**设计开发规程**

**编号：XZSKC-21**

**编写： 罗娇 日期： 2022/1/6**

**审核： 朱绍攀 日期： 2022/1/17**

**批准： 朱绍攀 日期： 2022/1/27**

**徐州市勘察测绘研究院有限公司**

目录

[1 简介 2](#_Toc72960548)

[1.1 目的 2](#_Toc72960549)

[1.2 适用范围与预期读者 3](#_Toc72960550)

[1.3 术语表 3](#_Toc72960551)

[1.4 参考资料 3](#_Toc72960552)

[2 过程总体描述 3](#_Toc72960553)

[2.1 过程概述 3](#_Toc72960554)

[2.2 过程流程图 4](#_Toc72960555)

[3 过程元素描述 5](#_Toc72960556)

[3.1 产品方案 5](#_Toc72960557)

[3.2 产品设计 6](#_Toc72960558)

[3.3 产品实现 10](#_Toc72960559)

# 简介

## 目的

本过程目的在于为需求设计、开发、实现解决方案。根据适当情况，解决方案、设计和实现包括单独的产品、产品组件以及产品相关的生命周期的过程，或者它们的组合，以及包括如何利用准则进行接口设计。

## 适用范围与预期读者

适用于本公司的所有软件项目。

预期读者：1、项目经理；

2、项目组；

## 术语表

* TS： TECHNICAL SOLUTION ，技术解决方案简称。

## 参考资料

《CMMI V2.0 Development Model》

# 过程总体描述

## 过程概述

技术解决方案过程包括：1、选择最佳解决方案；2、制定架构设计；3、制定概要设计；4、制定详细设计和数据库设计；5、利用准则进行接口设计；6、实现设计；7、进行单元测试；8、进行代码审查；9、建立和维护用户文档。

## 过程流程图



# 过程元素描述

## 产品方案



### 目的

针对需求选择出最佳解决方案。

### 角色和职责

* **项目经理：**选择产品的解决方案。
* **项目组：**选择产品的解决方案。

### 进入准则

* 《软件需求规格说明书》已完成，根据实际问题，已有备选解决方案。

### 输入

* 《软件需求规格说明书》
* 备选解决方案

### 工作任务

在《软件需求规格说明书》完成后，项目选择最佳技术解决方案时，依据《决策分析指南》启用《决策分析规程》，选择最佳解决方案。

### 退出准则

* 选择出最佳解决方案

### 输出

* 《决策分析报告》

## 产品设计



### 目的

进行架构设计、概要设计、详细设计、接口设计和数据库设计，完成产品或产品模块的设计。

### 角色和职责

* **项目经理：**组织进行架构设计、概要设计、详细设计、接口设计和数据库设计，组织相关人员进行设计文档评审。
* **项目组：**进行架构设计、概要设计、详细设计、接口设计和数据库设计，评审相关文档。

### 进入准则

* 已选择出最佳解决方案。

### 输入

* 《决策分析报告》

### 工作任务

1.概要设计

（1） 面向结构（数据流）设计方法

结合《软件需求规格说明书》及系统不同层次的数据流图，把整个系统分解为模块并确定各模块功能，确定每个功能模块的输入、输出数据流，并考虑各方面数据接口、存储方式等。迭代逐步分解各系统模块，直到确定每个功能模块只执行一个行为为止，同时细化数据结构设计，完善数据字典以确保系统能获得所需的输出结果，从而完成系统构架设计。

（2） 面向对象设计方法

结合《软件需求规格说明书》（包括原型图、类模型）和系统选用的编程语言，确定客户类（包括子类）及类的各种行为，并使用设计模型及相应的列表，确定每个对象的客户（模块）关联，并给予必要的方法设计。迭代逐步检查对象及其方法，直到系统中每个对象拥有所有必需的方法，从而完成系统构架设计。

2.工作步骤

项目经理确定系统设计人员，并讨论确定阶段出口准则，通盘考虑并跟踪上游顺延下来的进度、技术难度等风险、问题，与相关人员沟通，制定阶段计划。必要时，请高层参与问题解决。

评估、确认需求分析结果，并对可重用的软件或功能模块进行性能测试，通盘考虑整个系统结构、功能扩展性。

1. 系统结构设计可选（系统框架设计）

a）传统架构

用传统的架构，1000并发量需要2太服务器做tomcat集群。但是由于系统用的人越来越多，并发量越来越大待到并发量为10000时，需要20台服务器做tomcat集群。当tomcat集群中节点数量增加，服务器能量先增加后下降。所以集群中节点的数量不能太多，一般也就5个左右。难以实现高数据量的并发，5台服务器难以正常工作。

b）分布式架构

能用硬件解决的问题就不用软件解决、硬件解决不了就用软件解决。

当并发数高时，需要按照功能吧系统拆分，拆分成独立的功能。单独为某一个节点添加服务器。需要系统之间配合才能完成整个业务逻辑，叫做分布式。

分布式架构：把系统安装模块拆分成多个子系统，多个子系统相互协作才能完成业务流程，系统之间需要进行通信。

c）基于SOA的架构

为了解决分布式的架构的缺点，我们可以使用基于SOA的架构。SOA：Service Oriented Architecture 面向服务架构。也就是把工程拆分成服务层、表现层两个工程。服务层中包含业务逻辑，只需要对外提供服务即可。表现层只需要处理和页面的交互，业务逻辑都是调用服务层的服务来实现。

（2） 接口（包括用户界面）设计

a) 与客户、测试人员交流界面设计需求，明确用户界面、接口设计规则，包括：标准控件的使用规则，通用界面（包括主界面和子界面等）、接口设计原则等；

b) 界面设计

分析需求说明中对用户界面的需求，实施用户界面设计，包括界面及其关系、工作流程等，必要时采取原型设计，并请客户或同行评估后细化改进；

c) 接口设计

扩展子系统或功能模块及其之间的关系和限制条件，实施系统所需的接口设计，并消除冗余后，完善系统的数据流图，必要时形成功能说明和操作方式。若面向对象方法，则为子系统包、类间的属性、方法等设计；

由测试人员参与完善测试接口设计；

结合系统错误处理和数据验证方法，验证接口设计结果，并逆向需求求证；

d) 确定新开发、采购和复用方案

利用决策分析过程确定新开发、采购和复用方案。

（3）购买复用分析：

①确定哪些产品部件应当采购、外包开发或者自主研发，说明理由。

②分析相应的风险。

③列出多个可选方案。（即使没有可选方案也应有“用”和“不用”两个方案）

（3）非功能性设计（可选）

a）易用性与可执行

易用性会使产品提高符合用户习惯的能力以及其对使用的期望。它会对用户使用产品的生产效率、错误率以及用户对新产品的接收程度产生很大的影响。

可执行需求是指产品可以在给定的时间或者特定的精确度来执行某些任务，或者在一段时间内的极端状态值。在考虑执行需求时，可以从完成任务的速度、结果的精确度、容量、允许值的范围、单位时间内完成的任务数、资源的使用效率、两次故障间的平均无故障时间、连续不停机时间等方面入手。它还应该包括对风险的控制内容。

b)安全性

安全性指产品消除潜在风险的能力和对风险的承受能力。包含、保密性、可靠性和完整性三个子特性。保密性指的是数据不能被授权用户以外的任何人访问的能力。可靠性指的是授权用户可以不受阻止的访问数据、与其它软件的兼容的能力和产品的强壮度。完整性指的是安预期目标完成任务的能力。

一般分为程序安全、系统安全、数据安全。程序安全是指开发的程序是否是安全的，程序上有没有安全的漏洞，例如Web开发中服务器代码没有对输入的参数进行验证，从而导致客户端机器人轻易的获取数据。系统安全指的是系统整体的安全，例如安全的粒度，未经授权的用户是否可以轻易的访问非法的数据等。数据的安全是对数据的保护，数据库中数据有没有做审核，用户之间是否会共享数据等。

c)系统的可扩充性与可维护性

这里指的是当系统达到瓶颈的时候怎样在不修改代码的情况下提供系统的负载能力，扩展一般分为Scale UP和Scale Out。一般情况下会综合运用UP和OUT。例如，增加服务器的性能来提高系统的处理能力，但是任何计算机都会有一定的瓶颈，当增加服务器性能不能达到提高系统性能的时候，我们需要考虑横向的扩展服务器，也即Scale Out。在Scale Out时一般需要我们的系统是状态无关的，即Stateless。

（4） 整合及评审

项目经理或客户委派专人负责组织对设计的工作产品执行评审；

指定需求跟踪人负责跟踪系统设计结果，完善《需求跟踪矩阵》。

2.详细设计

（1） 细化系统设计

项目经理确定详细设计人员，并通盘考虑上游顺延下来的进度、技术难度风险、问题，制定阶段工作计划，确定阶段出口准则。必要时，请高层参与解决问题；

项目经理配合详细设计人员对概要设计方案进行评估，项目组间或组内达成共识；

结合设计方法、工具、需求文档和软件系统体系结构设计文档，逐步细化设计每个功能模块的主要接口与属性，必要时还须细化每个用户界面；若采用面向对象方法，则为设计类的函数和成员变量；

细化设计每个功能模块的数据结构与算法（若存在的话），并提高其效率，确认并完善重用软件及模块单元的算法和处理流程，确保系统一致性；

处理数据流程并充分考虑系统限制，逐步完善系统集成方案；

指定需求跟踪负责人对需求状态进行跟踪，完善《需求跟踪矩阵》，若发现问题，登记到《问题列表及跟踪》中；

重复执行以上步骤直到达到出口准则；

1. 数据库设计的内容
   1. 数据库设计准备：

系统设计人员阅读《需求规格说明书》、和《概要设计说明书》，明确数据库设计任务，准备设计工具。

* 1. 逻辑设计：

系统设计人员根据需求文档，创建与数据库相关的那部分实体关系图。明确用户需求，对于特殊用户创建特别的用户组并分配相应权限。

* 1. 物理设计：

系统设计人员进行表结构设计。一般地，实体对应于表，实体的属性对应于表的列，实体之间的关系成为表的约束。逻辑设计中的实体大部分可以转换成物理设计中的表，但是它们并不一定是一一对应的。系统设计人员要对表结构进行规范化处理。

* 1. 安全性设计：

系统设计人员要提高系统的安全性，应当从“管理”和“设计”两方面着手。这里仅考虑数据库的安全性设计。用户只能用账号登录到应用软件，通过应用软件访问数据，而没有其他途径可以操作数据库。对用户账号的密码进行加密处理，确保在任何地方都不会出现密码的明文。确定每个角色对数据库表的操作权限，如创建、检索、更新、删除等。每个角色拥有刚好能够完成任务的权限，不多也不少。在应用时再为用户分配角色，则每个用户的权限等于他所兼角色的权限之和。

* 1. 设计优化：

分析并优化数据库的效率，尽可能地“提高处理速度”并且“降低数据库占用的空间”。分析效率的瓶颈，找出优化对象（目标），并确定优先级。当优化对象（目标）之间存在对抗时，给出折衷方案，给出优化的具体措施，例如优化数据库环境参数，对表格进行反规范化处理等。

* 1. 撰写数据库设计文档：

系统设计人员利用工具撰写《详细设计说明书》。

* 1. 对数据库设计进行评审：

数据库设计结束之后，产生《详细设计说明书》，项目经理需组织同行专家等所有相关人员对设计成果进行评审；评审不通过，系统设计人员要重新修改。

（3） 整合及评审

项目经理负责组织整合设计内容，编写《详细设计说明书》并完善用户文档；

项目经理负责组织对阶段工作产品的验证和评审，执行评审。

### 3.2.6退出准则

* 设计完成。

### 3.2.7输出

* 《概要设计说明书》
* 《详细设计说明书》
* 《概要设计评审》
* 《详细设计评审》

## 产品实现



### 目的

依据已经完成的设计，实现各产品模块。

### 角色和职责

* **项目组：**实现设计。

### 进入准则

* 设计完成。

### 输入

* 《概要设计说明书》
* 《详细设计说明书》

### 工作任务

1．实现设计

（1） 引用编码规范

确定适用于项目组的编码规则，统一定义有意义的变量名、模块名，注释规则，提高程序的可靠性和可维护性；

（2） 编码

开发人员组建开发环境，避免硬件操作系统等的不兼容风险；

开发人员基于架构设计在配置环境下把功能模块翻译成代码，并进行必要的调试；

（3） 代码评审

项目经理和项目组相互进行代码评审，形成《代码检查表》。

（4） 单元测试

编码人员根据《单元测试用例》对同伴的代码进行交叉单元测试工作。项目经理可根据代码的重要程度选择需要进行交叉单元测试的模块。对测试中发现的BUG，由单元测试人员填写在《单元测试BUG管理表》，并在《单元测试报告》中对缺陷进行分析。

（5）形成用户文档

编制用户相关手册

### 退出准则

* 完成用户相关手册。

### 输出

* 《单元测试用例》
* 《代码检查表》
* 《用户手册》
* 源代码
* 《单元测试用例》
* 《单元测试BUG管理表》
* 《单元测试报告》

修订状况记录表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **版本号** | **页码** | **修改条款** | **审核** | **批准** | **批准日期** | **实施日期** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |