

中华人民共和国国家标准

软件开发规范

Discipline for Software Development

(报批稿)

目 录

1 引言	4.4 软件质量保证
1.1 目的和作用	4.5 确认测试
1.2 适用对象及范围	4.6 软件配置管理
2 定义	4.7 软件开发管理
2.1 文件	4.8 文件编制
2.2 规格说明	5 软件开发流程
2.3 软件生存周期	5.1 可行性研究与计划
2.4 软件管理	5.2 需求分析
2.5 内聚度	5.3 概要设计
2.6 耦合度	5.4 详细设计
2.7 软件配置管理	5.5 实现
2.8 模块开发卷宗	5.6 组装测试
3 软件生存周期	5.7 确认测试
4 总体要求	5.8 使用和维护
4.1 设计要求	附录:
4.2 软件实现	1. 软件问题报告
4.3 软件测试	2. 软件修改报告

1 引言

1.1 目的和作用

本规范详细规定软件开发过程的各个阶段及每一阶段的任务、实施步骤、实施要求、完成标志及交付文件。本规范的目的是使整个软件开发过程阶段清晰、要求明确、任务具体,使之规范化、系统化和工程化,向广大从事软件开发的技术人员和管理人员提供一系列行之有效的准则、方法和规程。

使用本规范,有利于提高软件开发过程的能见度,有利于开发过程的控制和管理,便于采用工程化的方法开发软件,从而提高所开发软件系统的质量,缩短开发时间,减少开发和维护费用,便于软件开发和维护人员之间的协作、交流,使软件开发活动更加科学、更有成效。

1.2 适用对象及范围

本规范是为软件管理人员、软件开发人员、软件维护人员、客户及用户制定的。其中:

软件管理人员——在软件开发过程中负责技术管理及项目管理的人员;

软件开发人员——在软件开发过程中从事需求分析、设计、编码、测试及文件编写的人员;

软件维护人员——负责对软件进行修改和维护的人员;

用户——所开发软件的使用者;

客户——指个人或单位,他们为产品的开发提供资金,通常(但有时也不必)还提出各种需求;

本规范旨在适用于尽可能广泛的软件系统的开发工作,包括不同类型的系统。

本规范所推荐的文件,在使用过程中可根据所

开发系统的不同规模进行选择。在“软件产品文件编制指南”中，将以文件的树形结构方式提供文件的选择。各文件的具体内容和格式可参见国家标准“软件产品开发文件编制指南”。

2 定义

2.1 文件 (document)

与程序开发、维护和使用有关的材料，它是软件的重要组成部分。

2.2 规格说明 (specification)

一种文件，它对软件所应满足的要求以可验证的方式做出完整、精确的陈述。规格说明的内容，可根据不同场合的需要而有所侧重。如：

功能规格说明——对软件所应具备的功能作出规定。

性能规格说明——对软件所应具备的内容(如：计算精度、响应速度等等)作出规定。

接口规格说明——对软件及其环境之间，软件各部分之间的接口关系作出规定。

2.3 软件生存周期 (software life cycle)

软件产品从形成概念开始，经过开发、使用 and 不断增补修订，直到最后被淘汰的整个过程。

一个软件产品的生存周期可划分成若干个互相区别而又有联系的阶段。每一个阶段中的工作均以前一阶段工作的结果为依据，并作为下一阶段工作的前提。

开发单位的技术负责人可根据所开发软件的性质、用途及规模等因素决定在软件生存周期中增加或减少相应的阶段。

2.4 软件管理 (software management)

软件开发与维护过程的管理，软件管理的任务是有效地组织人、技术和工具，来完成预定的软件项目。

2.5 内聚度 (cohesion)

单个程序模块所执行的诸任务在功能上的互相关联的程度。

2.6 耦合度 (coupling)

计算机程序中模块之间相互依赖的量度。

2.7 软件配置管理 (software configuration management)

标识和确定系统中配制项的过程，在系统整个生存周期内控制这些项的投放和更动，记录并报告配置项的状态和更动要求、验证配置项的完全性和

正确性。

2.8 模块开发卷宗 (modular development folder)

模块开发卷宗是在模块开发过程中逐步编写出来的。每完成一个模块或一组密切相关模块的复审时编写一份。在实现阶段结束时，应把所有的模块开发卷宗汇集在一起，以便于对整个模块开发工作进行管理和复审，为将来的维护工作提供有用的技术信息。

3 软件生存周期

本规范将软件生存周期划分为以下几个阶段：

可行性研究与计划；

需求分析；

概要设计；

详细设计；

实现；

组装测试；

确认测试；

使用和维护。

4 总体要求

约定：

甲方：软件使用部门

乙方：软件开发部门

4.1 设计要求

乙方有责任保证软件设计完全满足可行性研究报告中规定的所有技术要求，未经甲方同意不得修改要求。设计功能的复杂性和功能模块相互间的相关性必须降低到最低限度。

4.2 软件实现

应避免在设计尚未完成的情况下就急于编码的错误倾向。实现时应采用符合国家标准或国际标准的程序设计语言。应尽量采用结构化方法，使开发的软件具有良好的、清晰的结构。

4.3 软件测试

乙方必须全面完成所规定的测试(包括单元测试、组装测试、确认测试)，向甲方提出准确和完整的测试结果报告。必要时，甲方可委托第三方进行确认测试。

4.4 软件质量保证

甲、乙双方都必须把软件质量保证放在重要地位并制定软件质量保证计划，以确保完全遵守需求

说明中规定的所有的软件要求。为保证软件质量，乙方必须采用审查、文件评审、设计评审、审计、测试等具体手段。

4.5 确认测试

确认测试必须按双方协议执行，确认测试通常在模拟或模拟环境中实施，测试结果报告所说明的情形与软件要求不完全一致的偏差程度，必须限于正式批准认可的范围。

4.6 软件配置管理

乙方必须制订并执行配置管理规程，以便说明要管理的软件成分、定义及其修改的状态，同时要保证程序与其它文件完全一致。当乙方提出的修改影响到甲方的其它软件时，必须取得甲方同意。

4.7 软件开发管理

乙方必须设置或指定软件开发管理组织，实施对软件开发过程与资料的管理，并要制订执行一些规程，用来控制软件开发的整个过程，必须定期向有关部门和甲方提供工作进展情况。

4.8 文件编制

文件编制应以乙方为主，在编制过程中应充分听取甲方意见，使用的语言应通俗，无歧义，文件格式应按照国家标准“软件产品开发文件编制指南”进行书写，在开发过程中可根据系统的规模对文件的内容进行增删。

5 软件开发流程

5.1 可行性研究与计划

5.1.1 任务

了解客户的要求及现实环境，从技术、经济和社会因素等三个方面研究并论证本软件项目的可行性，编写可行性研究报告，制订初步项目开发计划。

5.1.2 实施步骤

5.1.2.1 软件开发部门的分析人员对客户要求及现实环境进行调查。

5.1.2.2 在调查研究的基础上编写有关客户提出的问题及书面材料。

5.1.2.3 依据书面材料及其有关资料对待开发的系统从经济、技术及社会因素等方面进行可行性研究，写出可行性报告。

5.1.2.4 评审和审批，决定项目是取消还是继续。

5.1.2.5 若项目可行，则制订初步的项目开发计划，并根据需要签署合同。

5.1.3 实施要求

5.1.3.1 经济可行性应是优先考虑的问题，它包括成本/效益分析，长期的经营策略等。

5.1.3.2 估计成本/效益应有确切的数据和估算方法，避免主观臆断。

5.1.4 完成标志

5.1.4.1 成本/效益分析应提供几种可供选择的解答。

5.1.4.2 项目开发计划中应具有明确的、可检查的标志。

5.1.4.3 所指定的文件要齐全，可验证。应交付的文件有：

- 可行性研究报告；
- 初步的项目开发计划。

5.2 需求分析

5.2.1 任务

确定被开发软件的运行环境、功能和性能要求，编写用户手册概要和确认测试准则，为概要设计提供需求说明书。

5.2.2 实施步骤

5.2.2.1 调查被开发软件的环境。

5.2.2.2 进一步明确客户和用户需求。

5.2.2.3 确定人机界面。

5.2.2.4 修改初步的项目开发计划。

5.2.2.5 制订确认测试计划。

5.2.2.6 编写用户手册概要。

5.2.2.7 评审。

5.2.3 实施要求

5.2.3.1 必须以运行环境为基础。

5.2.3.2 应该要有客户所指定的人员参加。

5.2.3.3 需求说明书必须明确，并经过客户确认。

5.2.4 完成标志

所指定的文件要齐全，并经过评审。应交付的文件有：

- 软件需求说明书；
- 修改后的项目开发计划；
- 用户手册概要；
- 确认测试计划；
- 数据要求说明书。

5.3 概要设计

5.3.1 任务

根据软件需求说明，建立目标系统的总体结构。

和模块间的关系,定义各功能模块的接口控制接口。设计全局数据库/数据结构,规定设计限制,制订组装测试计划。

5.3.2 实施步骤

5.3.2.1 建立目标系统的总体结构。

——对于大型系统,可按主要的软件需求划分成子系统,然后为每个子系统定义功能模块及各功能模块间的关系,并描述各子系统的接口界面。

——对于一般系统,可按软件需求直接定义目标系统的功能模块及各功能模块间的关系。

5.3.2.2 给出每个功能模块的功能描述,数据接口描述,外部文件及全局数据定义。

5.3.2.3 设计数据库或数据结构。

5.3.2.4 制订组装测试计划。

5.3.2.5 评审。

5.3.3 实施要求

5.3.3.1 在设计目标系统的整体结构时,应力争使其具有好的形态,各功能模块间应满足低耦合度,而各功能模块内应满足高内聚度。功能模块的作用范围应在其控制范围之内。

5.3.3.2 在设计目标系统的总体结构时,应降低模块接口的复杂性,提高目标系统的可移植性。

5.3.4 完成标志

5.3.4.1 所有已定义的软件需求均被所设计的系统所复盖。

5.3.4.2 建立了系统的结构,明确指出系统各模块的功能、模块间的层次关系及接口控制特征。

5.3.4.3 所指定的文件要齐全,可验证。

应交付的文件有:

- 概要设计说明书;
- 数据库(或数据结构)设计说明书;
- 组装测试计划。

5.4 详细设计

5.4.1 任务

对概要设计中产生的功能模块进行过程描述,设计功能模块的内部细节,包括算法和数据结构,为编写源代码提供必要的说明,建立“模块开发卷宗”。

5.4.2 实施步骤

5.4.2.1 将概要设计产生的构成软件系统的各个功能模块逐步细化,形成若干个程序模块(可编程模块)。

5.4.2.2 采用某种详细设计表示方法对各个程序模块进行过程描述。

5.4.2.3 确定各程序模块之间的详细接口信息。

5.4.2.4 建立“模块开发卷宗”。

5.4.2.5 拟定模块测试方案。

5.4.2.6 评审。

5.4.3 实施要求

5.4.3.1 确定程序模块内的数据流或控制流,对每个程序模块必须确定所有输入、输出和处理功能。

5.4.3.2 规定符号的使用,确定命名规则。

5.4.3.3 尽可能按结构化程序设计原则进行设计。

5.4.4 完成标志

5.4.4.1 详细地规定了各程序模块之间的接口,包括参数的形式和传送方式、上下层的调用关系等。

5.4.4.2 确定了模块内的算法及数据结构。

5.4.4.3 所指定的文件要齐全,可验证。

应交付的文件有:

- 详细设计说明书;
- 模块开发卷宗。

5.5 实现

5.5.1 任务

将详细设计说明转化为所要求的程序设计语言或数据库语言书写的源程序。并对编写好的源程序进行程序单元测试,验证程序模块接口与详细设计说明的一致性。

5.5.2 实施步骤

5.5.2.1 对每个程序模块用所选定的程序设计语言进行编码。

5.5.2.2 按照测试方案产生测试数据。

5.5.2.3 按照测试方案中规定的方法进行程序单元测试。

5.5.2.4 书写“模块开发卷宗”中相应于该阶段的内容。

5.5.2.5 编写操作手册和用户手册。

5.5.2.6 评审。

5.5.3 实施要求

5.5.3.1 尽可能使用符合国家标准和国际标准的程序设计语言。

5.5.3.2 为了提高程序的可理解性,应在源程序中加入适当的注解。

5.5.3.3 尽量采用增加程序可读性的排版格式。

5.5.3.4 不仅要考虑对合法的输入产生测试用例,而且要对非法的、非预期的输入产生测试用例。

既要对正常的处理路径进行测试,而且要考虑对出错处理路径进行测试。

5.5.3.5 程序模块的测试用例,预期结果及测试结果应存档保留。

5.5.4 完成标志

所指定的文件要齐全,可验证。应交付的文件有:

- 模块开发卷宗。

5.6 组装测试

5.6.1 任务

根据概要设计中各功能模块的说明及制订的组装测试计划,将经过单元测试的模块逐步进行组装和测试。

5.6.2 实施步骤

5.6.2.1 执行测试计划中所有要求做的组装测试。

5.6.2.2 分析测试结果,找出产生错误的原因。

5.6.2.3 提交组装测试分析报告,以便尽快修改错误。

5.6.2.4 评审。

5.6.3 实施要求

5.6.3.1 组装测试应保证模块间无错误的连接。

5.6.3.2 应对软件系统或子系统的输入、输出处理进行测试,使其达到设计要求。

5.6.3.3 应测试软件系统或子系统正确处理能力和经受错误的能力。

5.6.4 完成标志

所指定的文件要齐全,测试结果符合要求,并通过评审。应交付的文件有:

- 可运行的软件系统源程序清单;
- 组装测试分析报告。

5.7 确认测试

5.7.1 任务

根据软件需求说明书中定义的全部功能和性能要求及确认测试计划测试整个软件系统是否达到了要求。并提交最终的用户手册和操作手册。

5.7.2 实施步骤

5.7.2.1 在模拟的环境中进行强度测试,即在事先规定的一个时期内运行软件的所有功能,以证明该软件无严重错误。

5.7.2.2 执行测试计划中提出的所有确认测试。

5.7.2.3 使用用户手册和操作手册,以进一步证实其实用性和有效性,并改正其中的错误。

5.7.2.4 分析测试结果,找出产生错误的原因。

5.7.2.5 书写确认测试分析报告。

5.7.2.6 确认测试结果后,书写整个项目开发总结报告。

5.7.2.7 对所有的文件进行整理。

5.7.2.8 评审。

5.7.3 实施要求

5.7.3.1 必须邀请客户一起参加确认测试。

5.7.3.2 全部系统存储量、输入及输出通道,以及处理时间必须有足够的余量。

5.7.3.3 对于一次操作时间长于24小时的系统来说强度测试的最少时间须是25小时,对于一次操作时间短于24小时的系统来说,则最少测试时间是完成该任务所需时间与完成本标准的测试要求的时间之较长者。

5.7.3.4 全部预期结果、测试结果及测试数据应存档保留。

5.7.3.5 建立独立的测试小组进行确认测试。

5.7.4 完成标志

所指定的文件要齐全,可验证。应交付的文件有:

- 确认测试分析报告;
- 最终的用户手册和操作手册;
- 项目开发总结报告。

5.8 使用和维护

5.8.1 任务

对投入运行后的软件系统进行修改,以改正在开发阶段产生、在测试阶段又未发现的错误,使软件系统能适应外界环境的改变,并实现软件系统的功能扩充和性能改善。

5.8.2 实施步骤

5.8.2.1 软件系统的用户或维护人员根据软件系统出现的错误、产生的问题、或情况的变化向维护管理人员提交“软件问题报告”,然后由维护人员向维护管理人员提交“软件修改报告”。

5.8.2.2 由维护管理人员对“软件修改报告”进行评审,并赋予优先级。

5.8.2.3 维护人员分析维护需求,对解决该修改所需的时间与资源进行估计。

5.8.2.4 按照一定的步骤对程序进行修改或扩充。

5.8.2.5 重新测试被修改过的程序。

5.8.2.6 修改所有有关的文件。

5.8.2.7 通知用户修改已完成,并将修改以后的版本提交给用户。

5.8.2.8 评审

5.8.3 实施要求

5.8.3.1 软件维护必须在严格的管理控制和执行, 避免错上加错的情况出现。

5.8.3.2 尽量避免出现修改的副作用, 在修改前应权衡利弊, 全面考虑。

5.8.3.3 在有效的管理控制下, 有步骤地进行修改。修改后的软件须通过测试, 填写“软件修改报告”(包括错误类型、修改策略、修改状态和修改情况)。

5.8.4 交付文件

- 软件问题报告 (见附录)。
- 软件修改报告 (见附录)。

附录:

1 软件问题报告

软件问题报告		登记号: ④									
		登记日期: ⑤									
		时 间: ⑥									
① 单元 测试	<input type="checkbox"/> 组装 测试	<input type="checkbox"/> 确认 测试	<input type="checkbox"/> 运行 维护	状态 ②	1	2	3	4	5	6	7
报告人 姓名 ⑦		地址 ⑧		电话 ⑨							
③ 问题: 程序 <input type="checkbox"/> 数据库 <input type="checkbox"/> 文件 <input type="checkbox"/>											
子程序/子系统: ⑩		修订版本号: ⑪		磁带: ⑫							
数据库: ⑬		文件: ⑭									
测试用例: ⑮		硬件: ⑯									
问题描述/影响:											
⑰											
附注											
⑱											

“软件问题报告”(见29页上的表)中各项含义解释

A 登记号

由软件配置控制部门为该报告规定一个唯一的、顺序的编号。

B 登记日期

软件配置控制部门登记该报告的日期。

C 问题发现日期

发现该问题的日期和时间

D 活动

在哪个阶段发现的问题, 分为单元测试、组装测试、确认测试和运行维护四个阶段。

E 状态

这是在软件配置记录中维护的动态指示。它在“软件问题报告”中的作用是为了让开发者和测试者追溯和向软件配置控制部门报告最新的状态。状态编码如下:

1——“软件问题报告”正被复查, 以确定采用什么行动。

2——“软件问题报告”已由指定的开发人员去进行处理。

3——修改已做完、测试好, 正准备交给主程序库。

4——主程序库已经更新, 主程序库修改的重新测试尚未完成。

5——测试重做, 问题仍存在。

6——测试重做, 所做的修改无故障, “软件问题报告”被关闭。

7——留待以后关闭, 因问题不是可重复产生的, 或者是属于产品改善方面的, 或者只具有很低的优先级等等。

F 报告人

“软件问题报告”填写者的姓名、地址、电话。

G 问题属于哪一方面

区分是程序的问题还是子程序的问题, 或是数据库的问题, 文件的问题。也可能是他们的某种组合。

H 子程序/子系统

出现问题的子程序的名字。如果不知是哪个子程序, 可标出子系统名。总之, 尽可能给出细节。

I 修订版本号

出现问题的子程序版本。

J 磁带

包含有问题的子程序的主程序库磁带的标识符。

K 数据库

当发现问题时所使用的数据库标识符。

L 文件号

有错误的文件编号

M 测试用例

发现问题时所使用的计算机系统的标识。

软件修改报告		登记号: ①	
		登记日期: ②	
		时间: ③	
报告人 ④		子系统名⑤ 子程序名⑥	
响应哪些软件问题报告⑦			
⑧ 修改: 程序 <input type="checkbox"/> 文件 <input type="checkbox"/> 数据库 <input type="checkbox"/> 解释 <input type="checkbox"/>			
修改描述 ⑨ 批准人 ⑩			
修改:			
语句类型: ⑪ I/O <input type="checkbox"/> 计算 <input type="checkbox"/> 逻辑 <input type="checkbox"/> 数据处理 <input type="checkbox"/>			
程序名: ⑫		老版本号: ⑬ 新版本号: ⑭	
数据库 ⑮ 数据库修改报告P		文件: ⑯ 文件更新: ⑰	
修改已测试否?			
成功否? ⑱ 单元		子系统	组装 确认 运行
“软件问题报告”的问题叙述准确否 ⑲ 是 <input type="checkbox"/> 否 <input type="checkbox"/>			
附注: ⑳ 			
问题来自: ㉑ 软件需求说明书 <input type="checkbox"/> 设计说明书 <input type="checkbox"/> 数据库 <input type="checkbox"/> 程序 <input type="checkbox"/>			
资源估计: ㉒ 人时数: 计算机时间:			

B 登记日期

P 数据库修改报告

数据库修改申请号。

Q文件

如果也要求文件修改,这里给出它的名字。

R文件更新

文件更新通知单编号。

S修改是否已测试

指出已对修改做了哪些测试。如单元、子系统、
组装、确认和运行测试等,并注明测试成功与否。

T“软件问题报告”给出问题的准确叙述与否,
回答是或否。

U问题注释

准确地重新叙述要维护的问题。

V问题源

指明问题来自于哪里,如软件需求说明书、设计说明书、数据库、源程序等。

W资源

完成修改所需的资源估计,即总的人一时数和计算机时间的开销。

附加说明:

本标准由电子工业部提供。

本标准由北京信息工程学院、
上海软件开发中心负责起草。

本标准主要起草人:贾耀良、朱三元、
庞克琪、陈敏、
段小航

(上接40页)策支持模型可被用于各种水平的决策
制定过程,以在保持联系的同时提供特需的经常的
帮助。

例如, D S S模型可以通过与具有战术与战略
计划水平的计划和决策相连接,还可发展出有效的
作业计划表。

结 论

成组技术可以有组织地简化零件和工艺,以建

立有分向性的产品生产单元。使用这种组织和增加的
单元工人在工艺过程工作的家族性,可以提供更
佳的机会使单元班组更有效地控制和管理单元生产。

还有许多可以说明的组织性例子,以及计算机
——人类内接面的改进和决策支持。但是,其结果
是工作单元自主这种方法比其它方法都更为简便,
使制造系统反应迅速,因而更有前途。

(上接封四)

作好CAPP开发的前期工作	杨文魁 何天雷 陈士青 王禄根 黄河 (5.43)
CAPP系统在生产中的实际应用	周正华 (5.64)
自动工艺设计系统GJ—CAPP系统	余慧芬 高伟强 崔炳扶 黄凯峰 (6.43)
工艺标准化和成组工艺	张立兵 (6.52)
美国OIR公司的成组技术用户	潘泽琪 彭均中 (6.54)

成 组 夹 具

试论我国成组夹具发展方向	赵学良 赵波 (5.22)
成组铣槽夹具	黎自芳 (6.41)

FMS

柔性制造系统的成本计算	(美) James A. Hendricks 张胜文 译 李祖庚 校 (5.30)
FMS实验模型NHFMS—1中数据通信的实现	付晓辉 王宁生 (5.33)

CIMS

美国国家标准局自动化制造研究实验基地概况	吴锡英 (3.49)
----------------------------	------------

计 算 机 介 绍

适用于开发GT软件的微型机计算机“大苹果”——Macintosh	胡德明 (5.59)
----------------------------------------	------------

生 产 单 元

成组生产系统空间布置的优化	张传忠 (6.32)
利用成组技术来改进生产单元中工人之间的联系与协作	(美) Donald O. Knight, Michael

L. Wall 著 刘树萍 译 李祖庚 校 (6.38)

标 准

中华人民共和国国家标准——软件开发规范	(6.57)
---------------------------	--------