能或性能时就开始了生存期的定义阶段。

如前所述，定义阶段着重于软件项目的计划以及软件需求分析和规范书。用事先准备好的系统规范书作为一个定义，软件计划人员和分析人员要确定项目范围，并估计费用预算和进度。项目范围可以扩展成书写详细的需求规范书，以便复查。软件需求规范书将成为生存期开发阶段以后的全部工作的基本文档。

需求必须被转换为最终可以由计算机执行的形式。开发阶段开始这种转换，它应用各种设计方法产生一个软件结构和软件工序的表示，这种结构和工序的表示将通过编码产生程序设计源代码，即计算机程序。最后经过测试以保证质量并满足软件需求。

生存期的维护阶段涉及到前述各阶段的每项活动，但却是把软件工程应用于一个已经存在的程序。和维护活动相连的是一种正规的控制方法学，叫做软件配置管理，它可使程序在受到修改时能保证其完整性。

1.3.2 软件项目估算

软件工程指南中的许多地方都讨论了软件项目的计划和控制。软件项目估算在项目开始时就要向管理人员提供预算和进度指导。事实上，项目的“继续—停止”决策是基于用历史数据、经验模型或自动成本计划工具估计的预算和进度。

本指南还介绍了几种估算技术。每一种估算技术都要求透彻理解项目的范围，参考从过去项目中收集的数据，要完成的项目的逻辑分划，以及对用不同方法所得到的结果进行互相校核，以保证这些估算的实际意义。

1.3.3 需求分析

软件需求分析为后面介绍的软件工程工作提供了一个导向图。软件工程的每一道工序都必须显示出对需求的“可追溯性”，在定义阶段，当系统的所有组成部分(例如硬件、软件、信息、人员)已被验明、各个组成部分之间的接口也已规定时，则需求分析也即开始。可以使用人工的方法或者使用各种自动工具，把信息结构和信息流模型化，从而研制出一份软件规范书。

1.3.4 软件设计

设计是软件工程的技术核心。设计人员开始把软件需求映射成一个相应的体系结构，结构中的各个组成部分是意义明确的模块，每个模块都针对某一方面的需求。采用

许多种设计方法中的某一种，对软件的体系结构求精，并编制文档和进行复查。随着设计过程的深入，设计者关注的范围逐渐缩小。当软件体系结构已经建立，就可以把各种程序设计技术和工具分别应用于每个模块。产生的详细设计则是结构化编码的基础。

1.3.5 结构化编码

软件设计必须转换成机器可执行的表达形式。结构化编码采用一套程序设计规则，使得程序设计语言“源程序清单”呈现良好的程序风格，清晰的文档和格式。由于设计过程也已经结构化，在程序设计和源程序清单之间大体上有一一对应关系。

源程序清单是编译程序的输入，编译程序把源程序自动地转换为机器可执行的形式。

1.3.6 测试步骤及方法

在这本软件工程指南中，多处涉及保证软件质量的各个方面，软件测试是许多质量保证技术中的一种。

软件测试从模块一级开始。软件工程采用各种测试实例设计方法以及/或者使用自动工具，以保证发现潜在的错误。对每个模块进行测试之后，必须和其他已测试过的模块进行装配，从而形成一个完整的系统。集成测试是装配和测试计算机程序的一套策略和方法。最后，对组装后的软件还必须经过有效性测试，以保证满足全部需求。

1.3.7 小型项目的软件工程

软件工程可适用于任意大小的开发项目。然而承担各种小型项目时（例如，低于六个人月工作量，非关键性软件，以及常规的应用），软件工程过程中的有些步骤便可以简略，文档可以缩短和合并，复查可以不那么正式，但不可不彻底。本指南为小型软件项目的技术处理方法和管理提供了一套准则。

1.3.8 软件维护

通常，计算机软件在开发完成之后，可以使用十年至廿年。在此期间，应该修正软件在使用中出现的错误，完成对适应新环境的软件的修改以及增强系统功能的开发。总的说来，这些活动就是软件维护。在计算机软件的总成本中，这个阶段的耗资高达百分之六十以上。

从理论上说，软件维护期间采用的各种方法和工序与在软件开发阶段中采用的各种方法和工序大体上是相同的。不幸，没有按照软件工程规程开发出来的程序往往缺少文档，按照一种拒绝修改或者无法修改的方式设计出来，而且在过去，这些软件已经遭到任意的（有时甚至是无记载的）改变。只有通过软件工程的方法才能减轻软件维护的负担。

1.3.9 软件配置管理

在整个软件生存期中会产生各种文档，报告，清单和数据。这些项目构成软件配置，也就是以各种形式描述软件的信息集合。软件配置管理（简称SCM）是软件项目的标记、控制、审查、记帐的一整套活动。

本指南中专门有一章讨论软件配置管理，由于和软件配置管理相关的各种活动运用于整个软件生存期，因而必须恰当地标记各种文档，以便于归档和检索。对更改必须加以审查，以确保遵循标准和工序。而且必须把项目的当前状态报告给管理部门。

1.3.10 文档和复查

软件生存期的每个步骤都产生一个文档，这个文档就成为软件配置的一部分。文档是软件工程每个步骤的自然结果。文档复查则标志项目的里程碑。

1.4 软件工程指南的目的

这本软件工程指南能满足两种需要：

1. 作为一本教科书，运用工业环境的术语来描述软件工程处理过程。

2. 作为一本手册，为各道工序建立起软件工程方法学。软件工程手册的基本目标是：

i)为软件管理提供一套准则、方法和工序，以控制软件生存期中的每个步骤。

ii)为软件专业人员提供适用于需求分析、设计、编码、测试及维护的一整套方法和工具。

iii)规定各种文档格式、复查技术和配置控制过程，有助于确保软件质量。

iv)介绍一个能适用于各种应用和环境的软件工程规程。

软件工程指南是规范的一种综合性的、统一的描述，它涉及到管理理论和计算机科学领域中的许多问题，它在介绍一些已经被誉为“八十年代最重要的工程规程”中起到了极为重要的作用。

1.4.1 管理人员的指南

本书对已经直接或间接地负责过软件开发或获取的管理人员来说是一本很好的指南，它能满足以下许多需要：

(1) 为软件项目的计划和估算提供技术上的准则。介绍了软件项目的成本预算和进度的各种技术，并且详细说明了在一个工程项目初期就应该考虑到的某些要点。

(2) 提出了技术人员的组织结构的各种方案，详细讨论了小组组织结构。

(3) 详述了工程项目的里程碑和适合于评价这些里程碑的方法。通过回答“我所预期的是什么？应该如何表现它？何时应该提交？”等等问题，向管理人员提供贯穿软件项目始终的各个参考要点。

(4) 规定了专门的控制方法，用于在软件配置逐渐形成时对其加以管理。控制使管理人员在任何时候都知道项目的状态。

(5) 描述了许多种质量保证技术，包括复查工序、测试方法、配置控制以及管理等。

(6) 为编制正式的软件工程标准和工序手册提供了依据。

尽管一个管理人员应该知道软件工程指南中所讨论的每个问题，但许多章节事实上是技术性的，不必详细加以研究。本章第六节提供了本指南的使用指导。

1.4.2 专业人员的指南

应用软件工程方法和工具是专业技术人员的职责。软件工程指南通过对下列问题的阐述，指导软件工程的实践。

(1) 指南描述了软件需求规范书、设计、编码和测试的各种技术、工具与表示方法。指南概括了来自各个方面的信息，从而使专业人员能够回答诸如“什么方法可以适用?”，“每种方法的优缺点是什么?”这样一些问题。

(2) 指南描述了复查工序的方法，解释和论证了每项软件复查。

(3) 指南介绍了一套工具，这套工具中包括了图形表示法（如数据流图、结构图）和自动工具（如PSL/PSA，PDL）。

(4) 指南阐述了文档格式及其内容，对软件配置中的每一项都作了细述。

尽管专业人员的兴趣是各式各样的工具和技术，然而软件工程指南中几乎所有部分同样会受到普遍关心(见1.6节使用指导)。

1.5 指南的使用

软件工程指南提出了软件生存期的每一个阶段。

第二章　是软件工程工序概述，把软件工程介绍给那些不熟悉软件工程规程的人。

第三章至第七章　根据软件工程在定义

和开发阶段的产生顺序，叙述了软件工程的各个步骤。

第八章　描述了软件配置管理，即适用于整个软件生存期的一套标准和工序。

第九章　深入讨论了软件维护阶段。

第十章　详细讨论了软件工程方法用于小型项目时要作的调整。

第十一章　给出各种和软件开发相关的管理信息文章。

第十二章　详细介绍了一个实际项目的软件配置应用实例。

附录A　详细介绍文档格式。

附录B　讨论了各种辅助方法和工具。

1.5.1　方法与工序的来源

本指南的第三至第十章是各种软件开发方法和工序的来源。第三章描述了管理计划，估算方法以及项目追溯和控制的过程。第四章描述了软件需求分析，写规范书的方法和工具。第五章描述了概要设计和详细设计的方法、工序和工具。在第三至第五章中还讨论了文档和复查工序。第六章阐述了编码风格和清晰性，以及编码文档可移植性和编码工具的有关问题。第七章介绍每个测试步骤，还讨论了测试策略、技术及工具。第八章概述了软件配置管理，介绍了控制软件配置所必须考虑的过程和组织方面的问题。第九章描述了软件维护的各道工序和文档。第十章介绍了对较小的项目所需做的方法和工序的调整。这一章是以第三至第九章中提供的材料为基础的。

1.5.2　文档格式的来源

在本指南的附录A中，包含了软件配置中每一项的详细文档格式。附录A.1至A.9包括了每种文档的要点以及对于每种文档内容的逐段描述。

第十二章介绍了一个完整的软件文档例子，给出实例规范书，设计文档，源程序清单以及测试文档的示例。

1.5.3　一般的信息和参考文献信息的来源

软件工程指南中有多处介绍了和软件管理、软件开发相关的一般问题。第二章和第十一章是这些信息的主要来源。

1.6　如何使用软件工程指南

软件工程指南是同时作为教科书和参考书而设计编写的，一旦理解了整个软件生存期，考查了软件开发过程的各个步骤以后，其每章都可以独立阅读。

读者都应阅读第一章和第二章，因为这两章介绍了本指南的目的和内容，并且概述了软件工程。读者可以从中选取自己感兴趣的部分。下面各节将向您建议如何使用软件工程指南。

1.6.1　题目-读者相关对照表

尽管概括性论述要求相当的熟练程度，但软件工程指南中的某些题目仍将引起所有读者的关心。其中有些题目可能使管理人员感兴趣，而另一些题目则对技术人员有特殊的吸引力。图1.1的“题目-读者相关对照表”列出了软件工程指南中提出的34个主要题目，以及每个题目所面向读者范围的建议。用“A”标注的应是所有读者关心的题目，这些题目构成指南材料中应该精通的最小子集。管理人员应该阅读从“A”型题目扩大到用“M”标注的所有题目。技术管理员还可以进一步扩大选择技术范畴的题目（用“T”符号标注的）来阅读。专业人员的阅读范围应该从“A”型题目扩大到用“T”标注的所有题目。

在图1.1中：

M——基本上是管理人员关心的题目；

T——主要是技术人员感兴趣的题目；

A——所有人都关心的题目。

	章												附录	
感兴趣的题目	1	2	3	4	5	6	7	8	9	10	11	12	A	B
软件工程概述 开发小组 成本估算方法 软件生产问题	A	A M	 M M											
自动工具和成本 成本因素 项目进度 需求分析任务 分析员的特性			M M M	 A A										
规范书格式 设计问题 概要设计方法学 设计复查 详细设计工具 设计遍查 结构化程序设计 编码风格及清晰度 防错 编码工具				A	 A T A T A	 T T T A								
软件测试问题 软件测试策略 软件测试方法 测试小组 排错							A A T M T							
测试工具 配置管理 修改控制 软件维护 小型项目							T	 A M	 A	 A				
组织结构 标准 教育 文档格式及内容 辅助方法及工具											M M A	 A	 A	 T

图 1.1 题目-读者相关对照表

1.6.2 对管理人员阅读方法的建议

下列的指导意见表示对不同级别和兴趣的管理人员建议的各章阅读顺序。章节排列顺序是用章节号及附录字母表示的章节阅读顺序。括号中的数字或字母表示可以选读的章节。

高级管理人员	1,2,(11)
中级(项目)管理人员	1,2,3,8,9,10,11,(4—7, A)
技术管理人员	1,2,3,4,8(5—7)9, 10, (11), A, 12(B),

1.6.3 对专业人员阅读方法的建议

下列指导意见表示对不同级别和兴趣的专业人员所建议的各章阅读顺序。章节阅读顺序与上述 1.6.2 节相同。

系统工程师和分析人员	1,2,4,8,(3),(5—7,9—12,A,B)
软件设计人员	1,2,4,5,8,9,(3),6,7,(10—12A)
软件工程师	1,2,4—9,(3,10,11),12,A,B

1.7 小　结

软件工程指南适用于不同读者的需要。把本节给出的阅读指导和目录表，以及索引结合起来，为你在软件工程的学习、研究中提供了极为有用的指南。