

软件开发规范

Software Development Specification

Version: V1.0

Date: 2010-06-22

Prepared by

Document Revision History 文档修订记录

VERSION 版本	DATE 日期	DESCRIPTION 内容说明	INDIVIDUAL 修订人
1.0	2010-06-22	初稿	

docin 豆丁
www.docin.com

Table of Contents 目录

1	Introduction 简介	5
1.1	Purpose 目标	5
1.2	Scope 范围	6
1.3	Definitions, Acronyms, and Abbreviations. 术语, 缩略词	6
1.4	References 引用	7
1.5	Overview 文档组织	7
2	The Overall Description 概述	8
2.1	Software Development Organizing 开发团队组织结构	8
2.2	Project Base Process 项目基本流程	9
2.3	CMM Base Process CMM 基本过程	10
2.3.1	SCM 软件配置管理	10
2.3.2	SPP 计划策划	12
2.3.3	SPTO 项目追踪	16
2.3.4	PR 同行评审	18
2.3.5	SQA 质量保证	19
2.4	SDLC 生命周期选择	20
2.5	Development Process 开发过程	21
2.5.1	Development Phase 开发阶段	21
2.5.2	Phase Product 阶段制品	22
2.6	Role Duty 角色职责	23
2.7	Constraints 限制	24
3	Specific Requirements 详细描述	25
3.1	Precondition 前提	25
3.1.1	SCM 配置库	25
3.1.2	Test Environment 测试环境	26
3.2	Development Control Process 开发控制流程	26
3.2.1	项目启动和策划阶段	26
3.2.2	需求分析、设计、编码阶段	27
3.2.3	提交测试阶段	27
3.2.4	生产发布、终测	28
3.2.5	发布后问题反馈修改过程	28
3.3	TSP 团队软件过程	30
3.3.1	会议组织	30
3.3.2	沟通问题	30
3.3.3	代码走查	30

3.3.4	其它	31
3.4	<i>PSP 个人软件过程</i>	31
3.4.1	工作原则	31
3.4.2	日常工作	31
3.4.3	DE 开发工程师	32
3.4.4	SCME 配置管理员	33
3.4.5	DBA 数据库管理员	33
3.4.6	Deployer 发布人员	33
4	Tool Specification 工具规范	34
4.1	通用工具	34
4.2	计划	34
4.3	需求分析	34
4.4	设计	35
4.5	编码	35
4.6	测试	35
5	Documents 文档	36
5.1	<i>项目管理文档</i>	36
5.1.1	项目策划	36
5.1.2	项目追踪	36
5.1.3	质量保证	36
5.1.4	项目终止	36
5.2	<i>开发过程文档</i>	36
5.2.1	软件配置管理	36
5.2.2	会议管理	37
5.2.3	计划跟踪	37
5.2.4	评审管理	37
5.2.5	质量管理	37
5.2.6	测试过程	37
5.2.7	问题解决过程	37
5.2.8	其他	38
6	Appendix 附录	38
6.1	<i>易于理解的代码</i>	38
6.2	<i>Log 输出</i>	38

1 Introduction 简介

一个成熟稳定的组织或者团队，能够减少风险，经常地成功地达成目标。成功的含义是：按时、预算内【即符合成本要求】、符合质量要求。换言之，成熟稳定的团队，能够避免以下问题：

- 组织方面出现问题
- 对需求缺乏管理
- 缺乏计划和控制
- 估算错误

同时，还要在以下几个方面做得比较出色：

- 人员调度与工作安排
- 工作量估计
- 预算管理
- 责权分配与平衡
- 执行与监控
- 沟通

本文档是软件开发规范，力求使团队打下一个良好的基础，以便逐步成长为成熟稳定的团队。团队需要一个逐步标准、规范的开发过程，在这个过程中，团队得到锻炼，成员能力得到提高，风险得到控制。

主要内容是：

- 定义软件开发的流程；
- 定义软件开发的文档格式；
- 定义涉及的角色；
- 定义涉及的信息；
- 描述开发流程；

1.1 Purpose 目标

本文档的目标是：

- 统一软件开发团队的流程、文档；
- 促进团队成员的沟通，减少误解；
- 促使程序员书写易维护的代码；
- 提高代码编写效率；
- 使每个成员成为一个高效的程序员；

1.2 Scope 范围

本文档，包含：

- 项目管理的流程：
 - 项目策划
 - 项目追踪
 - 配置管理
 - 质量保证
 - 同行评审
- 涉及文档：
 - 项目计划 mpp
 - 需求规格说明书 SRS
 - Delphi 估算
 - 项目状态报告
 - 配置库样式
 - CheckList
 - 评审表
 - 变更申请表
- 开发工具的规范：
 - 数据库设计工具
 - 功能设计工具
 - IDE
 - 配置工具

1.3 Definitions, Acronyms, and Abbreviations. 术语，缩略词

- | | |
|-----------------|--|
| ➤ SPP | 项目策划 Software Project Planning |
| ➤ SPTO | 项目追踪 Software Project Tracking & Oversight |
| ➤ SCM | 配置管理 Software Configuration Management |
| ➤ SQA | 质量保证 Software Quality Assurance |
| ➤ PR | 同行评审 Peer Review |
| ➤ BaseLine | 基线 |
| ➤ SCCB
Board | 软件配置控制委员会 Software Configuration Control |
| ➤ CR | 变更请求 Change Request |
| ➤ SDLC | 软件开发生命周期 Software Development Life Cycle |
| ➤ RUP | 统一开发过程 Rational Unified Process |
| ➤ XP | 极限【敏捷方法】eXtreme Programming |
| ➤ TDD | 测试驱动 Test Driven Development |

1.4 References 引用

《CMM2》

《CMM3》

1.5 Overview 文档组织

本文档主要分为四大部分：

➤ 概述；

描述了团队组织开发过程的高层视图；

➤ TSP 和 PSP；

按照团队和个人描述流程规范；

➤ 工具规范；

描述了开发工具的详细规范；

➤ 文档；

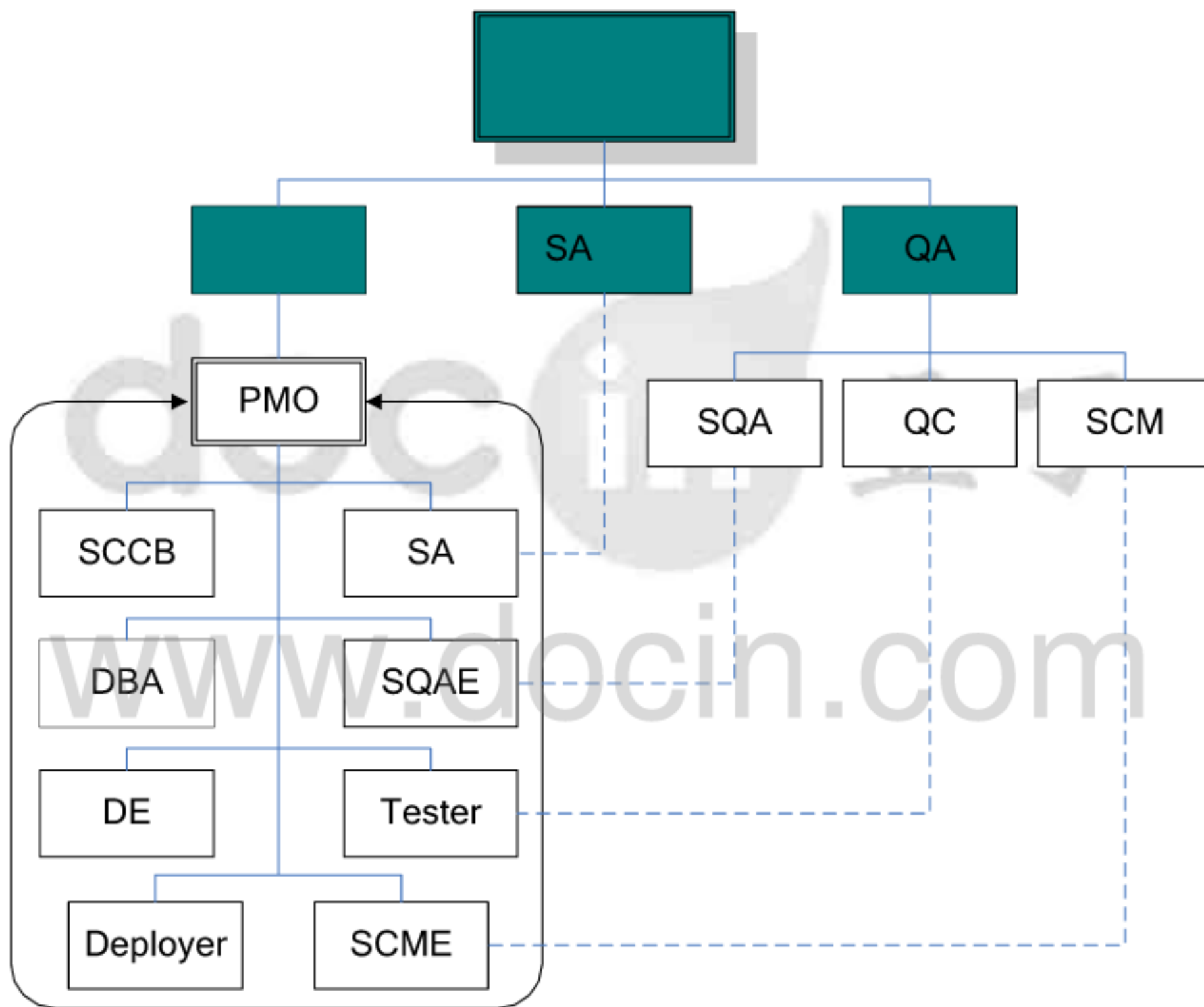
涉及的文档格式；

docin 豆丁
www.docin.com

2 The Overall Description 概述

本部分是开发团队开发过程的高层描述。它描述了开发过程规范的背景，用来和所有涉及各方就基本过程达成共识。

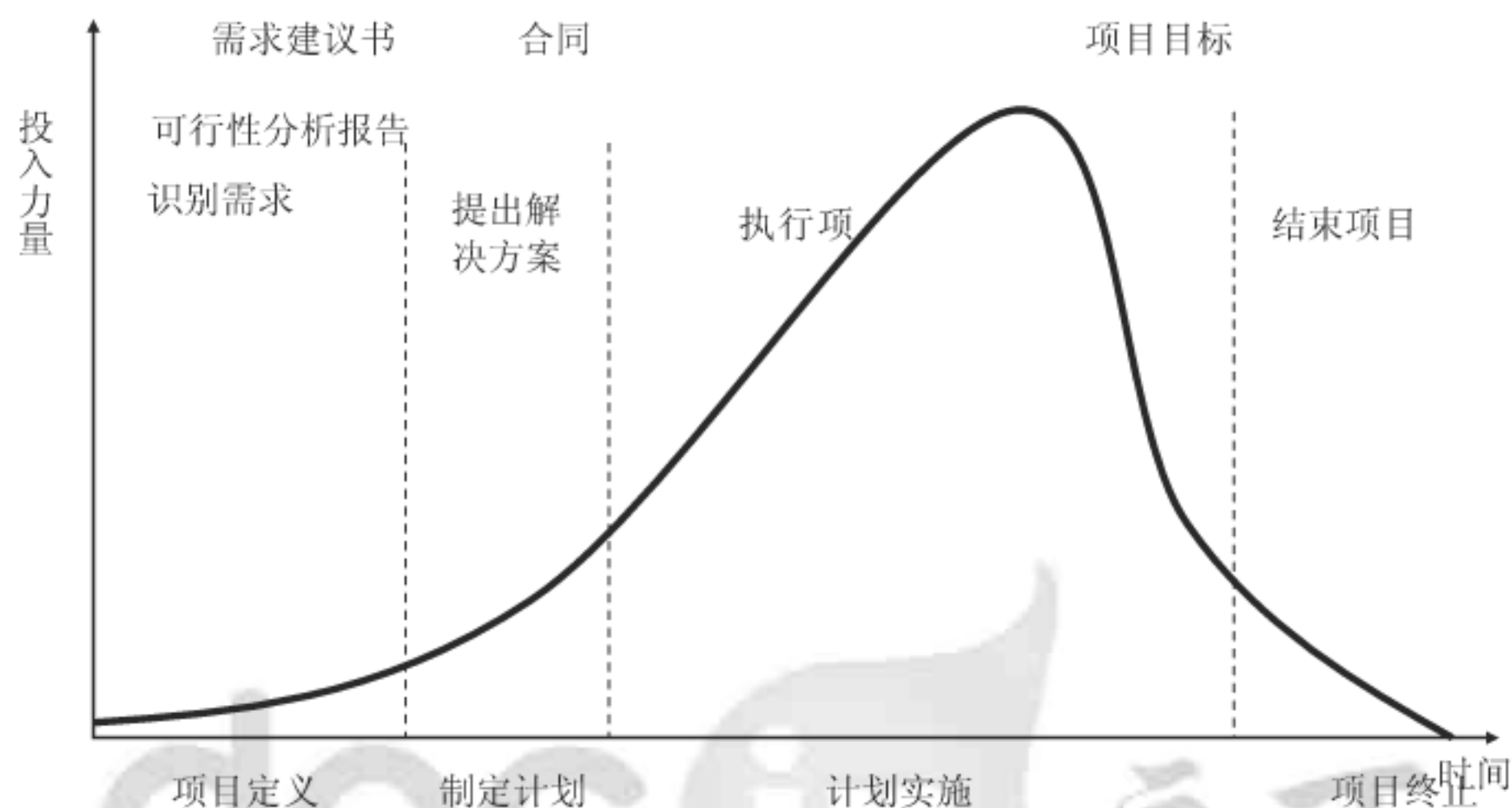
2.1 Software Development Organizing 开发团队组织结构



说明：

- 表示公司的行政部门
- 表示公司的逻辑部门
- 实线表示参加产品实现的组织和人员（不表示所属关系）
- 虚线表示工作的汇报关系，如 SQA 向 SQA 经理汇报。

2.2 Project Base Process 项目基本流程

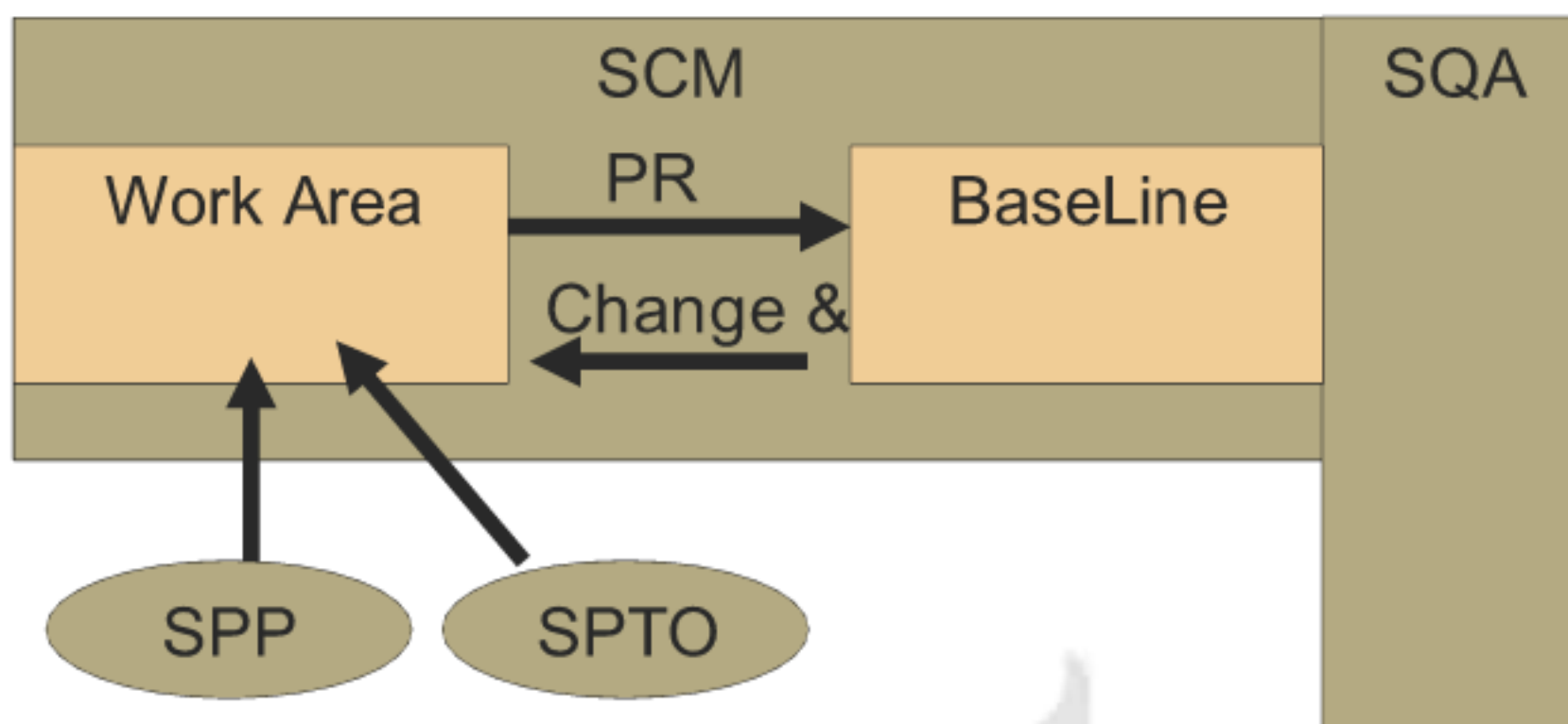


基本流程说明:

- 项目启动: 本阶段主要是进行可行性分析, 定义项目, 识别需求;
- 制定计划: 本阶段主要是计划策划, 估算工作量, 制定具体的可执行的计划;
- 计划实施: 本阶段主要是实施计划, 完成计划中的各项任务, 报告计划状态;
- 项目终止: 计划执行完毕, 总结项目;

www.docin.com

2.3 CMM Base Process CMM 基本过程



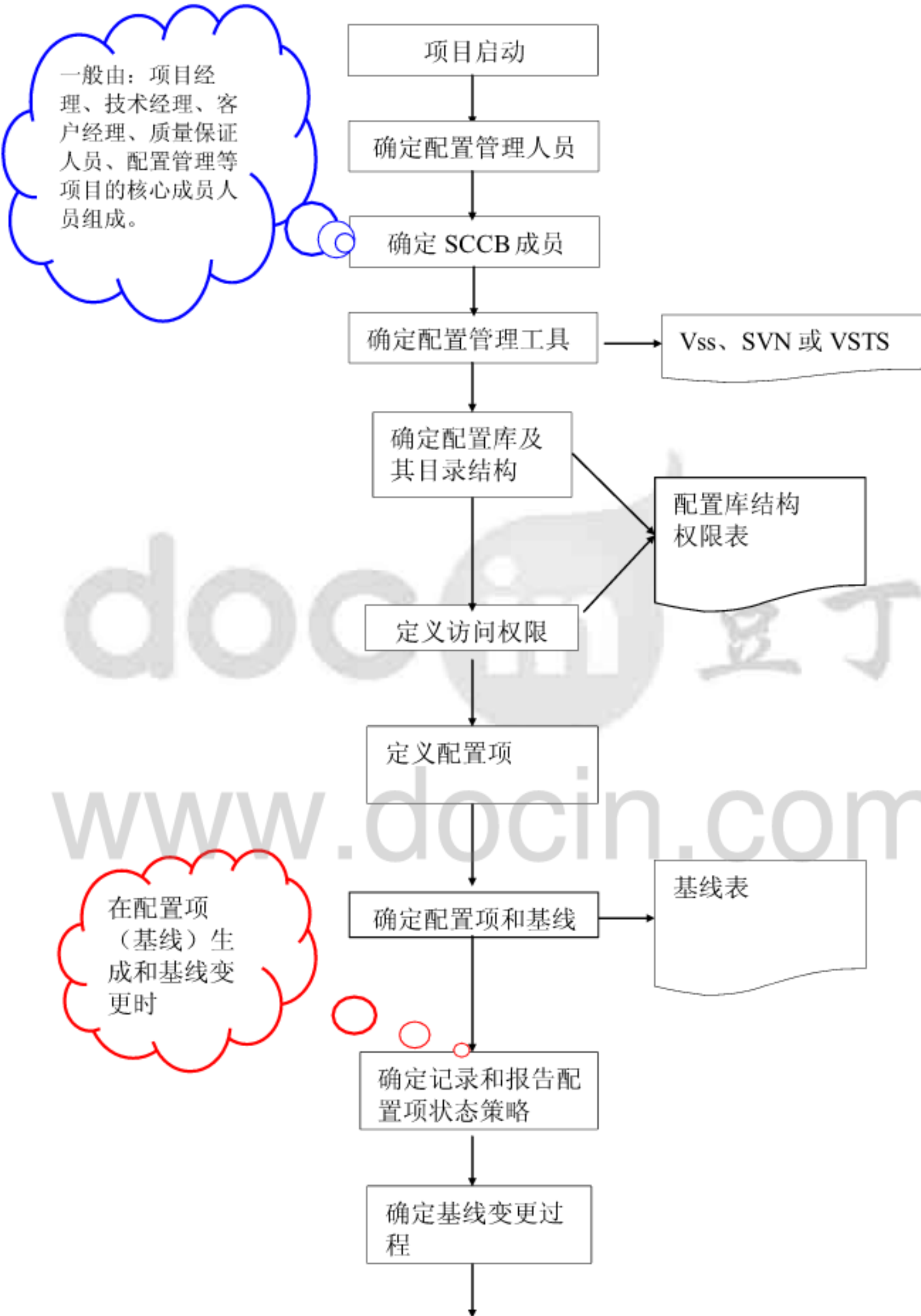
基本过程说明：

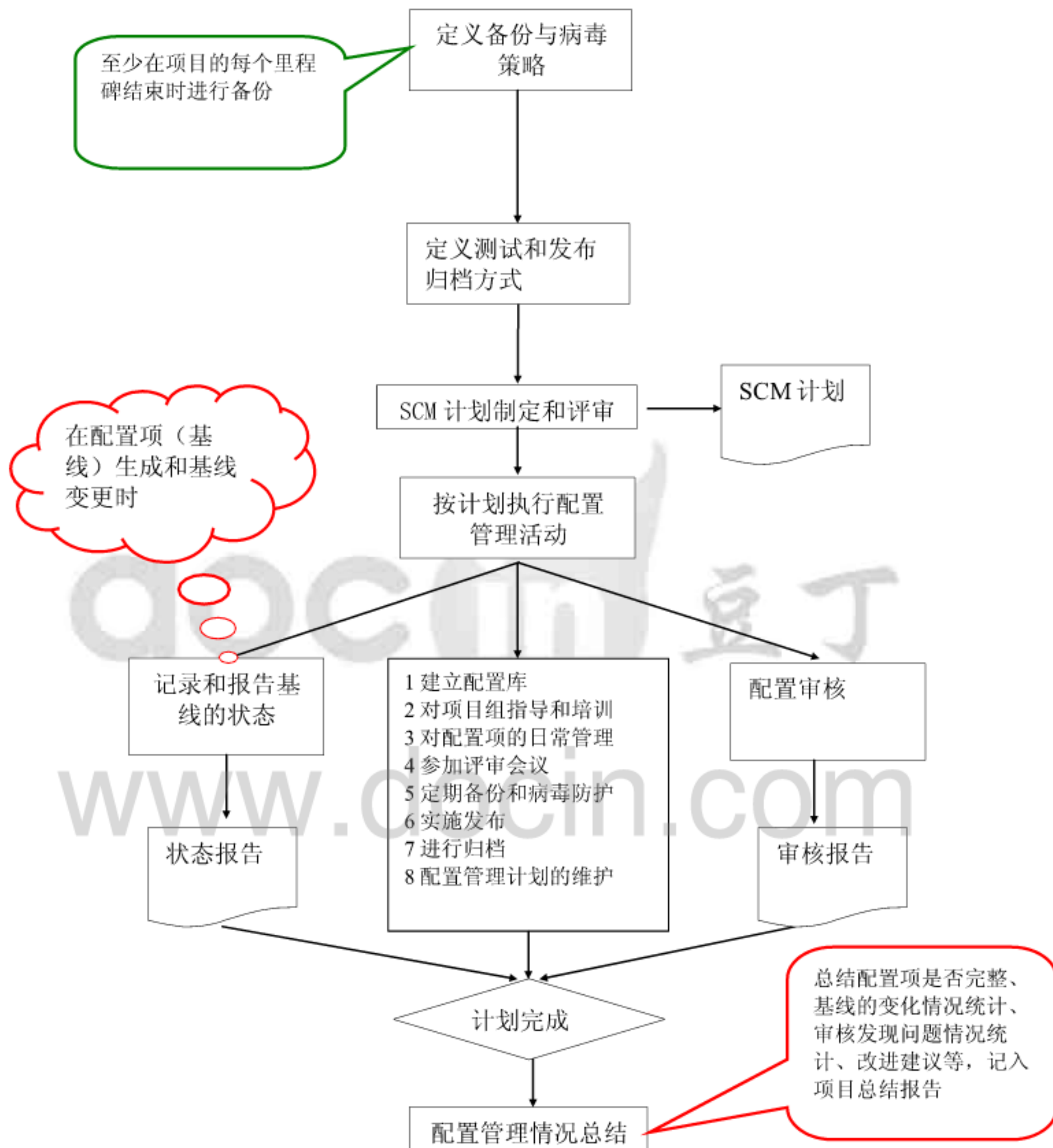
- SCM： 软件配置管理，所有活动的基础，一切制品必须放入配置库；
- SPP： 软件项目策划，估算工作量，制定详细计划【项目的制定计划阶段】；
- SPTO： 项目追踪，报告项目状态，评估并更新计划【项目的计划实施阶段】；
- PR： 同行评审，进入基线的前提条件，降低风险，提高质量的有效手段；
- SQA： 质量保证，预防风险的有效手段；

2.3.1 SCM 软件配置管理

配置管理主要解决：

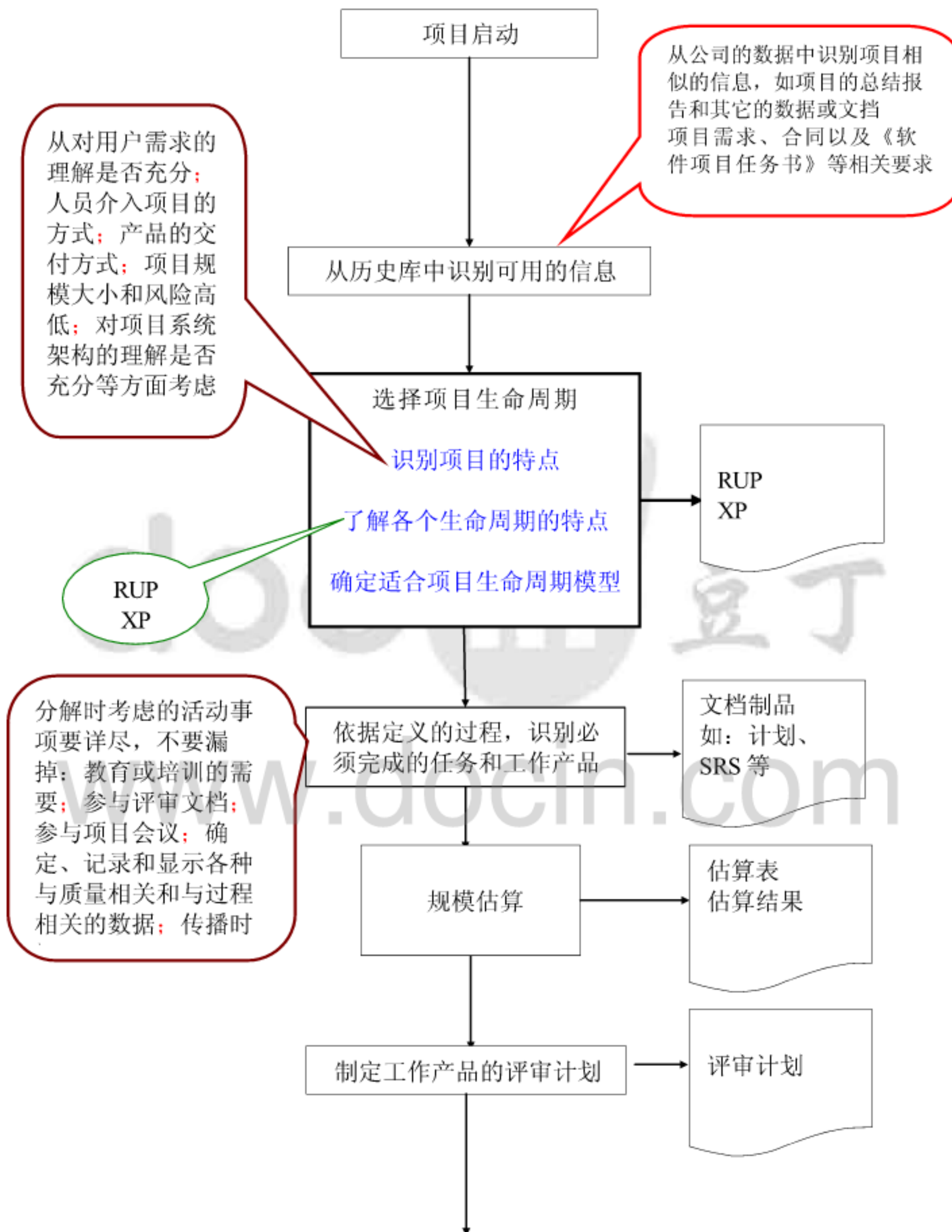
- 版本
- 变更





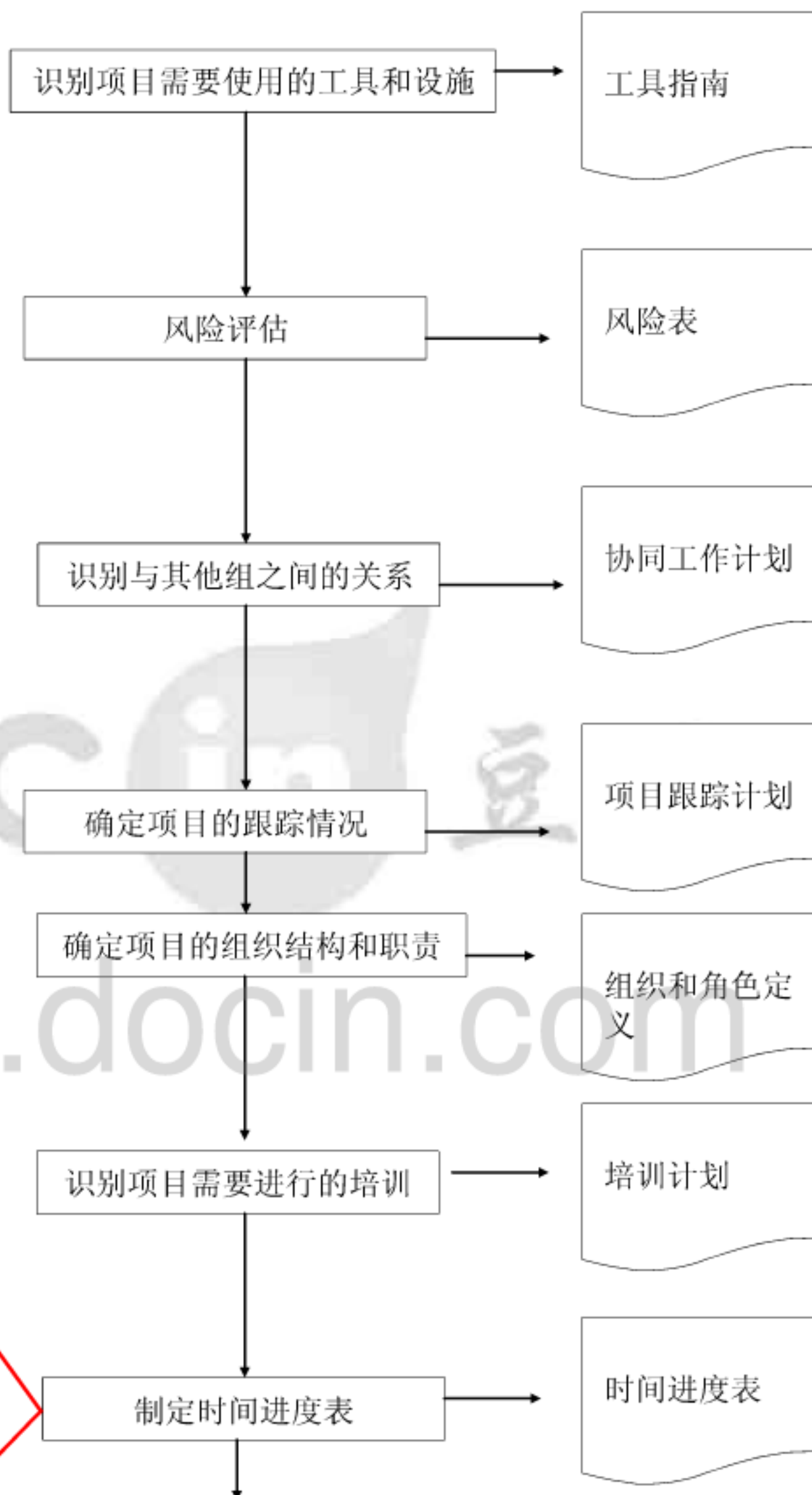
2.3.2 SPP 计划策划

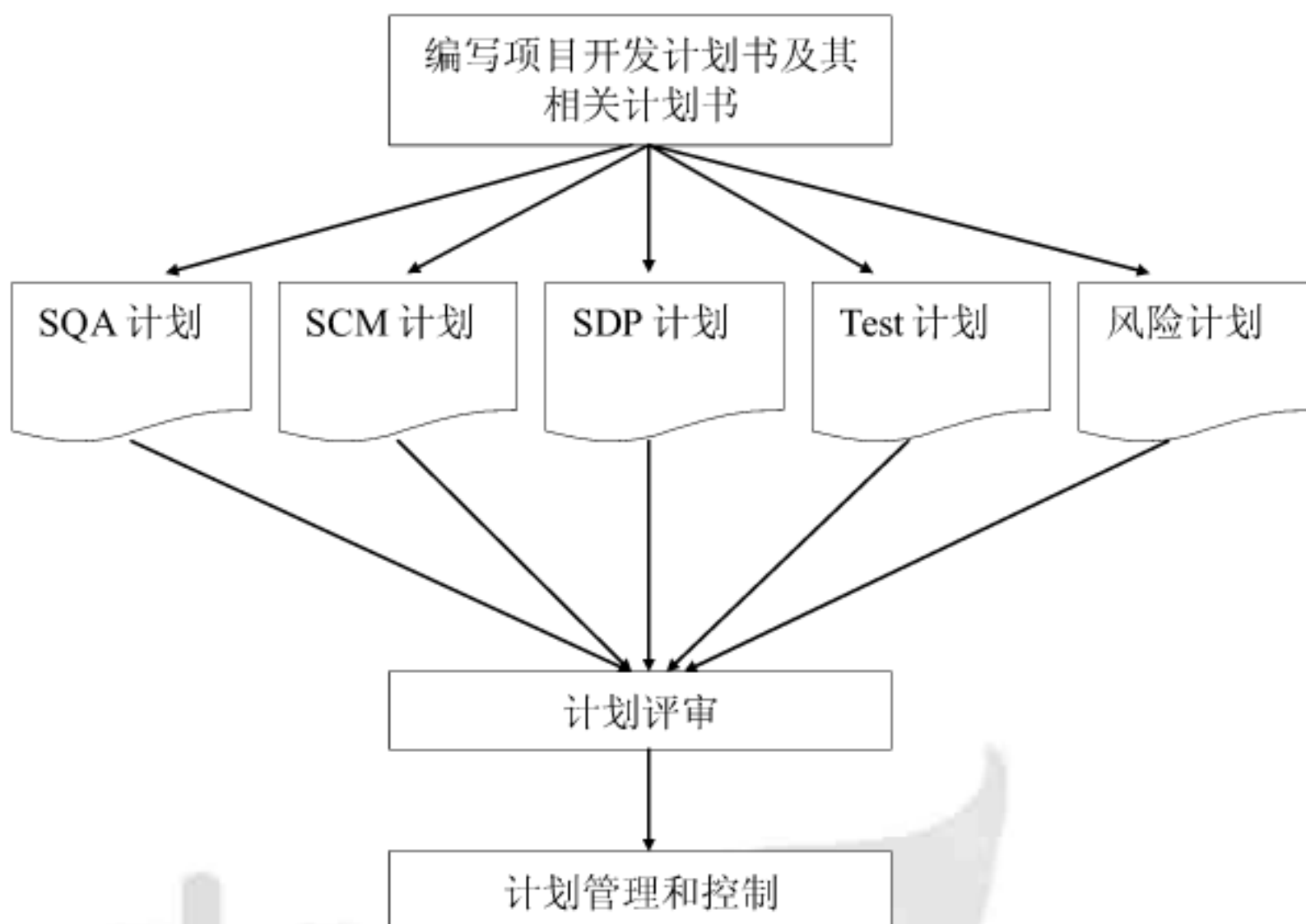
计划策划的核心是工作量估算





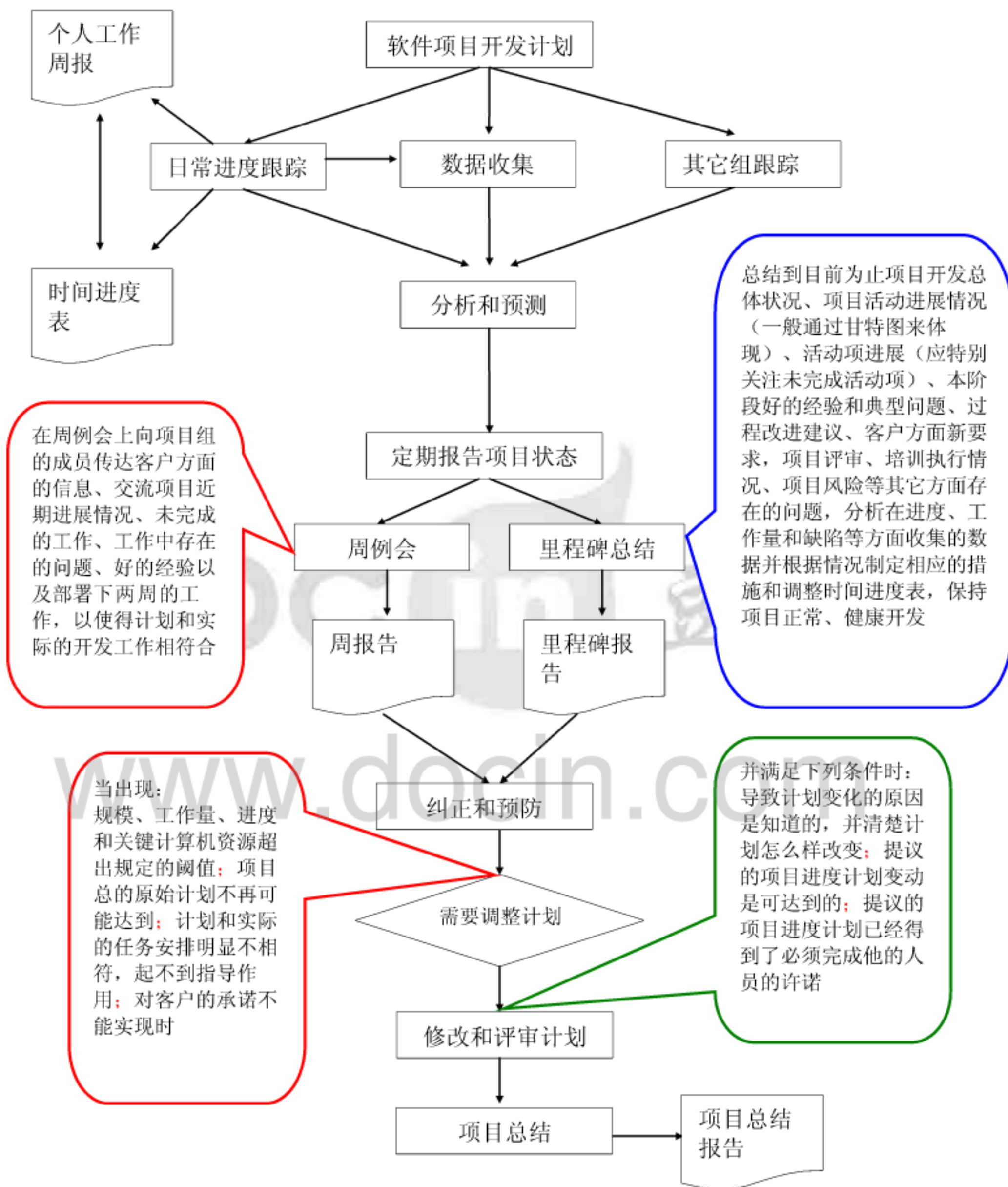
在已知的停工和节假日时间不安排工作；不考虑加班时间；考虑测试及评审中发现问题的返工需要的时间；考虑客户需求的稳定情况；考虑各项活动的交接和信息的传递时间；识别出的风险对活动的影响；在安排工作时应考虑整个项目的效率因素，在正常估算的工期内增加 20~40% 的余量，分配到项目的所有活动中——特别是关键路径中的活动中



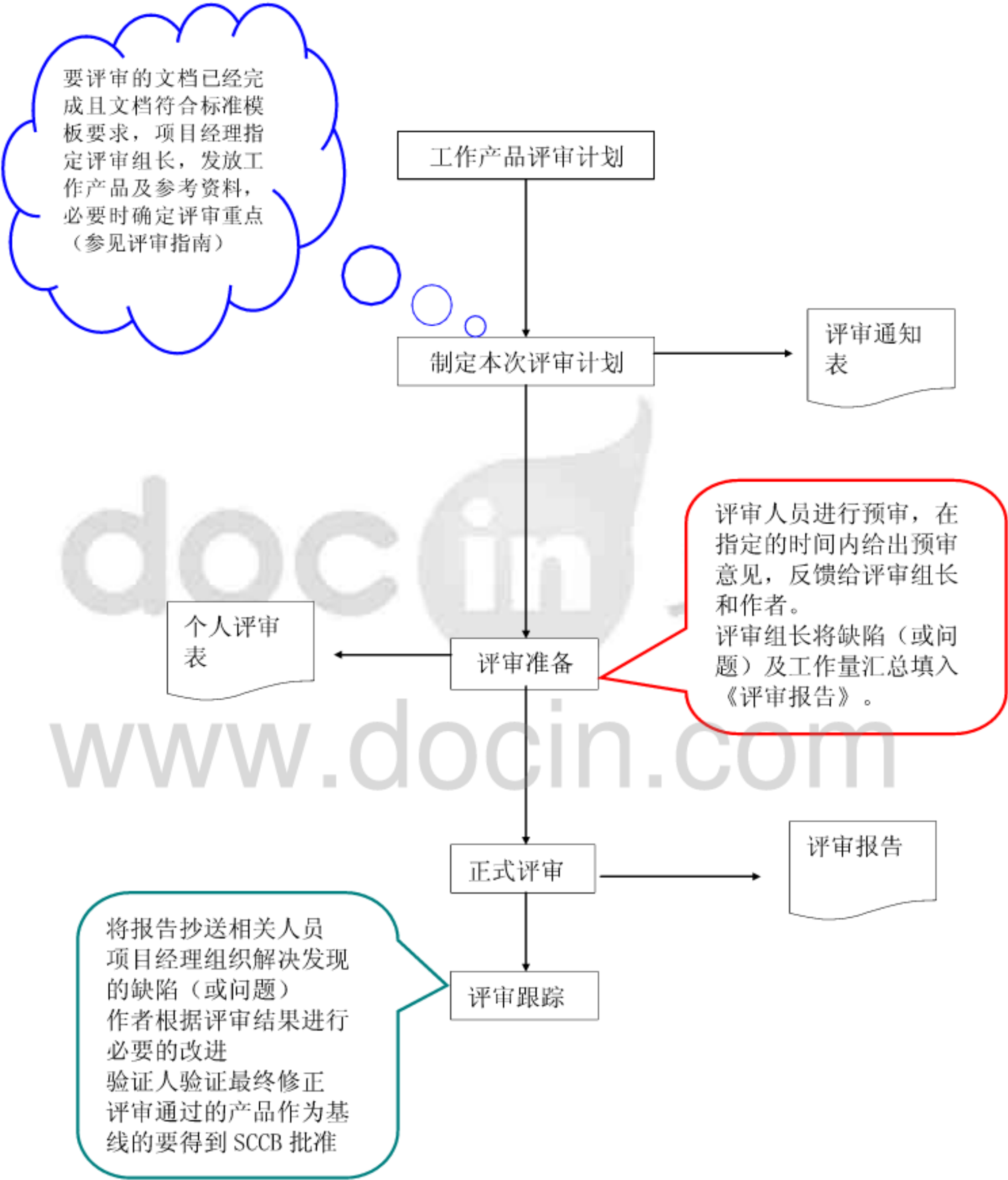


2.3.3 SPTO 项目追踪

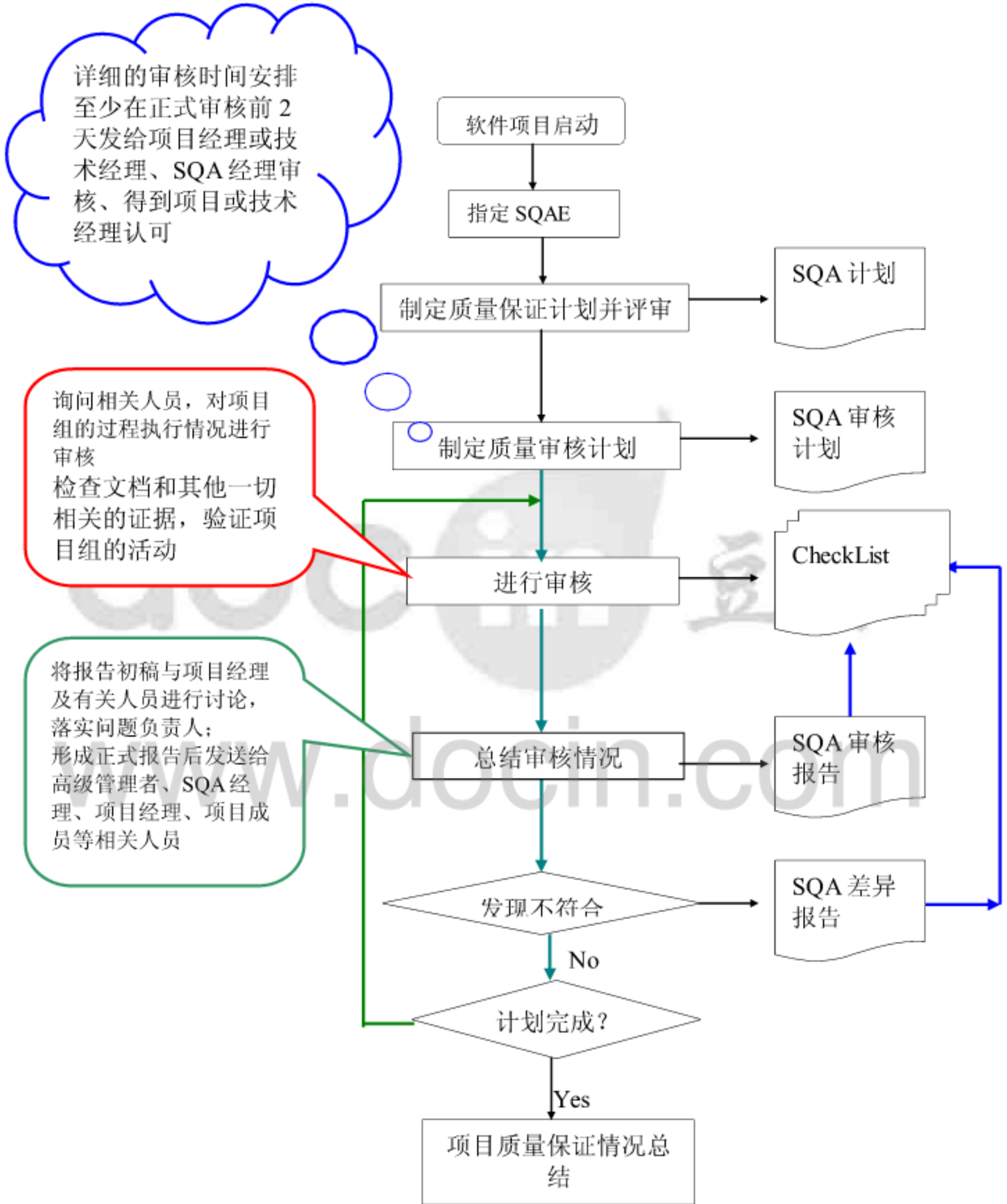
www.docin.com



2.3.4 PR 同行评审



2.3.5 SQA 质量保证



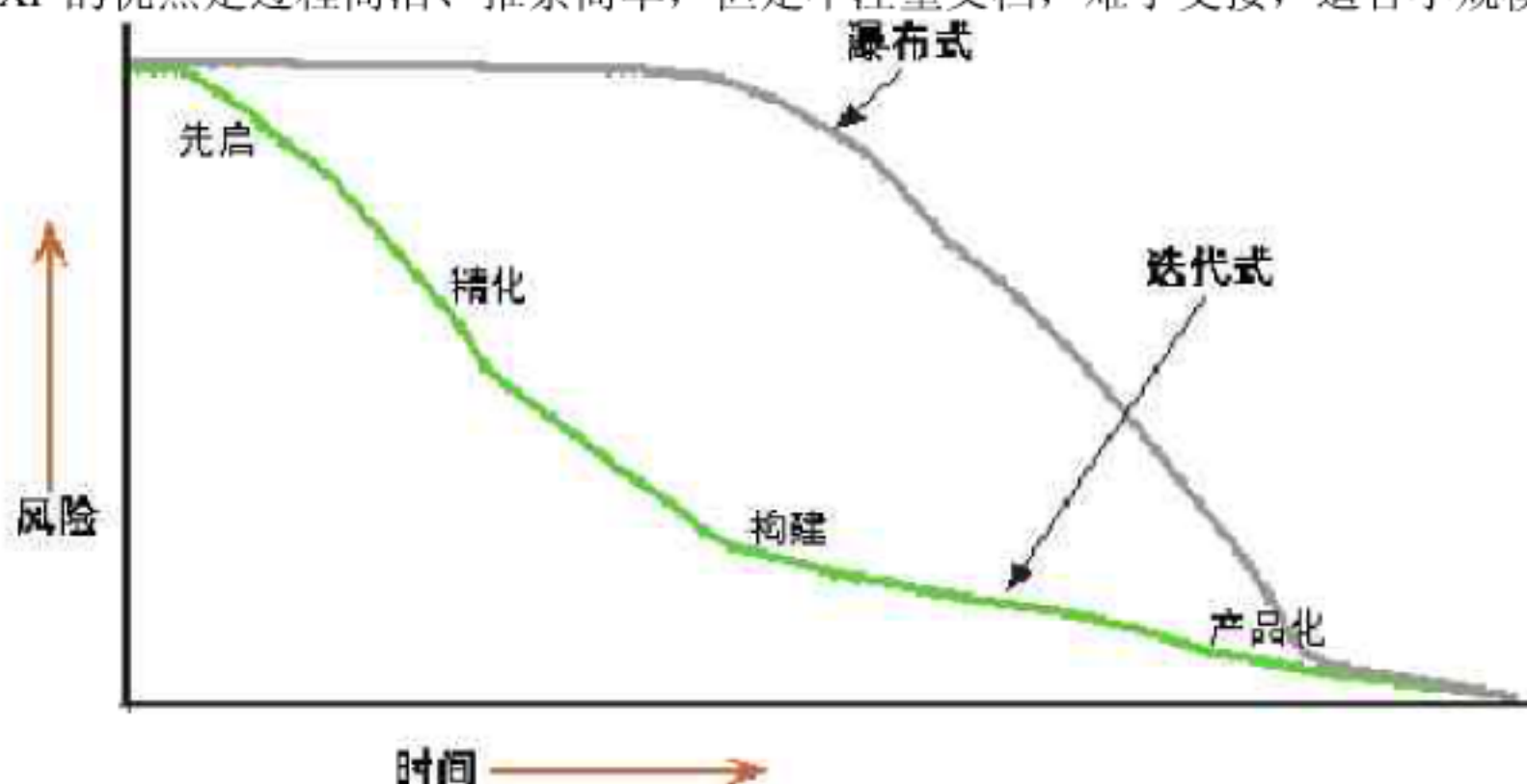
2.4 SDLC 生命周期选择

当前比较成熟稳定的 SDLC 是：

- WaterFall
- RUP
- XP

其中：RUP 和 XP 是迭代式开发过程，风险是可控的。

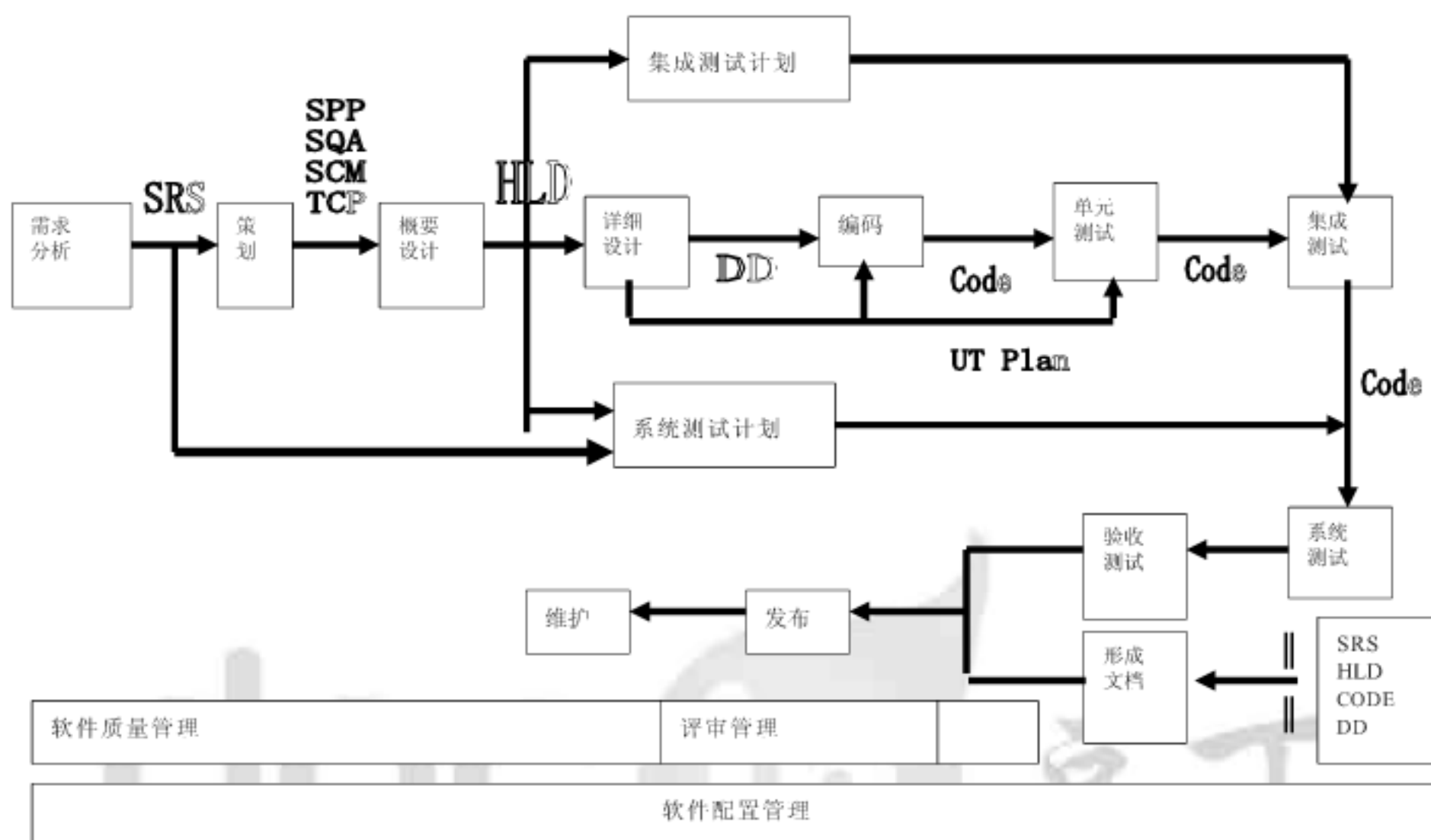
- RUP 的优点是过程清晰、文档齐全，但是过于庞杂，比较适合大规模的团队；
- XP 的优点是过程简洁、推崇简单，但是不注重文档，难于交接，适合小规模团队。



对于中等规模的团队来说，应该基于 RUP 和 XP，进行裁剪，找到适合的 SDLC：

- SDLC 的核心是：迭代式和 TDD
- 从全局看：
 - Use-Case Driven 用例驱动
 - 基于 Architecture
 - 迭代和递增的
- 从微观看：
 - TDD 测试驱动
 - ReFactor 重构
 - Pair 结对编程

2.5 Development Process 开发过程



2.5.1 Development Phase 开发阶段

- 需求分析阶段
 - 需求收集
 - 需求总结
- 总体设计阶段
 - 总体架构
 - 部署模型
- 概要设计阶段
 - 模块划分
 - 数据库设计
- 详细设计阶段
 - 具体实现
- 编码阶段
 - 测试用例
 - Coding
 - 单元测试
- 测试阶段
 - 测试用例
 - 测试

- 修正
- 发布阶段
 - 安装测试
 - 安装系统
 - 维护

2.5.2 Phase Product 阶段制品

- 需求阶段
 - SRS: 需求规格说明书
- 总体设计阶段
 - 总体设计说明书
- 概要设计阶段
 - HLD: 概要设计说明书
 - DB: 数据库设计
 - DFD: 数据流图
 - UI: 用户界面
- 详细设计阶段
 - DD: 详细设计说明书
- 编码阶段
 - Test Case: 测试用例
 - Coding: 源代码
 - UT Test Result: 单元测试报告
- 测试阶段
 - Test Task: 测试任务书
 - Test Case: 测试用例
 - Test Result: 测试报告
 - Test Approvals: 测试总结
- 发布阶段
 - 发布申请书
-

2.6 Role Duty 角色职责

角色	责任
研发经理 【研发团队】	<p>为软件项目提供足够的资源。 保证 SQA 小组的独立性。 解决 SQA 检查时发现的问题。 审批对外的承诺。 定期审查 SCM、SQA、项目计划和跟踪的相关活动。</p> <p>规定系统需求；将系统需求分配给硬件、软件和其他成分； 规定硬件、软件和其他成分的界面； 监控设计和开发以保证他们符合其规格说明； 代表公司下达任务书。</p>
SA 团队	<p>负责网络工程计划的制定及实施； 负责对客户的技术支持与培训； 负责工程服务部内部人员素质与技术培训 负责系统集成工程标识、测试、验收及质量保证； 负责硬件、网络和系统软件产品的最后交付； 负责组织自产软件储运、防护、交付和安装； 负责工程项目的配置管理</p>
QA	<p>研究制定测试规范和方案； 参加实施测试和质量保证过程； 对系统测试中发现的缺陷进行验证；</p> <p>负责组织软件项目任务书、开发计划、里程碑等管理评审； 负责公司的配置管理；</p>
项目经理	<p>负责软件和硬件整个项目的协调、管理</p> <p>进行需求分析，并进行文档的编写 组织技术评审等活动 组织制定项目开发计划（SDP）、风险管理计划等计划 配合与协调 SQA 和 SCM 小组的活动。 管理项目组，执行 SQA 方针和过程以及 SDP。 监督和跟踪 SDP、项目估算</p>
SA	<p>负责硬件工程的实施； 负责系统的上线； 负责系统的维护；</p>

SCCB	授权建立软件基线和标识配置项/单元； 审查和审定对软件基线的更改； 审定由软件基线库制造的产品的生成。
SCM	协助软件项目经理制定SCM计划、维护SCM计划； 制定并维护项目标识规范；按时归档配置项； 标识并管理置于配置管理过程之下的软件工作产品集合； 进行软件项目的软件基线生成、管理和备份； 软件配置状态的统计和审计，并向项目组、软件项目经理、 高级管理者汇报有关活动情况； 将基线的变更情况通知受影响的组和个人； 保存并管理各项评审记录、与项目相关的技术文档、标准和 规程。
SQC	依据测试计划模板制定测试计划。 执行测试计划进行测试并记录测试发现的缺陷 提供测试报告。
SQA	主要是策划软件质量保证活动、检验软件产品或活动对可用的 标准、需求和规则的遵守程度、组织处理项目内部不能解 决的不一致问题； 定期报告检查情况，发现偏差组织制定纠正、预防措施并监 督更正； 参与制定 SQA 计划，实施 SQA 活动，并向 SQA 经理、软件 项目经理项目组、高级管理者汇报有关的情况。
DBA	负责 DB 的创建和维护； 为 DE 提供一个稳定的环境；
DE	按软件开发计划进行开发，并记录相关数据； 遵守公司质量管理体系的要求。
Deployer	根据发布申请，提取代码，发布系统 和 SA、DBA 一起配置环境 重构和重建系统

2.7 Constraints 限制

3 Specific Requirements 详细描述

本部分按照角色划分详细描述开发过程。

3.1 Precondition 前提

3.1.1 SCM配置库

➤ 目录结构

■ 开发库：开发工作区文档和代码

◆ 项目文档

- 项目启动
- 项目策划
- 项目计划
- 项目报告

◆ 开发文档

- 需求
- 设计
- 测试

◆ 代码

- 代码目录

◆ 参考资料

- 客户资料等等

■ 基线库：评审通过后的文档

◆ 《文档同开发库》

■ 测试库：测试代码和测试发布包

◆ 文档

- 计划
- 用例
- 测试报告

◆ 代码

- 版本 1
- 版本 2

◆ 参考资料

■ 产品库：测试通过后的文档和代码

◆ 项目交付制品

- 项目总结
- 验收报告
- ...

- ◆ 项目产品
 - 版本 1
 - 版本 2
- 权限
 - 测试库：
 - ◆ 测试人员可以读写
 - ◆ 其它人员只能读，不能增加、修改和删除
 - ◆
 - 基线库：只能增加，不能删除和修改
 - 产品库：只能增加，不能删除和修改
 - 开发库：

3.1.2 Test Environment 测试环境

- 测试需要一个独立的环境
 - DB 独立
 - FTP 等资源独立
 - Pass9 等外部系统独立
- 最好是一个单独的局域网环境，完全和开发分开
 - 开发是 172.18.0.0 环境
 - 测试是 192.168.0.0 环境
- 每次测试，应当是一个完整的测试过程
 - 安装系统
 - ◆ DB
 - ◆ Web
 - ◆ AppServer
 - ◆ Client
 - ◆ 其它
 - 配置系统
 - ◆ DB 配置
 - ◆ AppServer 配置
 - 系统初始化
 - ◆ 清除所有历史数据
 - ◆ 执行初始化脚本，插入初始数据
 - 测试系统

3.2 Development Control Process 开发控制流程

3.2.1 项目启动和策划阶段

本阶段的关键是定义项目、估算工作量和制定详细计划。

一个软件项目的正式启动从《软件项目任务书》的下达开始。任务书中写明项目的基本信息及相关责任人和详细分工，规定项目必须提交的产品清单。任务书由研发经理或者项目负责人起草，研发经理批准后下达给相关负责人。项目任务书必须为打印纸质文档，由相关人员签字确认后，入配置管理库归档。

软件项目任务书主要作用是明确项目人员职责以及各组之间的协调确认。

估算工作量，从确认需求后开始。由项目经理指定评估人员，先按照头脑风暴法估计各个子系统或者模块的难易程度，然后按照 Delphi 法估算各个部分的工作量。

项目经理和 PMO 成员，根据估算的工作量，制定项目计划。

SQA 和 SCM 分别制定各自的计划。

SCM 需要确定资源库的目录结构和权限结构。

项目经理召集 PMO、SQA、SCM 评审及审核项目计划、SQA 计划、SQA 审核计划、SCM 计划和测试计划。

对于发布后的一般性程序修改，不需要下达软件项目任务书。对于关系重大，需要各组人员协调工作的重大修改，项目负责人可以以任务书的形式明确职责、协调关系。

测试负责人评估测试资源【人员及机器】，并决定测试人员是否介入项目的需求分析和设计阶段。

3.2.2 需求分析、设计、编码阶段

本阶段的关键是评审和修订控制，关键评审需要需求、设计、编码、测试、项目管理、用户等的参与。

需求阶段，需求分析人员收集需求，根据 SRS 模版，作出需求规格说明书。

设计阶段，设计人员根据总体设计、概要设计、数据库设计和详细设计，作出设计文档。

编码阶段，编码人员根据详细设计，设计单元测试用例，编写代码，进行单元测试。

关键评审：SRS 评审，设计评审，代码走查

3.2.3 提交测试阶段

项目启动后，项目经理填写测试任务通知单，将测试任务下达给测试组。概要设计评审完成后，由各子系统或者模块的负责人测算完成时间，在确定完成后（正式开始编码前）将测试任务通知单提交给项目测试负责人，项目测试负责人审核通过在通知单上签字

后返回给予项目负责人。开发及单元测试完成后，由开发人员将测试内容提交配置管理员入测试库后，将测试任务通知单提交给发布人员申请测试发布。发布人员将测试库中本次测试的内容发布到测试机后，在测试任务通知单上签字后，提交给测试人员开始测试。

测试完成后，测试人员在任务单上填写测试意见后，交测试负责人确认后，返还给开发人员。

如测试没有通过，开发人员修改测试内容，进入下一个测试流程。

如通过测试，开发人员将测试任务通知单提交给项目负责人，由项目负责人、SCCB 签字确认后，提交配置管理员将测试内容入基线库。

过程关键：发布实施人员确保发布到测试机上的源程序在配置管理库中得到了有效的标识。

3.2.4 生产发布、终测

程序通过测试入库以后，根据需要，由项目的负责人负责填写发布申请单。发布申请单由项目测试负责人、配置管理员、SCCB、客户代表、研发经理签字确认后，由项目负责人提交给实施发布人员。发布人员拿到签完字的发布申请后，才能从基线库中提取程序向生产机上发布。如以上发布确认人员没有全部签字同意发布，必须由项目经理签字同意后发布。

程序发布到生产机上以后，进入终测【UAT】流程。测试人员和用户代表要对生产机上的程序进行最后测试，确保生产机上的系统符合需求。项目负责人负责同用户协调，项目负责人、测试人员和用户共同编写测试用例。项目负责人将《终测意见书》提交三方签字，根据签字意见决定修订系统或者提交正式发布。

终测出现的问题修改按照基线变更流程进行。

实施人员只有拿到有三方签字的《终测意见书》后才能将系统正式公开发布。系统正式发布三天之后一周之内，由实施人员负责到用户处取得有用户主要负责人签字的《系统运行报告》，项目负责人负责监督执行。根据《系统运行报告》做相应的处理。

过程关键：发布到生产机上的程序都在基线库中得到了有效的标识。

3.2.5 发布后问题反馈修改过程

系统发布之后，用户反馈的意见要形成问题清单或者变更申请单，记录需要修改的地方，提交给项目负责人。项目负责人负责判断改动是否会影响需求或者设计，负责将任务分配给相关人员进行修改。修改完成后，提交测试直至发布。

这个阶段的最重要的是保证所做的修改（文档、代码）都在配置管理库的基线库中得到体现。即基线库中的文档和代码要进行同步更新，关键是发布人员严格根据发布申请单进行控制，并确保发布的代码都是从基线库中取出的。没有经过流程直接要求发布的，发布人员必须予以拒绝。



3.3 TSP 团队软件过程

3.3.1 会议组织

- 会议前，确定会议主持人和记录员
- 向参与会议人员发送会议资料
- 参与会议人员阅读会议资料
- 确定会议主题、日期时间和地点
 - 注意：留出阅读资料的时间
- 确定会议议程
- 准备会议用品【如投影仪等】
- 重要会议，需要签到
- 会议开始前，申明会议纪律
 - 发言时间限制
 - 发言顺序
 - 除主持人外，不得打断别人
- 记录员记录会议纪要
- 会议后，发送会议总结

3.3.2 沟通问题

- 原则
 - 目标明确
 - 明确反馈
 - 反复沟通
- 请求—答复
 - 当有疑问时，发出请求
 - 明确求助对象，指定第一对象和辅助对象
 - 第一对象接收到请求后，不能及时答复的应当转发给自己认为合适的答复人，并告知求助者
 - 求助方式【高一低】：当面，电话，邮件
- 公告

3.3.3 代码走查

- 项目负责人指定代码走查对象
 - 相互走查
 - 循环走查
- 代码走查发现的问题
 - 首先记录

- 告知代码作者
- 更新 CheckList

3.3.4 其它

3.4 PSP 个人软件过程

3.4.1 工作原则

- 计划管理：把你想做的写下来
- 行为管理：按照你写下来的去做
- 报告管理：把做的事情记录下来
- 跟踪管理：出现的问题要设法解决

3.4.2 日常工作

- 每日工作
 - 每日早晨，规划当日工作；
 - ◆ 计划必须细化到一个明确的目标
 - ◆ 计划要有余地，比如会议等
 - ◆ 计划是可执行的，能够完成的
 - ◆ 计划是可监控的
 - 每日下班，总结当日工作；
 - ◆ 计划完成情况
 - ◆ 未能完成原因
 - ◆ 个人心得：新的发现，新的方法，新的问题
- 会议
 - 会议之前，仔细阅读会议资料
 - 如有疑问，可以发邮件向会议主持人提出
 - 或者在会议上提出
 - 会议中，记录会议要点
 - 如要参与讨论，请在别人发言结束后发言，不要打断别人
 - 会议后，如有新的想法，发邮件或者当面向会议主持人提出
- 求助
 - 如果一个问题 20 分钟还不能理出一个头绪，应当立即求助
 - 求助对象：

- ◆ 个人认为能够解决该问题的人为第一对象
- ◆ 不能确认的，项目负责人为第一求助对象
- 发出求助后，个人负责追踪求助，直到解决

➤ 报告

- 认为个人负责的任务不能按时完成的，应当立即报告给负责人
 - ◆ 重要任务：Leader 和项目负责人
 - ◆ 其它任务：Leader
- 提前期：
 - ◆ 重要任务：至少提前 3 天
 - ◆ 其它任务：至少提前任务期的 1/3

3.4.3 DE开发工程师

- 接受任务
- 阅读详细设计文档
 - 从 SCM 获取详细设计文档
 - 阅读文档
 - 如有疑问，向设计人员请教
- 规划个人开发计划
 - 估计开发工作量
 - 制定计划
 - ◆ 单元测试用例
 - ◆ 代码
 - ◆ 单元测试
 - 和负责人协商
 - 提交 SCM
- 设计单元测试用例
 - SCM 获取测试用例模版
 - 编写测试用例
 - 提交 SCM
- 编写代码
 - 从 SCM 获取代码库
 - 编写代码
 - 本地调试
 - 提交 SCM
- 单元测试
 - 代码发布到开发机
 - 请 DBA 协助
 - 如果有其它模块，请负责人协调
 - 测试
- 代码走查

- 根据负责人安排，检查他人的代码
- 和代码作者讨论代码
- 填写走查报告
- 提交代码
 - 提交 SCM
 - 做 Tag 或者其它标记，以便提交集成测试

3.4.4 SCME配置管理员

- 规划 SCM
 - 资源库目录结构
 - 权限
 - 基线
 - 备份
- 基线
 - 经过评审
 - 发送通知
 - 转移资源到相应基线
- 变更管理
 - 接收变更申请
 - 向 SCCB 发送申请
 - 申请通过后，发送变更通知
 - 提取基线到工作区

3.4.5 DBA 数据库管理员

- 规划 DB 的管理
 - DB 的大小
 - 权限划分
 - 备份和恢复
 - 建立 DB 脚本
 - 和 SCM 协商进入 SCM 的资源
- 管理 DB
 - 建立 DB
 - 建立权限
 - 评审数据库设计
 - 导入和导出数据
 - 建立 Table、view 和 index
 -

3.4.6 Deployer 发布人员

- 规划
 - 发布策略
 - 发布脚本
 - 发布计划
- 重构和重建
 - 根据项目特点制定重构和重建计划
 - 编写重建脚本
 - 编写测试脚本
 - 从 SCM 提取资源
 - 重建系统
 - 执行测试脚本
- 发布
 - 接收发布任务书
 - 从 SCM 提取资源
 - 和 DBA 建立 DB
 - 和 SA 建立环境
 - 发布系统

4 Tool Specification 工具规范

4.1 通用工具

- 会议报告
 - PPT
 - Word
 - MPP
- 会议记录
 - Word
- CheckList
 - Excel

4.2 计划

- 项目计划
 - MPP
- 风险计划
 - Word

4.3 需求分析

- SRS 文档
 - Word
- Use Case 图
 - Rational Rose
- 对象图
 - Rational Rose
- 序列图
 - Rational Rose
- 流程图
 - Visio

4.4 设计

- 总体设计
 - Word
 - Rational Rose
- 概要设计
 - Word
 - Rational Rose
- 数据库设计
 - Visio
- 详细设计
 - Word

4.5 编码

- 测试用例
 - Word
- 编码
 - Visual Studio 2005
- 单元测试
 - Word

4.6 测试

- Test Director

5 Documents 文档

5.1 项目管理文档

5.1.1 项目策划

- Delphi 估算表
- 项目计划模版
- 测试计划模版
- 项目清单

5.1.2 项目追踪

- 项目日进度报告
- 项目周报
- 项目周计划报告
- 项目月度报告
- 项目月度状态报告
- 签到簿

5.1.3 质量保证

- 项目策划过程检查表
- 配置管理活动检查表
- 项目跟踪情况检查表

5.1.4 项目终止

- 软件合格验收标准
- 项目验收报告

5.2 开发过程文档

5.2.1 软件配置管理

- 软件配置管理工具
- 变更申请表

- 变更通知单

5.2.2 会议管理

- 会议记录

5.2.3 计划跟踪

- 开发计划
- 测试计划
- 工作日志

5.2.4 评审管理

- 评审指南
- 评审申请表
- 个人评审记录
- 评审报告

5.2.5 质量管理

- 文档编写规范
- SRS
- 总体设计报告
- 概要设计报告
- 详细设计报告
- 编码规范

5.2.6 测试过程

- 测试用例
- 测试报告

5.2.7 问题解决过程

- 问题跟踪表
- 软件项目任务书
- 测试任务通知单
- 发布申请单
- 系统运行报告
- 终测意见书

5.2.8 其他

- 用户手册

6 Appendix 附录

6.1 易于理解的代码

- 经常注释
- 注释内容是做了什么，不能是为什么做
- 检查错误，尤其是函数入口
- 变量名要简洁易懂
- 经常使用 Static 常量

6.2 Log 输出

- 截获抛出异常的地方，要输出 Error
- 返回 True/False 的方法，如果返回 False，要输出 Debug
- 所有调用其它系统接口的地方，都要输出 Info，内容是参数和返回值；
- 输出前，判断是否允许输出，比如：isDebugEnabled

本文由成都网站建设：<http://www.cddgg.com/>成都软件开发：<http://www.dggri.com/> 成都公司代理：<http://www.dggcw.com/> 组织发布。