

第 7 部分：软件开发过程规范

Part 7: The software development process specification

(征求意见稿 V1.0)

2013 - xx-xx 发布

20xx - xx-xx 实施

目 次

前 言.....	II
1. 范围.....	1
2. 规范性引用文件.....	1
3. 术语和定义.....	2
4. 总体要求.....	6
5. 项目规划.....	6
6. 需求分析.....	7
7. 概要设计.....	9
8. 详细设计.....	11
9. 软件编码.....	12
9.1 编码.....	12
9.2 单元测试.....	14
10. 测试.....	16
10.1 集成测试.....	16
10.2 系统测试.....	19
10.3 验收测试.....	22
11. 项目配置管理.....	24
11.1 管理原则.....	24
11.2 接口控制.....	24
11.2.1 文档及介质管理办法.....	24
11.2.2 文档变更和发布.....	24
11.2.3 代码变更和发布.....	25
11.3 配置管理环境.....	25

前 言

《国家食品药品监管信息化标准》发布以下几个部分：

- 第 1 部分：标准化指南；
- 第 2 部分：基本术语；
- 第 3 部分：标准编写规则；
- 第 4 部分：信息分类与代码；
- 第 5 部分：数据元与代码集；
- 第 6 部分：数据集元数据；
- 第 7 部分：软件开发过程规范；
- 第 8 部分：应用支撑平台通用技术规范；
- 第 9 部分：数据共享与交换接口；
- 第 10 部分：数据库设计规范；
- 第 11 部分：网络及通信；
- 第 12 部分：信息安全；
- 第 13 部分：工程管理。

本标准第 7 部分。

本标准依据 GB/T1.1—2009 给出的规则起草。

本标准由国家食品药品监督管理总局信息中心提出。

本标准由国家食品药品监督管理总局信息办归口。

本标准起草单位：国家食品药品监督管理总局信息中心、广东省食品药品监督管理局、中科软科技股份有限公司。

第 7 部分：软件开发过程规范

1. 范围

本规范规定了国家食品药品监管信息化在应用软件开发中的规划、需求、设计、开发、测试等阶段，各应用系统承建商应实施和执行的相关要求。

本规范适用于国家食品药品监管信息化的规划、需求、设计、开发、集成、测试等过程。

2. 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 8566-2007 信息技术 软件生存周期过程

GB/T 8567-2006 计算机软件文档编制规范

GB/T 9385-2008 计算机软件需求说明编制指南

GB/T 9386-2008 计算机软件测试文件编制规范

GB/T 12504-90 计算机软件质量保证计划规范

GB/T 12505-90 计算机软件配置管理计划规范

GB/T 14394-1993 计算机软件可靠性和可维护性管理

GB/T 15532-1995 计算机软件单元测试

GB/T 16260.1-2006 软件工程 产品质量 第 1 部分：质量模型

GB/T 16260.2-2006 软件工程 产品质量 第 2 部分：外部度量

GB/T 16260.3-2006 软件工程 产品质量 第 3 部分：内部度量

GB/T 16260.4-2006 软件工程 产品质量 第 4 部分：使用质量的度量

GB/T 18905.1-2002 软件工程 产品评价 第 1 部分：概述

GB/T 18905.2-2002 软件工程 产品评价 第 2 部分：策划和管理

GB/T 18905.3-2002 软件工程 产品评价 第 3 部分：开发者用的过程

GB/T 18905.4-2002 软件工程 产品评价 第 4 部分：需方用的过程

GB/T 18905.5-2002 软件工程 产品评价 第 5 部分：评价者用的过程

GB/T 18905.6-2002 软件工程 产品评价 第 6 部分：评价模块的文档编制

GB/T 20157-2006 信息技术 软件维护

GB/T 20917-2007 软件工程 软件测量过程

3. 术语和定义

CFDAB/T-0102-2013 收录的部分术语和定义适用于本标准。为了便于使用，以下重复列出了 CFDAB/T-0102-2013 中的部分术语和定义。

3.1

软件开发计划 Software Development Plan (SDP)

一个综合的组装工件，用来收集管理项目时所需的所有信息。它包括先启阶段中开发的许多工件，并且在整个项目过程中保留下来。

[CFDAB/T 0102-2013 定义 3.2.8.10]

3.2

工作分解结构 Work Breakdown Structure (WBS)

把项目可交付成果和项目工作分解成较小的、更易于管理的组成部分的过程。

[CFDAB/T 0102-2013 定义 3.2.8.11]

3.3

配置管理 Configuration Management (CM)

通过技术或行政手段对软件产品及其开发过程和生命周期进行控制、规范的一系列措施。配置管理的目标是记录软件产品的演化过程，确保软件开发者在软件生命周期中各个阶段都能得到精确的产品配置。

[CFDAB/T 0102-2013 定义 3.2.8.12]

3.4

软件配置管理 Software Configuration Management (SCM)

软件配置管理是一种标识、组织和控制修改的技术。软件配置管理应用于整个软件工程过程。SCM 活动的目标就是为了标识变更、控制变更、确保变更正确实现并向其他有关人员报告变更。从某种角度讲，SCM 是一种标识、组织和控制修改的技术，目的是使错误降为最小并最有效地提高生产效率。

[CFDAB/T 0102-2013 定义 3.2.8.13]

3.5

软件配置管理计划 Software Configuration Management Plan (SCMP)

用于策划配置管理活动的计划。

[CFDAB/T 0102-2013 定义 3.2.8.14]

3.6

质量保证 Quality Assurance (QA)

为了提供足够的信任表明实体能够满足质量要求,而在质量体系中实施并根据需要进行证实的全部有计划和有系统的活动。

[CFDAB/T 0102-2013 定义 3.2.8.15]

3.7

质量保证计划 Quality Assurance Plan (QAP)

针对特定的产品、项目或合同规定专门的质量措施、资源和活动顺序的文件。

[CFDAB/T 0102-2013 定义 3.2.8.16]

3.8

过程和产品质量保证和评价 Product & Process Quality Assurance (PPQA)

CMMI 模型进一步强调的是过程和产品质量保证和评价,即 PPQA。虽然对 QA 的独立性放宽了要求,提高了实施 PPQA 的灵活性,但更加强调了 PPQA 功能的客观性。PPQA 人员可以在项目间交叉,但还是不允许项目成员做国家食品药品监管信息化项目的 QA。

[CFDAB/T 0102-2013 定义 3.2.8.17]

3.9

变更控制 Configuration Control (CC)

变更控制是指对通过正式评审后的基线产品的变更进行管理,确保变更有序进行。

[CFDAB/T 0102-2013 定义 3.2.8.18]

3.10

变更控制委员会 Configuration Control Board (CCB)

项目范围变更很可能需要额外的项目资金、额外的资源与时间,因此,应建立包括来自不同领域的项目利益相关者在内的变更控制委员会,以评估范围变更对项目或组织带来的影响。这个委员会应当由具有代表性的人员组成,而且有能力在管理上作出承诺。

[CFDAB/T 0102-2013 定义 3.2.8.19]

3.11

软件需求规格说明书 Software Requirement Specification (SRS)

对所从事的软件产品进行定义,在该文档中详尽说明了这个产品的软件需求,包括范围、特征、功能、运行环境、接口等内容。

[CFDAB/T 0102-2013 定义 3.2.8.20]

3.12

概要设计 Software Preliminary Design (SPD)

概要设计的主要任务是需求分析得到的 DFD 转换为软件结构和数据结构。设计软件结构的具体任务是：将一个复杂系统按功能进行模块划分、建立模块的层次结构及调用关系、确定模块间的接口及人机界面等。数据结构设计包括数据特征的描述、确定数据的结构特性、以及数据库的设计。

[CFDAB/T 0102-2013 定义 3.2.8.21]

3.13

详细设计 Software Detailed Design (SDD)

详细设计的主要任务是设计每个模块的实现算法、所需的局部数据结构。详细设计的目标有两个：实现模块功能的算法要逻辑上正确和算法描述要简明易懂。

[CFDAB/T 0102-2013 定义 3.2.8.22]

3.14

软件编码 Software Code (SC)

软件编码是将上一阶段的详细设计得到的处理过程的描述转换为基于某种计算机语言的程序，即源代码。需注意根据项目的应用领域选择适当的编程语言、编程的软硬件环境以及编码的程序设计风格等事项。

[CFDAB/T 0102-2013 定义 3.2.8.23]

3.15

系统测试 Software System Testing (SST)

系统测试是将已经确认的软件、计算机硬件、外设、网络等其他元素结合在一起，进行信息系统的各种组装测试和确认测试，其目的是通过与系统的需求相比较，发现所开发的系统与用户需求不符或矛盾的地方，从而提出更加完善的方案。

[CFDAB/T 0102-2013 定义 3.2.8.24]

3.16

角色与职责 Roles and responsibilities

软件开发过程中各个阶段主要参与的人员或机构为该过程的角色，参与角色在该过程中的主要工作职责为该角色的职责。

[CFDAB/T 0102-2013 定义 3.2.8.25]

3.17

主要输入 The main input

软件开发过程中各个阶段开始所需要的输入项目，为主要输入。

[CFDAB/T 0102-2013 定义 3.2.8.26]

3. 18

入口准则 Entrance criteria

软件开发过程中各个阶段开始的条件，即为入口准则。

[CFDAB/T 0102-2013 定义 3.2.8.27]

3. 19

活动 Activities

在软件开发过程中各个阶段的主要工作，即为活动。

[CFDAB/T 0102-2013 定义 3.2.8.28]

3. 20

主要输出 The main output

在软件开发过程中各个阶段主要输出的各类文档产品，即为主要输出产品，简称主要输出。

[CFDAB/T 0102-2013 定义 3.2.8.29]

3. 21

软件基线 The software baseline

基线是软件文档或源码（或其它产出物）的一个稳定版本，它是进一步开发的基础。

[CFDAB/T 0102-2013 定义 3.2.8.30]

3. 22

基线产品 Baseline products

在软件开发过程各阶段形成软件基线的提交物，是项目内部发布的提交物。

[CFDAB/T 0102-2013 定义 3.2.8.31]

3. 23

需控制的非基线产品 Non baseline product control

在软件开发过程中各个阶段除了提交基线产品外，还需要提交一些其它的需要控制的提交物，即需控制的非基线产品。

[CFDAB/T 0102-2013 定义 3.2.8.32]

3. 24

其它工作产品 Other work products

在软件开发过程中各个阶段除了需要提交一系列的控制产品以外，还需要管理一些其它的过程性文档，例如项目会议记录等，这些文档即其它工作产品。

[CFDAB/T 0102-2013 定义 3.2.8.33]

出口准则 Exit criteria

在软件开发过程中，各个阶段所需要完成的主要工作，即为出口准则。

[CFDAB/T 0102-2013 定义 3.2.8.34]

约束 Constraint

在软件开发过程中，各个阶段所需要完成的考量标准，即为该阶段的约束。

[CFDAB/T 0102-2013 定义 3.2.8.35]

4. 总体要求

本规范主要描述项目规划、需求分析、概要设计、详细设计、软件编码、测试、项目配置管理几个阶段软件开发的规范。

各阶段主要从编写目的、执行的角色与职责、阶段输入信息、阶段执行的入口准则、主要活动、主要输出产品（过程文档或交付物）、出口准则以及约束等方面进行描述。

5. 项目规划

目的	项目规划是项目开发的初始阶段，目的是为开发过程和过程管理做好必要的准备。项目规划的主要工作是进行估计和制定管理项目的计划。
角色与职责	<ul style="list-style-type: none"> ● 项目经理：与项目相关组商谈项目计划活动，制定项目计划（主体计划）、质量管理计划、量化管理计划和项目定义过程；依据项目计划分配人员和资源；组织和参与本阶段所有评审（同行评审和管理评审）；度量项目数据； ● PPQA 人员：制定质量保证计划；对项目计划过程、项目定义过程、评审过程的执行进行监督，对项目策划过程中的工作产品进行检查； ● SCM 人员：负责制定配置管理计划；建立配置管理环境；对本阶段的产品进行配置管理； ● CCB：批准基线产品入库，对基线产品的变更进行评审和审批； ● 测试人员：负责制定测试计划； ● 业主方：提供客户需求； ● 项目组主要成员：参与项目计划制定和评审； ● 领域专家：对项目可行性分析和研究提供指导。

主要输入	<ul style="list-style-type: none"> ● 工程初步设计； ● 投标技术文件、合同书； ● 初步客户需求/需要。 	
入口准则	<ul style="list-style-type: none"> ● 初步客户需求/需要已被批准； ● 项目合同已被批准。 	
活动	<ul style="list-style-type: none"> ● 构建 WBS 及估计项目的规模、工作量和成本等 (project)； ● 标识和分析风险； ● 编制项目计划（包括软件开发计划、配置管理计划、质量保证计划）； ● 评审和批准项目计划（总体）。 	
主要输出	基线产品	<ul style="list-style-type: none"> ● 软件开发计划； ● 配置管理计划； ● 质量保证计划。
	需控制的非基线产品	<ul style="list-style-type: none"> ● WBS 评估记录； ● 风险分析表和风险评估报告。
	其它工作产品	<ul style="list-style-type: none"> ● 项目计划的评审记录和报告； ● PPQA 检查报告； ● 配置管理状态报告； ● 项目会议记录； ● 里程碑评审记录和报告； ● 项目管理评审记录和报告。
出口准则	<ul style="list-style-type: none"> ● 受影响的组和个人参加了软件开发计划的评审，项目约定和计划并得到受影响的组 and 个人的认可； ● 质量管理计划得到质量经理的批准； ● 本阶段所有输出工作产品已置于配置管理之下。 	
约束	项目相关组代表应参与项目计划和项目定义过程的评审。	

6. 需求分析

目的	需求分析阶段的主要目的是生成一个正确说明客户所有需求的文档, 需求分析的主要工作是需求提炼及分析、需求归档和需求评审等。
----	--

角色与职责	<ul style="list-style-type: none"> ● 项目经理：为需求分析工作提供各种必要的环境和条件；和业主方、总集成方一起确定项目的范围，完成和检查验收标准；对项目计划进行维护跟踪和监控；组织和参与本阶段所有评审（同行评审和管理评审）；度量项目数据；风险管理与跟踪； ● 需求分析员：作为作者参加需求评审；收集和分析客户需求，定义《软件需求规格说明书》，建立需求库和需求跟踪矩阵； ● PPQA 人员：参加需求评审；对需求析分过程、需求评审过程的执行进行监督；对需求分析阶段产生的工作产品进行检查； ● SCM 人员：参加需求评审；对本阶段的产品进行配置管理； ● CCB：参加需求评审；批准基线产品入库，对基线产品的变更进行评审和审批； ● 测试人员：参加需求评审；细化测试计划，编写系统测试用例； ● 业主方和总集成方代表：提供业主方需求，对需求进行确认； ● 领域专家：参加需求评审；对项目需求分析提供指导。 	
主要输入	<ul style="list-style-type: none"> ● 工程初步设计； ● 投标技术文件、合同书； ● 初步客户需求/需要； ● 总集成方定义的标准。 	
入口准则	<ul style="list-style-type: none"> ● 合同已签定； ● 初步客户需求/需要客户已确认。 	
活动	<ul style="list-style-type: none"> ● 采集详细客户需求/需要； ● 分析客户需求，编写《软件需求规格说明书》SRS； ● 建立需求库和需求跟踪矩阵； ● 编写测试计划、测试大纲、编写系统测试用例； ● 评审 SRS、系统测试计划和系统测试用例、需求跟踪矩阵； ● 维护和跟踪项目计划； ● 需求分析里程碑评审。 	
主要输出	基线产品	<ul style="list-style-type: none"> ● 《软件需求规格说明书》； ● 系统测试计划、测试大纲、测试用例； ● 软件开发计划、配置管理计划、质量保证计划。

	需 控 制 的 非 基 线 产 品	<ul style="list-style-type: none"> ● 需求跟踪矩阵。
	其 它 工 作 产 品	<ul style="list-style-type: none"> ● SRS/系统测试计划、测试用例/需求跟踪矩阵的同行评审记录和报告； ● PPQA 检查报告； ● 配置管理状态报告； ● 项目会议记录； ● 里程碑评审记录和报告； ● 项目管理评审记录和报告。
出口准则	<ul style="list-style-type: none"> ● 已建立了原始需求与需求规格说明之间以及需求规格说明与已编写的系统测试用例之间的映射关系（需求跟踪矩阵）； ● 每一个软件需求都可编写出相应的测试用例； ● 已编写了 25-30%的系统测试用例并通过了评审； ● 100%修复需求阶段发现的缺陷； ● 已制订了系统测试计划； ● 业主方、监理方、总集成方、项目经理及项目相关人员参加需求的评审； ● 《软件需求规格说明书》通过评审并得到用户的确认和签字。 	
约束	对 SRS、需求跟踪矩阵、测试用例都要进行技术评审；要求采用 UML2.0 标准进行模型的统一描述。	

7. 概要设计

目的	概要设计阶段是从实现的角度开发针对客户需求的解决方案。在这个阶段给出的是高层的方案，这个方案包含两个主要部分，即应用的功能体系结构和数据库设计，并形成《概要设计说明书》文档。
----	---

角色与职责	<ul style="list-style-type: none"> ● 项目经理：为概要设计工作提供各种必要的环境和条件；对项目计划进行维护、跟踪和监控；组织和参与本阶段所有评审（同行评审和管理评审）；度量项目数据； ● 需求分析员：维护需求库和需求跟踪矩阵； ● 设计人员：对项目进行概要设计； ● PPQA 人员：对概要设计过程、概要设计评审过程的执行进行监督；概要设计阶段产生的工作产品进行检查； ● SCM 人员：对本阶段的产品进行配置管理； ● CCB：批准基线产品入库，对基线产品的变更进行评审和审批； ● 测试人员：细化测试计划，编写集成测试用例； ● 领域专家：对项目概要设计提供指导。 	
主要输入	<ul style="list-style-type: none"> ● 《软件需求规格说明书》； ● 需求跟踪矩阵； ● 系统测试计划和系统测试用例； ● 总集成方定义的标准。 	
入口准则	<ul style="list-style-type: none"> ● SRS 通过评审和批准。 	
活动	<ul style="list-style-type: none"> ● 进行功能设计； ● 开发物理数据库设计； ● 编写概要设计文档； ● 细化测试计划，编写集成测试用例； ● 评审概要设计文档、集成测试计划和集成测试用例。 	
主要输出	基线产品	<ul style="list-style-type: none"> ● 《概要设计说明书》； ● 集成测试用例。
	需控制的非基线产品	<ul style="list-style-type: none"> ● 需求跟踪矩阵； ● 测试计划、测试大纲。
	其它工作产品	<ul style="list-style-type: none"> ● 《概要设计说明书》评审记录和报告； ● PPQA 检查报告； ● 配置管理状态报告； ● 项目会议记录； ● 里程碑评审记录和报告； ● 项目管理评审记录和报告。

出口准则	<ul style="list-style-type: none"> ● 项目相关人员参加概要设计的评审； ● 《概要设计说明书》，集成测试用例通过了评审； ● 更新的需求与概要设计、集成测试的对应关系； ● 100%修复概要设计阶段发现的缺陷； ● 概要设计阶段评审的功能数占需求阶段已确定的功能数的 100%； ● 本阶段所有输出工作产品已置于配置管理之下。
约束	对所有的概要设计文档、需求跟踪矩阵、测试用例都要进行技术评审；要求采用 UML2.0 标准进行模型的统一描述。

8. 详细设计

目的	在详细设计阶段，概要设计阶段开发的整体应用被分成几个模块和程序。为每个程序进行逻辑设计，然后归档作为程序规格。同时，为每个程序生成一个单元测试计划。详细设计阶段的活动包括通用例程和程序的确定（如数据有效性例程、框架程序的开发及用于提高生产率的实用程序和工具的开发）。
角色与职责	<ul style="list-style-type: none"> ● 项目经理：为详细设计工作提供各种必要的环境和条件；对项目计划进行维护、跟踪和监控；组织和参与本阶段所有评审（同行评审和管理评审）；度量项目数据； ● 需求分析员：维护需求库和需求跟踪矩阵； ● 设计人员：对项目进行详细设计； ● PPQA 人员：对详细设计过程、详细设计评审过程的执行进行监督；对本阶段产生的工作产品进行检查； ● SCM 人员：对本阶段的产品进行配置管理； ● CCB：批准基线产品入库，对基线产品的变更进行评审和审批； ● 测试人员：细化测试计划，编写单元测试用例； ● 领域专家：对项目详细设计提供指导。
主要输入	<ul style="list-style-type: none"> ● 《软件需求规格说明书》SRS； ● 需求跟踪矩阵； ● 《概要设计说明书》； ● 系统测试计划、测试大纲和测试用例； ● 集成测试计划和集成测试用例。
入口准则	<ul style="list-style-type: none"> ● 《概要设计说明书》经过评审和授权。

活动	<ul style="list-style-type: none"> ● 编写详细设计文档； ● 评审详细设计文档。 	
主要输出	基线产品	<ul style="list-style-type: none"> ● 《详细设计说明书》（包括数据结构设计）。
	需控制的非基线产品	<ul style="list-style-type: none"> ● 需求跟踪矩阵。
	其它工作产品	<ul style="list-style-type: none"> ● 《详细设计说明书》评审记录和报告； ● PPQA 检查报告； ● 配置管理状态报告； ● 项目会议记录； ● 里程碑评审记录和报告； ● 项目管理评审记录和报告。
出口准则	<ul style="list-style-type: none"> ● 项目相关人员参加详细设计的评审； ● 《详细设计说明书》通过了评审； ● 更新的需求与概要设计、详细设计、集成测试的对应关系； ● 100%修复详细设计阶段发现的缺陷； ● 详细设计阶段评审的功能数占需求阶段已确定的功能数的 100%； ● 本阶段所有输出工作产品已置于配置管理之下。 	
约束	对所有的详细设计文档都要进行技术评审；要求采用 UML2.0 标准进行模型的统一描述。	

9. 软件编码

9.1 编码

目的	在编码阶段，根据详细设计用编程语言和合适的编码规范产生源代码、可执行代码和数据库。这个阶段的输出是随后测试和验证的主体。
----	--

角色与职责	<ul style="list-style-type: none"> ● 项目经理：为编码试工作提供各种必要的环境和条件；对项目计划进行维护、跟踪和监控；组织和参与本阶段所有评审（同行评审和管理评审）；度量项目数据； ● 需求分析员：维护需求库和需求跟踪距阵； ● 开发人员：编码测试；修复代码中发现的缺陷； ● PPQA 人员：对编码测试过程、编码评审过程的执行进行监督；对编码测试阶段产生的工作产品进行检查； ● SCM 人员：对编码测试阶段的产品进行配置管理； ● CCB：批准基线产品入库，对基线产品的变更进行评审和审批； ● 测试人员：编写编码测试计划，进行编码测试； ● 领域专家：对项目编码提供指导。 	
主要输入	<ul style="list-style-type: none"> ● 《详细设计说明书》； ● 总集成方定义的标准； ● 编码审查规范； ● 代码审查表； ● 测试计划。 	
入口准则	<ul style="list-style-type: none"> ● 详细设计文档经过评审和授权。 	
活动	<ul style="list-style-type: none"> ● 编码和编译； ● 执行测试用例； ● 核心代码评审； ● 代码走查； ● 记录和修正缺陷； ● 根据度量指标收集项目数据。 	
主要输出	基线产品	<ul style="list-style-type: none"> ● 源代码、编译的目标代码；
	需控制的非基线产品	<ul style="list-style-type: none"> ● 代码审查表； ● 缺陷跟踪过程检查表。

	其它工作产品	<ul style="list-style-type: none"> ● PPQA 检查报告; ● 配置管理状态报告; ● 项目会议记录; ● 里程碑评审记录和报告; ● 项目管理评审记录和报告。
出口准则		<ul style="list-style-type: none"> ● 所有的编码满足编码审查规范; ● 核心代码应经代码评审, 同时代码评审的比例不低于 20%; ● 更新的需求与概要设计、详细设计、集成测试的对应关系; ● 100%修复编码测试阶段发现的缺陷; ● 本阶段所有输出工作产品已置于配置管理之下。
约束		<ul style="list-style-type: none"> ● 总集成方定义的标准; ● 编码审查规范。

9.2 单元测试

目的	单元测试的目的是检查每个软件单元能否正确地实现设计说明中的功能、性能、接口和其他设计约束等要求, 发现单元内可能存在的各种差错。
角色与职责	<ul style="list-style-type: none"> ● 项目经理: 为本阶段工作提供各种必要的环境和条件; 对项目计划进行维护、跟踪和监控; 组织和参与本阶段所有评审 (同行评审和管理评审); 度量项目数据; ● 需求分析员: 维护需求库和需求跟踪矩阵; ● 开发人员: 修复发现的代码中的缺陷; ● PPQA 人员: 对单元测试过程、单元测试评审过程的执行进行监督; 对本阶段产生的工作产品进行检查; ● SCM 人员: 对本阶段的产品进行配置管理; ● CCB: 批准基线产品入库, 对基线产品的变更进行评审和审批; ● 测试人员: 编写单元测试用例, 进行单元测试; ● 领域专家: 对项目单元测试提供指导。

技术要求	<ul style="list-style-type: none"> ● 确定测试充分性要求。根据软件单元的重要性、软件单元测试目标和约束条件，确定测试应覆盖的范围及每一范围所要求的覆盖程度（如，分支覆盖率、语句覆盖率、功能覆盖率、单元的每一软件特性应至少被一个正常的测试用例和一个异常的测试用例所覆盖）； ● 确定测试终止的要求。指定测试过程正常终止的条件（如，测试充分性是否达到要求），确定导致测试过程异常终止的可能情况（如软件编码错误）； ● 确定用于测试的资源要求。包括软件（如操作系统、编译软件、静态分析软件、测试数据产生软件、测试结果获取和处理软件、测试驱动软件等）、硬件（如计算机、设备接口等）、人员数量、人员技能等； ● 确定需要测试的软件特性。根据软件设计文档的描述确定软件单元的功能、性能、状态、接口、数据结构、设计约束等内容和要求，并对其标识。若需要，将其分类。并从中确定需要测试的软件特性； ● 确定测试需要的技术和方法，如，测试数据生成与验证技术、测试数据输入技术、测试结果获取技术； ● 根据项目计划的要求和被测软件的特点，确定测试准出条件； ● 确定由资源和被测软件单元所决定的单元测试活动的进度； ● 对测试工作进行风险分析与评估，并制定应对措施； ● 根据上述分析研究结果，按照 GB/T 9386 的要求编写软件单元测试计划； ● 应对软件单元测试计划进行评审。评审测试的范围和内容、资源、进度、各方责任等是否明确，测试方法是否合理、有效和可行，风险的分析、评估与对策是否准确可行，测试文档是否符合规范，测试活动是否独立。由软件的供方自行组织评审。评审细则也自行制定。在软件单元测试计划通过评审后，进入下一步工作；否则，需要重新进行单元测试的策划。
主要输入	<ul style="list-style-type: none"> ● 源程序、编译后的目标程序； ● 应用系统测试规范； ● 测试计划。
入口准则	<ul style="list-style-type: none"> ● 要测试代码经过走查或是评审； ● 单元测试计划及单元测试用例经过评审。
活动	<ul style="list-style-type: none"> ● 更新需求跟踪矩阵； ● 执行单元测试用例； ● 单元测试报告； ● 缺陷修正记录； ● 编写产品使用手册。

主要输出	基线产品	<ul style="list-style-type: none"> ● 编码测试后的源代码、编译的目标代码。
	需控制的非基线产品	<ul style="list-style-type: none"> ● 需求跟踪矩阵； ● 测试计划——单元测试计划； ● 单元测试报告； ● 产品使用手册。
	其它工作产品	<ul style="list-style-type: none"> ● 单元测试的评审记录和报告； ● PPQA 检查报告； ● 配置管理状态报告； ● 项目会议记录； ● 里程碑评审记录和报告； ● 项目管理评审记录和报告。
出口准则	<ul style="list-style-type: none"> ● 完成软件集成，并成功地执行了单元测试计划中的所有测试用例； ● 被测对象模块功能覆盖率 100%； ● 更新的需求与概要设计、详细设计、集成测试的对应关系； ● 100%修复单元测试阶段发现的二级以上缺陷，并通过回归测试的验证； ● 本阶段所有输出工作产品已置于配置管理之下。 	
约束	对单元测试用例和报告进行技术评审。	

10. 测试

10.1 集成测试

目的	<p>软件集成是把设计阶段制定的，已通过单元测试的模块（或应用系统）构建成一个完整的软件结构的系统方法。在该阶段，同时要进行集成测试，以发现和接口相关的缺陷。集成按集成计划中制定的顺序进行，并执行每个集成阶段的相应测试用例。</p>
----	--

角色与职责	<ul style="list-style-type: none"> ● 项目经理: 为本阶段工作提供各种必要的环境和条件; 对项目计划进行维护、跟踪和监控; 组织和参与本阶段所有评审 (同行评审和管理评审); 度量项目数据; ● 需求分析员: 维护需求库和需求跟踪矩阵; ● 开发人员: 修复发现的代码中的缺陷; ● PPQA 人员: 对集成测试过程、集成测试评审过程的执行进行监督; 对本阶段产生的工作产品进行检查; ● SCM 人员: 对本阶段的产品进行配置管理; ● CCB: 批准基线产品入库, 对基线产品的变更进行评审和审批; ● 测试人员: 编写集成测试用例, 进行集成测试; ● 总集成方: 对各应用系统集成提出要求, 并指导完成系统集成; 统一安排各应用系统开发商进行集成测试。
技术要求	<ul style="list-style-type: none"> ● 应对已集成软件进行必要的静态测试, 并先于动态测试进行; ● 软件要求的每个特性应被至少一个正常的测试用例和一个被认可的异常测试用例覆盖; ● 测试用例的输入应至少包括有效等价类值无效等价类值和边界数据值; ● 应采用增量法, 测试新组装的软件; ● 应逐项测试软件设计文档规定的软件功能、性能等特性; ● 应测试软件之间、软件和硬件之间的所有接口; ● 应测试软件单元之间的所有调用, 达到 100% 的测试覆盖率; ● 应测试软件的输出数据及其格式; ● 应测试运行条件 (如数据结构、输入/输出通道容量、内存空间、调用频率等) 在边界状态下, 进而在人为设定的状态下, 软件的功能和性能; ● 应按设计文档要求, 对软件的功能、性能进行强度测试; ● 对完整性级别高的软件, 应对其进行安全性分析, 明确每一个危险状态和导致危险的可能原因, 并对此进行针对性的测试。

主要输入	<ul style="list-style-type: none"> ● 《软件需求规格说明书》; ● 《概要设计说明书》; ● 《详细设计说明书》; ● 用户使用手册; ● 总集成方定义的标准; ● 应用系统测试规范; ● 集成测试计划; ● 单元测试后的源代码、编译的目标代码。 	
入口准则	<ul style="list-style-type: none"> ● 被集成的模块通过了单元测试; ● 集成测试计划和测试用例通过评审。 	
活动	<ul style="list-style-type: none"> ● 确定集成环境和集成规程; ● 进行集成和集成测试; ● 集成测试报告; ● 根据度量指标收集项目数据。 	
主要输出	基线产品	<ul style="list-style-type: none"> ● 集成测试后的完整软件产品（源代码、目标代码）。
	需控制的非基线产品	<ul style="list-style-type: none"> ● 需求跟踪矩阵; ● 集成测试计划; ● 集成测试报告; ● 用户使用手册。
	其它工作产品	<ul style="list-style-type: none"> ● 集成测试的评审记录和报告; ● PPQA 检查报告; ● 配置管理状态报告; ● 项目会议记录; ● 里程碑评审记录和报告; ● 项目管理评审记录和报告。

出口准则	<ul style="list-style-type: none"> ● 完成软件集成，并成功地执行了集成测试计划中的所有测试用例； ● 被测对象模块功能覆盖率 100%； ● 接口功能覆盖率 100%； ● 通过回归测试； ● 通过压力测试； ● 通过稳定性测试； ● 更新的需求与概要设计、详细设计、单元测试、集成测试的对应关系； ● 100%修复集成测试阶段发现的二级以上缺陷； ● 本阶段所有输出工作产品已置于配置管理之下。
约束	对集成测试用例和报告进行技术评审。

10.2 系统测试

目的	系统测试是依据《软件需求规格说明书》验证软件产品有效性的活动。这个阶段是为了发现那些只有通过测试整个系统才能暴露的缺陷，像外部接口、性能、安全和可靠性等只有在这个阶段才能判断其是否有效。
角色与职责	<ul style="list-style-type: none"> ● 项目经理：为本阶段工作提供各种必要的环境和条件；对项目计划进行维护、跟踪和监控；组织和参与本阶段所有评审（同行评审和管理评审）；度量项目数据； ● 需求分析员：维护需求库和需求跟踪矩阵； ● 开发人员：修复发现的代码中的缺陷； ● PPQA 人员：对系统测试过程、系统测试评审过程的执行进行监督；对本阶段产生的工作产品进行检查； ● SCM 人员：对本阶段的产品进行配置管理； ● CCB：批准基线产品入库，对基线产品的变更进行评审和审批； ● 测试人员：编写系统测试用例，进行系统测试； ● 业主方和总集成方代表：参与系统测试。

技术要求	<ul style="list-style-type: none"> ● 系统的每个特性应至少被一个正常测试用例和一个被认可的异常测试用例所覆盖； ● 测试用例的输入应至少包括有效等价类值、无效等价类值和边界数据值； ● 应逐项测试系统/子系统设计说明规定的系统的功能、性能等特性； ● 应测试软件配置项之间及软件配置项与硬件之间的接口； ● 应测试系统的输出及其格式； ● 应测试运行条件在边界状态和异常状态下，或在人为设定的状态下，系统的功能和性能； ● 应测试系统访问和数据安全性； ● 应测试系统的全部存储量、输入/输出通道和处理时间的余量； ● 应按系统或子系统设计文档的要求，对系统的功能、性能进行强度测试； ● 应测试设计中用于提高系统安全性、可靠性的结构、算法、容错、冗余、中断处理等方案； ● 应对修复的测试问题进行回归测试； ● 对完整性级别高的系统，应对其进行安全性、可靠性分析，明确每一个危险状态和导致危险的可能原因，并对此进行针对性的测试； ● 对有恢复或重置功能需求的系统，应测试其恢复或重置功能和平均恢复时间，并且对每一类导致恢复或重置的情况进行测试； ● 对不同的实际问题应外加相应的专门测试。
主要输入	<ul style="list-style-type: none"> ● 《软件需求规格说明书》； ● 《概要设计说明书》； ● 《详细设计说明书》； ● 用户使用手册； ● 源程序、可执行程序； ● 应用系统测试规范； ● 测试计划、测试大纲、测试用例； ● 集成测试后的源代码、编译的目标代码。
入口准则	<ul style="list-style-type: none"> ● 软件产品通过了集成测试； ● 系统测试计划和测试用例通过评审。

活动	<ul style="list-style-type: none"> ● 确定系统测试环境和测试规程； ● 进行系统测试； ● 系统测试报告； ● 评审用户使用手册； ● 根据度量指标收集项目数据。 	
主要输出	基线产品	<ul style="list-style-type: none"> ● 用户使用手册（包括安装手册）； ● 集成测试后的完整软件产品（源代码、目标代码）。
	需控制的非基线产品	<ul style="list-style-type: none"> ● 需求跟踪矩阵； ● 系统测试报告。
	其它工作产品	<ul style="list-style-type: none"> ● 系统测试的评审记录和报告； ● PPQA 检查报告； ● 配置管理状态报告； ● 项目会议记录； ● 里程碑评审记录和报告； ● 项目管理评审记录和报告。
出口准则	<ul style="list-style-type: none"> ● 成功地执行了系统测试计划中的所有测试用例，完成系统测试的软件产品、测试驱动程序、系统测试报告； ● 被测对象要求功能覆盖率 100%； ● 通过回归测试； ● 通过压力测试； ● 通过稳定性测试； ● 通过破坏测试； ● 更新了（需求跟踪矩阵）中需求与详细设计、单元测试、集成测试、系统测试的对应关系； ● 最终二级以上缺陷为零； ● 本阶段所有输出工作产品已置于配置管理之下。 	
约束	对系统测试报告进行技术评审。	

目的	在验收和安装阶段，软件产品被集成到它的生产环境中，并在这个环境中接受测试，以确保它按需求执行。这个阶段包括两个基本任务：使软件得以验收和在客户处安装。验收指的是用户根据早期准备的验收测试计划而进行正式的测试，并对测试结果进行分析，以确定系统是否满足验收准则。当分析结果满足验收准则时，用户接受软件。安装指的是把接受的软件置于实际的生产环境中。	
角色与职责	<ul style="list-style-type: none"> ● 项目经理：为本阶段工作提供各种必要的环境和条件；对项目计划进行维护、跟踪和监控；组织和参与本阶段所有评审（同行评审和管理评审）；度量项目数据； ● PPQA 人员：对验收和发布过程、验收和发布评审过程的执行进行监督；对验收和发布阶段产生的工作产品进行检查； ● SCM 人员：对本阶段的产品进行配置管理； ● CCB：批准发布的产品； ● 测试人员：参与验收测试； ● 业主方、监理方和总集成方代表：参与验收测试。 	
技术要求	<ul style="list-style-type: none"> ● 《软件需求分析说明书》中定义的所有功能已全部实现，性能指标全部达到要求； ● 在验收测试中发现的错误已经得到修改，各级缺陷修复率达到标准； ● 所有测试项没有残余一级、二级、三级和四级错误； ● 需求分析文档、设计文档和编码实现一致； ● 验收测试工件齐全（测试计划、测试用例、测试日志、测试通知单、测试分析报告，待验收的软件安装程序）。 	
主要输入	<ul style="list-style-type: none"> ● 应用系统测试规范； ● 测试计划。 	
入口准则	<ul style="list-style-type: none"> ● 软件产品通过了系统测试。 	
活动	<ul style="list-style-type: none"> ● 按计划执行验收； ● 执行安装； ● 根据度量指标收集项目数据。 	
主要输出	基线产品	<ul style="list-style-type: none"> ● 完整软件产品（源代码、目标代码）； ● 安装后的软件。

	需控制的非基线产品	<ul style="list-style-type: none"> ● 验收测试报告。
	其它工作产品	<ul style="list-style-type: none"> ● 验收和发布的评审记录和报告； ● PPQA 检查报告； ● 配置管理状态报告； ● 发布报告； ● 项目会议记录； ● 里程碑评审记录和报告； ● 项目管理评审记录和报告。
出口准则		<ul style="list-style-type: none"> ● 用户验收测试； ● 客户在验收测试报告上签字、软件运行在实际的生产环境中； ● 本阶段的输出的工作产品通过评审和批准并置于配置管理之下。
约束		管理评审。

11. 项目配置管理

11.1 管理原则

- a) 配置管理贯穿项目开发的全过程；
- b) 管理文档资料齐全、查询调阅快捷方便；
- c) 资料安全，不易丢失、泄密；
- d) 内容能反映项目开发生命周期内任一时期的配置情况。

11.2 接口控制

11.2.1 文档及介质管理办法

a) 文档管理方法

1) 文档标识

文档标识中应能体现项目名称、文档类型等；

文档标识应唯一、便于查询。

2) 文档控制

使用控制管理工具对配置管理库进行版本管理。

3) 文档命名规范

文档的命名由项目名称+文档名称组成，不能包括文档版本号，版本号由配置管理工具统一管理。

b) 介质管理办法

项目中介质包括磁盘和纸张。

对于项目中涉及的磁盘介质（包括提交给业主方和从业主方获取）、纸张（各种用户确认的纸制文档、原始调研纸制材料等），需要由项目组配置管理员统一管理，并注册登记。

c) 配置管理员：各应用系统开发商需安排专员担任配置管理员。

11.2.2 文档变更和发布

a) 提出变更

变更申请人填写配置变更申请。

b) 评估、批准变更

项目经理对变更申请进行评估，根据评估结果填写配置变更申请的变更批准。指定责任人完成变更。

c) 实施变更

修改需要变更的文档，经过评审确认，变更实施人填写变更申请表的变更实施部分。CCB、项目经理签字确认变更结束。

d) 发布变更

配置管理员提交、配置已变更的配置项。

11.2.3 代码变更和发布

代码变更的情况有两种：

- a) 设计说明书、软件需求、客户需求需要发生变更，导致已配置代码变更。
- b) 代码版本定义

代码版本分为 ABC 三段，表现形式如图 1 所示：

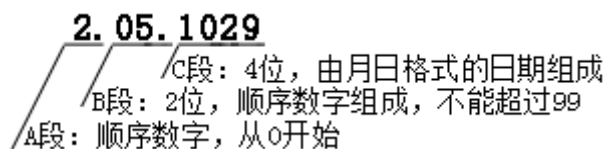


图 1

其中：A 段，为顺序数字，从 0 开始，每次递增 1；

B 段，由 2 位顺序数字组成，数字不足 2 位，前面补 0，每次递增 1，超过 99，A 段递增 1，B 段置 0；

C 段，4 位，为当前日期的月和日的组合，例如 10 月 29 日，则为：1029

- c) 测试中发现问题，需要对代码进行修改。

第一种情况的流程，如下：

- 1) 对整个项目工程做版本标记；
- 2) 实施变更。提交修改代码和日志。填写配置申请表中变更实施部分的内容。CCB、项目经理签字确认变更结束；
- 3) 发布。配置管理员更新代码库，发布新版本。如果涉及功能模块的变更，A 段版本号增加 1，如果只是功能点的变更，B 段版本增加 1。

第二种情况的流程，如下：

- 1) 项目经理分析软件问题报告单，判断是否发生代码变更；
- 2) 实施变更。提交修改代码和日志。填写软件问题报告单的问题修改部分；
- 3) 发布。配置管理员更新代码库，发布新版本，如涉及一级、二级 BUG 修改，B 段版本号增加 1，其它 BUG 则 C 段版本为当前日期。

11.3 配置管理环境

在项目建设之前应提供配置管理服务器，配置管理服务器建议使用成熟的配置管理工具搭建(例如：SVN、VSS、CVS)。