

# 第五章 软件文档写作要求

5.1 系统设计生命周期文档的例子

5.2 软件编档的灵活性

5.3 制定文档编制策略

5.4 制定文档计划

5.5 建立编档规程

5.6 软件编档步骤

## 5.1 系统设计生命周期文档的例子

### 系统设计生命周期文档

---

NetReptile推荐 [2005-4-4]

出处：zdnnet

作者：不详

多数押宝微软 .NET技术的公司通常也会选用微软的生产性工具。只要采用正确的过程，布局和规范，公司就能结合他们的开发环境和生产性工具创建一个可用的系统设计和系统维护结构。让我们先回忆一下书写文档过程，然后再分析一下你的系统设计工具箱里应该有哪些产品和工具。

## 书写文档过程

即使每家公司都赞同他们应该有一个软件开发周期 (SDLC), 但是没有两家公司对每个阶段以及工作产品下的定义完全一样。在创建你自己的软件设计方法以及选择实现工具的时候, 你必须首先定义你的设计过程和书写文档过程。定义的一部分应该包含对每个项目创建和维护的一组文档。最起码, 你应该创建下面的文档 (或者其它包含所有这些部分的文档)。

## 商业需求

商业需求文档应该概括说明开发系统的原因, 预期的系统总成本, 以及实现的收益等。还应该列出预期的使用和性能量度, (例如, 什么类型的用户将使用该系统, 系统的响应时间等)。商业需求文档将用来指导系统设计所有的后继阶段。对该文档所做的任何改动必须反映实际的设计和开发中, 而且在设计和开发之前进行核准。

## 过程图表 (process diagram)和使用案例 (use cases)

一旦定义了商业需求，系统架构师书写文档的内容就转到系统的关键参与者之间信息的流动方式上来。定义参与者和过程是创建使用案例的关键，它根据参与到这个系统的人或系统来定义每个主要的系统功能。过程图表和使用案例文档放在一起将提供连接商业需求和高层技术实现的信息。同样地，文档应该易于理解，易于客户 (或者商业倡导者) 编辑，易于系统架构师创建详细技术文档。

## 建模文档

系统架构师应该使用过程图表和使用案例模型作为源文档来创建类模型和数据库模型。类模型将系统表示为一组定义在使用案例中的，用于实现系统主要功能的逻辑类。数据库模型将提供用来存储瞬间过程状态信息、瞬间系统输出以及历史事务数据的数据库的物理实现。

## 书写系统文档的三个主要工具

当然，你可能永远都不会在初期得到所有的完成得很好的系统需求文档和系统设计文档。选用那些在开发过程中允许你改变设计以及公布这些改变的工具是很重要的。要获取所有用来创建设计环境的工具，你的开发人员至少应该有Microsoft Office Professional以及Microsoft Visual Studio .NET Enterprise Developer。所有系统构架师应该使用相同版本的Office，使用Microsoft Visual Studio .NET Enterprise Architect。这将为系统构架师提供一个使用Microsoft Visio for Enterprise Architects的许可。

对于你的设计环境，关键工具是[Microsoft Word](#)，[Microsoft Visio](#)以及[SharePoint Team Services\(STS\)](#)。Word有一些非常好的团队协作工具。工具菜单中的跟踪变化功能允许多人编辑一个文档，并将其改变记入文档。编辑者(例如商业倡导者、系统构架师或者开发人员等)可以往文档中添加注释，而不是直接改变文档。他们可以向复查人员发送文档的拷贝。之后，他们就可以根据那些注释将所有的变化合并到原始文档中，决定接受那个改变，丢弃那个改变。

Visual Studio .NET Enterprise Architect版本中的Visio的版本包含创建使用案例、类模型和数据库图表的模板。你甚至可以使用类模型产生函数模板，使用数据库图表工具产生SQL Server以及其它的兼容ODBC的数据库的数据库模式。事实上，对于在目前在用的系统来说，你可以反向地从VB代码得出类模型，从数据库得出数据库图表。Visio不像Word一样能跟踪多个用户对文档的改变，但是它却可以得到设计的反馈信息，这一点是很重要的。为了允许在类模型和数据库图表上做注释，我推荐系统架构师将它们的模型和图表做成JPEG文件贴到一个Word文档中。这样你不仅可以打印、查看图表，也可以使用Word的跟踪改变的特性。

一旦创建了文档之后，将它们放到多个用户可以查看并对其注释的地方。在Office2000、Office XP以及Microsoft FrontPage 2000或者更高版本的每个拷贝中都带有一个简单协作服务器。这个工具，即STS，可以用来创建一个基于Web的入口 (portal)，作为关键设计文档的一个仓库。

STS使用Web浏览器或者Word或Visio中打开/保存文件机制提供一个简单机制用于上传和下载文档。此外，入口用户可以将文档提交给站点或者个人，从而接收STS在任何时间侦测到的有关文档的改变的通知。事实上，如果你使用Word或者IE 5.0+查看Word文档，你可以在线做注释、改变，而不用事先下载它们。很多系统构架师宁愿使用Visual SourceSafe存储他们的系统文档，因为它具有登记/检验功能。但是它缺乏通知功能，因此不是一个很理想的选择。

## 重视文档书写

系统构架师最终会决定选用哪套工具来管理SDLC。你可以选用平均合每人5000美元的Rational环境，还可以选用已经有使用许可的工具来创建设计环境，但是不管采用哪种方式，如果在开发过程中不遵从更新文档原则，那么你将不会取得成功。但是有一个书面的强制开发方法和一个好的设计协作环境的话，你就可以按时满足商业需求并在预定的预算内交付应用程序。

## 5.2 软件编档的灵活性

文档的灵活性一般表现在以下几个方面：

### 1. 编制文档的种类

如按照标准，一般软件开发过程需要产生的文档有 14 种之多。但针对一个具体的软件开发项目，可以根据灵活性原则，决定哪些文档需要标准，哪些可以合并，哪些又可以省略。

一般而言，当项目的规模、复杂性和潜在风险增大时，文档编制的范围、数量、管理手续和详细程度都将随之增加；反之，则可适当减少。

对于一个具体的项目，项目负责人应根据实际情况，确定文档编制计划，具体包括：

- (1) 应该编制哪些文档，详细程度如何？
- (2) 各个文档的编制负责人和进度要求。
- (3) 审查、批准的负责人和时间进度要求。
- (4) 开发期间，各文档的维护、修改和管理的负责人，以及批准手续。

## 2. 文档的详细程度

文档的详细程度取决于项目的规模、复杂性和项目负责人对项目开发及运行环境需求情况的判断。当然，还取决于其自身的经验和对文档涉众的了解程度。

## 3. 文档的扩展

对于规模很大的开发项目，如源码超过 100 万行，其文档就需要分卷编制。分卷既可以按子系统，也可以按内容。如：

项目开发计划可能包括以下内容：

- 质量保证计划
- 配置管理计划
- 用户培训计划
- 安装实施计划

系统设计说明书可分写成以下内容：

- 系统设计说明书
- 子系统设计说明书



**程序设计说明书可分写成以下内容：**

程序设计说明书

接口设计说明书

版本说明

**操作手册可分写成以下内容：**

操作手册

安装实施过程

**测试计划可分写成以下内容：**

测试计划

测试设计说明

测试规程

测试用例

**测试分析报告可分写成以下内容：**

综合测试报告

验收测试报告

## 4. 章节的扩张与缩并

标准一般对文档的章、条标题都有描述，在编档时既可以参照使用，也可以根据实际情况予以开展或缩并。

## 5. 程序设计的表现形式

由于标准，特别是国标对程序设计的表现形式没有作出具体的规定或限制，因此，可以使用多种方法、工具和表达形式来进行表现。

## 6. 文档的表现形式

国标对文档的表现形式也没有作出规定和限制，因此，编档时可根据具体涉众的需要，选择自然语言或形式语言表达。

## 7. 文档的其它种类

国标给出的文档种类通常已能满足项目内容的表达，但当项目有特殊要求时，可以创建新的文档种类，以满足使用。

## 5.3 制定文档编制策略

文档策略通常由上级 ( /资深 ) 管理者制定，以规约下级并提供指导。策略通常都只是规定主要的方向，而不作具体的目标、过程描述。

文档策略的陈述要简要、明确，通告全体并理解，进而使策略能被贯彻执行。文档策略表可以在完整性、有效性及对策略的评估等方面提供帮助。

### 文档策略检查表：

- 适当的文档决策作出了吗？
- 涉及文档的策略声明发布了么？
- 负责文档编制的人员或部门落实了么？
- 用于文档编制的资源落实了么？
- 负责文档质量的人员或部门落实了么？
- 各管理层次和组织分支间的关系确定么？其职责、活动和联系的途径？
- 所有文档都已列入整个项目计划了么？
- 适用的文档标准都已确定了么？
- 是否已考虑采用支持工具和自动化文档编制？

## 支持有效文档策略的基本条件有以下这些：

### 1. 文档需要覆盖整个软件生存期

从项目早期文档开始，文档在整个开发过程应是可用、可维护的，在开发完成后，文档应满足软件的使用、维护、升级、移植或传输。

### 2. 文档应是可管理的

指导和控制文档的管理者、发行者，应准备文档产品、进度、可靠性、资源、质量保证和评审规程的详细计划大纲。

### 3. 文档应适合其涉众

文档发行者应针对不同涉众，设计、发行不同类型的文档。

### 4. 文档效应应贯穿软件开发的整个过程

文档应指导软件全部的开发过程。

### 5. 文档标准应被标识和使用

文档应尽可能采用现行标准，否则，也应及时研制适用标准或指南。

### 6. 应规定支持工具

使用编档工具是经济、可行的，也有助于开发和维护。

## 5.4 制定文档计划

文档计划应是整个项目计划的一部分，大型项目甚至是一个独立的子计划。

编制文档计划的工作应尽早开始，并将对计划的评审工作贯穿项目的全过程。文档计划一般包括以下几个方面：

1. 列出应编制文档目录
2. 提示编档应参考的标准
3. 制定文档管理员
4. 提供编档条件，落实编档人员、经费及工具等
5. 明确保证文档质量的方法，应采取的措施等
6. 绘制进度表，明确软件生存期各阶段应产生的文档、编制人员、编制日期、完成日期、评审日期等

此外，文档计划还应规定文档质量等级、确定该计划和文档的分发，并明确相关编档人员的职责等。

下表将为制定文档计划或评估已有文档计划的完整性提供帮助。当然，下表并不总是完整的。

#### 文档计划检查表：

- 文档计划编制好了吗？
- 所需文档类型确定了吗？
- 所需内容是否已列入提纲并描述？
- 文档标准是否已确定？
- 文档标准是否已经制定？
- 有关编档、文档管理员、备用文档存储及文档评审职责分配了吗？
- 质量准则建立了吗？
- 有关提交草稿概要、初稿、修订稿、图示的计划表是否已经建立？
- 评审日期是否确定？
- 一个认可的周期是否确定？
- 制作方法是否已选定并作出计划？

## 例 1: 某国内著名大公司的 ERP 实施规范 (35页)

gotd推荐 [2005-3-16]

出处: internet

作者: internet

——文档类型、标准、涉众等没有确定

——质量准则、文档评审职责等没有确定

——进度计划、提交日期等没有确定

导	言 (Introduction)	3
1	实施步骤 (Implementary Approach)	4
1.1	项目启动 (Project Startup)	4
1.1.1	项目交接	4
1.1.2	项目组织	5
1.2	实施规划 (Layout of Implement)	8
1.2.1	实施方案制订	9
1.2.2	现场调研	9
1.2.3	实施计划制订	11
1.2.4	预算计划制订	12
1.3	教育培训 (Education and Teach)	12
1.3.1	企业中高层培训	13
1.3.2	关键用户培训	13
1.3.3	系统管理员培训	14
1.3.4	最终用户培训	14
1.4	系统初始 (System Commencement)	15

1.4.1	系统安装	15
1.4.2	基础数据准备	15
1.4.3	工作准则拟订	16
1.4.4	系统初始化	17
1.5	系统并行 (System Concurrency)	18
1.5.1	试运行	18
1.5.2	系统并行	19
1.5.3	系统切换	20
1.6	项目结束 (Project end)	20
1.6.1	项目验收	20
1.6.2	售后交接	21
2	项目管理 (Project Management)	21
2.1	范围管理 (Scope Management)	22
2.2	时间管理 (Time Management)	22
2.3	沟通管理 (Communications Management)	23
2.4	风险管理 (Risk Management)	25
2.5	质量管理 (Quality Management)	29
2.6	人力资源管理 (Human Resource Management)	30
2.7	费用管理 (Cost Management)	31
2.8	档案管理 (Records Management)	31
2.9	变更控制 (Change Control)	32
2.10	交接管理 (Connect Management)	34
3	附录 (Appendix)	34



## 例 2: SVC人力资源管理系统详细设计

### 引言

编制《SVC人力资源管理系统详细设计报告》(下称报告),是为了得出对目标系统的精确描述,设计出程序的蓝图,使我公司的程序员根据此蓝图写出代码。

本《报告》的读者是:参与该系统开发的管理人员以及系统设计人员、代码编制人员和系统实施人员。

本《报告》中所涉及的数据流图是描绘系统的逻辑模型,图中没有任何具体的物理元素,只是描绘信息在系统中流动和处理的情况。其各个符号意义说明如下:

**箭头**:表示数据流,即特定数据的流动方向。(如图 1-4)



图 1-1

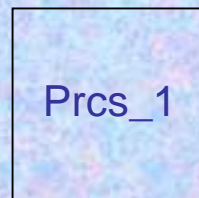


图 1-2

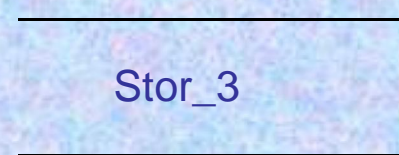


图 1-3



图 1-4

正方形(或立方形):表示数据的源点或终点,即指外部实体。(如图 1-1)

圆形:表示变换数据的处理。(如图 1-2)

两条平行线:表示数据存储。(如图 1-3)

箭头:表示数据流,即特定数据的流动方向。(如图 1-4)

## 5.5 建立编档规程

由于编档工作涉及多人、多阶段、跨时段、跨专业，还需要规定计划、编制、评审、制作和分发等的逻辑顺序，需要设定文档审批、质量保证的控制点，规定文档修改、存储、维护和更新的方法等，因此，建立编档规程是一个好方法。

下表就有可能帮助设计好的规程或评价已有规程：

### 文档管理规程检查表：

- 评审规程是否已建立？
- 软件生存期相关各方的参与合作是否已考虑？
- 一个认可的周期是否已确定？
- 为每个文档或文档类型建立分配表了吗？
- 使文档内容保持最新的方法确定了吗？
- 用于获得用户对文档的建议和反应的反馈机制建立了吗？
- 有关保管和分发的维护规程建立了吗？
- 有关文档的标识和管理的规程建立了吗？
- 保存重要文档的设施建立了吗？

## 5.6 软件编档步骤

一旦项目确定，就可以进入开发阶段，而此时，也是编档工作的开始时间。具体的，软件编档大致可以按照以下步骤：

### 1. 文档计划制定

文档计划的制定应遵照 5.4 节中规定的内容进行。

### 2. 文档编制

软件编档是一项非常细致，甚至是繁琐的工作。从最初提出任务，编写提纲开始，经过逐步充实、完善，并经反复检查、修改，通过评审、审批，直至最终交付使用时止。

软件编档时，应注意以下几点：

编档时间应于软件开发同步，在软件生存期按时完成相应文档的编制工作

严格按照文档计划规定的文档数量和质量要求编制文档

严格按照 GB/T 8567-1988 或企业规定的标准内容和格式编制文档

文档记录介质及尺寸、字体、字号、颜色等规格，应事先确定，统一使用

文档必须集结成册，纸质文档必须装订，并加封面、摘要和目录

归档用的文档还应有扉页，用于相关责任者签署





序号	阶段 文档名称	可行性研究阶段	开发阶段						运行与维护阶段
			需求分析	概要设计	详细设计	实现	测试	试运行	
39	软件问题报告								
40	软件修改报告								
41	软件维护通报								

注：à 表示在软件开发过程某阶段的开始或结尾时完成该文档，不持续整个阶段

○ 表示在软件开发过程某阶段结束时进行评审活动，并编制相应报告

■ 表示该阶段应完成的对应文档

### 3. 文档编号

为便于使用和管理，文档应按代码编制方法进行编号。

常用的代码编制方法有：十进制顺序码、十进制分类码、助记码、隶属码等。

各开发单位可根据自己的需要确定一种编号方法。五伦采用何种方法，编号必须唯一。

## 4. 文档评审

提高软件文档质量的一个有效方法就是在软件开发的各个阶段，对形成的文档进行严格评审。

评审除了可以尽早发现问题，并及时采取措施予以解决，从而确保文档的正确性，同时，也为下一阶段的工作做好组织和技術上的准备。评审主要包括以下几个方面：

### (1) 评审的内容

提交评审的最新版本的文档，检查文档描述内容的正确性、符合性。

### (2) 需求评审

产生一个基于系统要实现功能的共同理解、被用户认可的需求规格说明书。

### (3) 设计评审

产生的最终文档规定系统将如何设计、开发和测试。

### (4) 其它评审

如文档格式、技术准确度、覆盖范围、使用工具、涉众符合性等的评审

## 软件开发过程各评审点文档评审内容要点：

评审点	评审人员	评审文档	评审内容
软件需求评审	软件开发人员 用户 管理人员 标准化人员 特邀专家 质量管理人员	软件需求说明书 数据要求说明书 项目开发计划	<ul style="list-style-type: none"> <li>- 软件需求说明书是否涵盖了用户的所有要求</li> <li>- 软件需求说明书和数据要求说明书的明确性、完整性、一致性、可测试性、可跟踪性</li> <li>- 项目开发计划的合理性</li> <li>- 文档是否符合有关标准规定</li> </ul>
概要设计评审	软件开发人员 管理人员 标准化人员	概要设计说明书	<ul style="list-style-type: none"> <li>- 概要设计说明书是否与软件需求说明书的要求一致</li> <li>- 概要设计说明书是否正确、完整、一致</li> <li>- 系统的模块划分是否合理</li> <li>- 接口定义是否明确</li> <li>- 文档是否符合有关标准规定</li> </ul>
详细设计评审	软件开发人员 管理人员 标准化人员	详细设计说明书 测试计划 数据库设计说明书	<ul style="list-style-type: none"> <li>- 详细设计说明书是否与概要设计说明书的要求一致</li> <li>- 模块内部逻辑结构是否合理，模块间接口是否清晰</li> <li>- 数据库设计说明书是否完整，是否正确反映详细设计说明书的要求</li> <li>- 测试是否全面、合理</li> <li>- 文档是否符合有关标准规定</li> </ul>
鉴定 (验收)	软件开发人员 用户 管理人员 特邀专家 标准化人员 双方上级领导	成套文档	<ul style="list-style-type: none"> <li>- 开发的软件是否已达到软件需求说明书规定的各项技术指标</li> <li>- 使用手册内容是否完整、正确</li> <li>- 文档是否齐全，是否符合有关标准规定</li> </ul>



## (5) 评审程序

评审一般以会议形式进行。其步骤主要有：

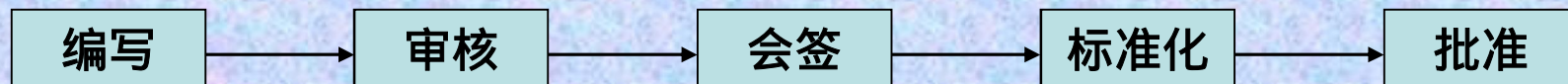
组成评审组，推选组长。成员由开发单位负责人、用户代表、管理人员、标准化人员等担任，必要时可请外单位专家参加；

会前先由开发小组等相关机构完成被评审文档的准备工作，提交评审组成员做好评审准备；

会议时，由编制者说明文档条文，评审组成员进行评议、评审，结束时作出结论，评审组成员在评审结论上签字确认。

## 5. 文档签署

软件产品的所有文档，都应按规定在完成审核、评审后进行签署。签署的过程按下图所示进行，会签仅在必要时才进行。



文档签署不允许代签，文档修改等的签署也一样。

## 6. 文档归档与保管

软件文档是软件开发过程与使用的真实记录，是十分重要的信息资源，必须集中管理、存档，使之能被完善、安全、有效的利用。

## 7. 文档维护

软件产品投入使用后，还会因各种原因需要维护，因此，其配套的文档也需要及时的加以维护。

部分文档资料网站：

1. 中国文档中心 <http://www.28238.com>
2. 中文方案文档站 <http://www.cn-doc.com>
3. 行业文档下载中心 <http://www.35581.com/hywd.asp>