

# 东软实训手册

东软 IT 实训中心



一、整体说明  
二、项目策划  
三、需求理解  
四、概要设计  
五、详细设计  
六、编码  
七、单元测试  
八、系统测试  
九、部署  
十、项目总结  
十一、项目答辩

# 一、整体说明



# 整体说明

## 手册目的

- 用于规范整个实训项目的流程。
- 用于指导实训经理对项目的整体控制。
- 用于指导实训学员对项目过程的理解。
- 用于实训结束后对实训项目质量的验证。

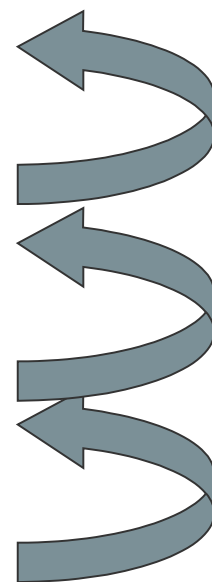
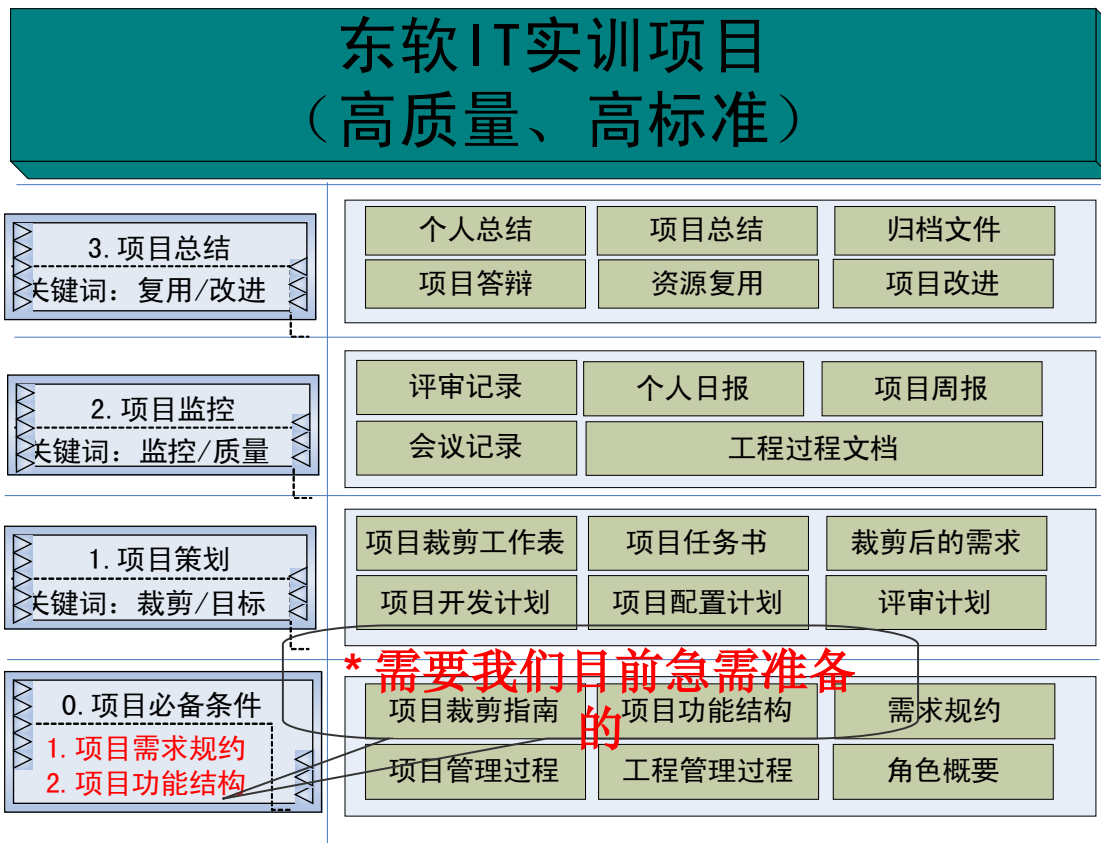
## 适用范围

- 本手册适用 IT 实训中心的 Java 开发项目，其他方向的项目可在此基础上进行修改。



# 整体说明

## 实训项目整体开发流程



案例: 电信报表系统功能结构、人事系统需求规约

# 整体说明

## 实训目标

目标：多维度能力提升

技术能力：从知识到技能的转变

工程能力：需求、设计、编码、测试的全过程

职业素质能力：沟通、主动性、团队等

技术能力

职业素质能力

工程能力

# 整体说明

## 实训特色

项目管理

- CMMI
- RUP
- OSSP

方法 / 工具

- 面向对象
- UML/Eclipse/powerDesinger

实训方式

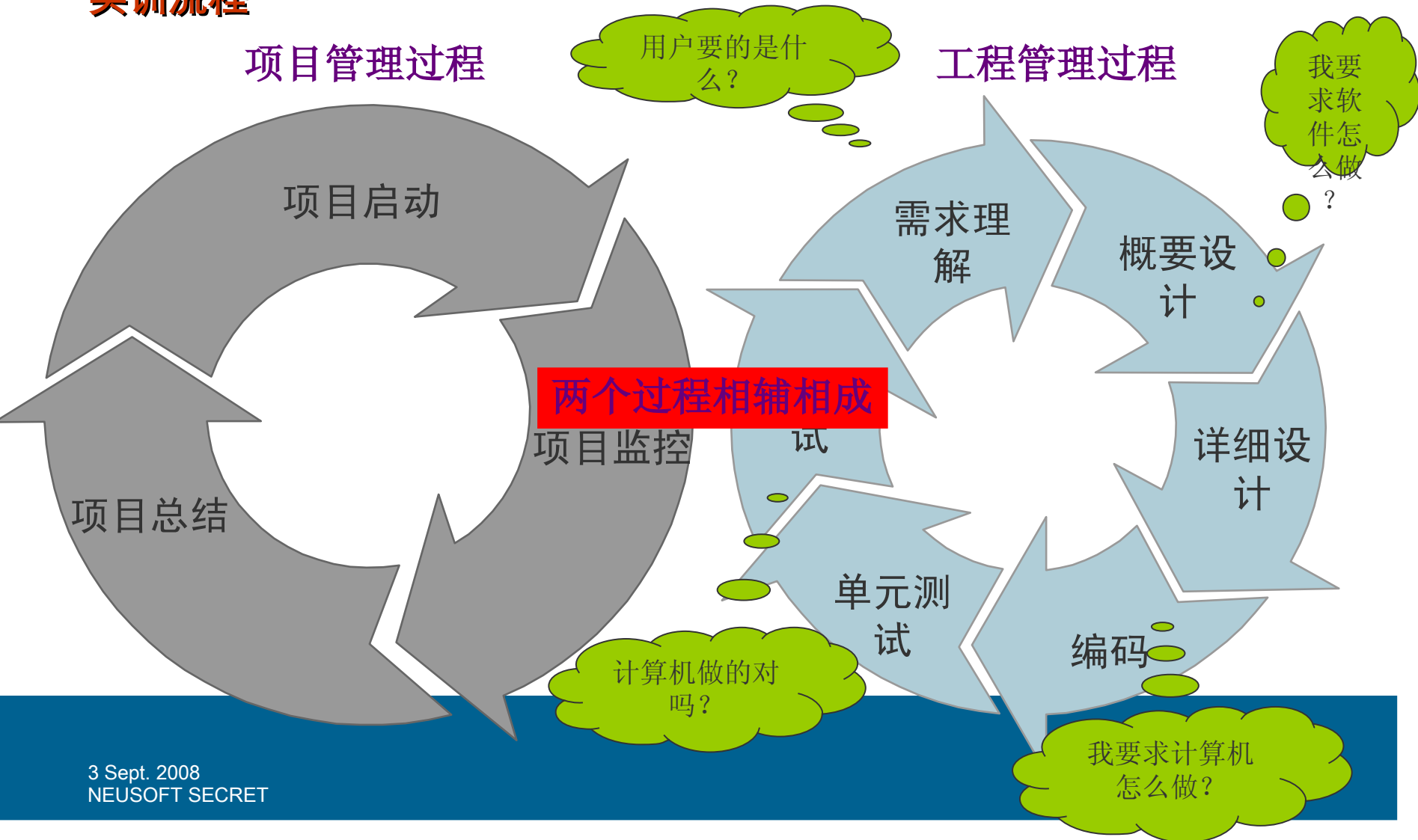
- 学员主动思考，实训经理引导
- 定期评审
- 定期授课

实训考核

- 多维度
- 多视角

# 整体说明

## 实训流程





## 二、项目策划

# 项目策划

## 项目启动前期

### 主要任务描述

- 1、实训负责人根据项目背景，周期等因素确定本次的实训项目名称、实训经理，并形成《工作陈述》表。
- 2、实训经理根据《工作陈述》表、《项目功能结构》表形成《项目裁剪工作表》，确定本次实训项目的过程和功能模块，由实训负责人审批。
- 3、实训经理根据《工作陈述》《项目裁剪工作表》，制定《项目任务书》。
- 4、实训经理根据《项目任务书》裁剪项目《需求规约》。
- 5、实训经理制定《项目开发计划》、《配置管理计划》、《项目评审计划》。

# 项目策划

## 项目启动

任务描述	<div>1、开班仪式：导师介绍，学员介绍。</div> <div>2、团队组建：确定 PL 等（注：关于团队的详细介绍参见附 1）。</div> <div>3、需求规约、项目任务书讲解。</div> <div>4、项目开发计划、配置管理计划、项目评审计划讲解。（注：因为学生没有过多经验，所以本步骤由实训经理代为编写）。</div> <div>5、项目章程（个人日报，项目周报，每周双周例会，会议纪要，定期评审）等讲解。</div>	
下发文档	<div>1、《项目任务书》</div> <div>3、《项目开发计划》</div> <div>5、《项目评审计划》</div> <div>7、《项目周报》</div>	<div>2、《需求规约》</div> <div>4、《配置管理计划》</div> <div>6、《个人日报》</div> <div>8、《会议纪要》</div>
培训	<div>1、Vss 使用。</div> <div>2、各种下发文档的讲解。</div>	

注意：对于所下发的文档定要严格控制，不可流失，否则会引起法律纠纷，切记 !!

# 项目策划

## 项目启动 - 文档讲

解

### 个人日报

- 1、用来记录、跟踪团队成员的每天日常活动，它记录了工作内容、工作量，以及存在的问题。
- 2、这是一份重要的文档，主要用于开发团队内部的控制，开发的全过程始终发挥作用，直到项目结束的总结。
- 3、团队成员每周使用一个 sheet 。
- 4、这是事务性的记录文档。由团队的每个成员自行填写，并放在 VSS 上，以备查阅。
- 5、由本组 PL 督促检查。

### 项目周报

- 1、用来记录跟踪团队本周的工作内容、进度、人员调度、项目大事纪等项内容，以及所发现的问题。另外一项重要内容就是给出下周的工作计划。
- 2、这份周报主要用于开发团队与实训经理的联系，实训经理一定要仔细查看项目周报，及早发现问题。
- 3、此文件由团队的 PL 负责填写，每周五下班前向实训经理提交，并放在 VSS 上，以备查阅。

### 会议纪要

- 1、用来记录团队的所有会议内容的备忘性文件。
- 2、每次开会前，由 PL 指定的记录人负责将会议记录草稿整理后填写，发送给与会人员，并放在 VSS 上，以备团队成员查阅。
- 3、会议发起人要跟踪会议遗留的尚未解决问题，并督促解决。

### 三、需求理解



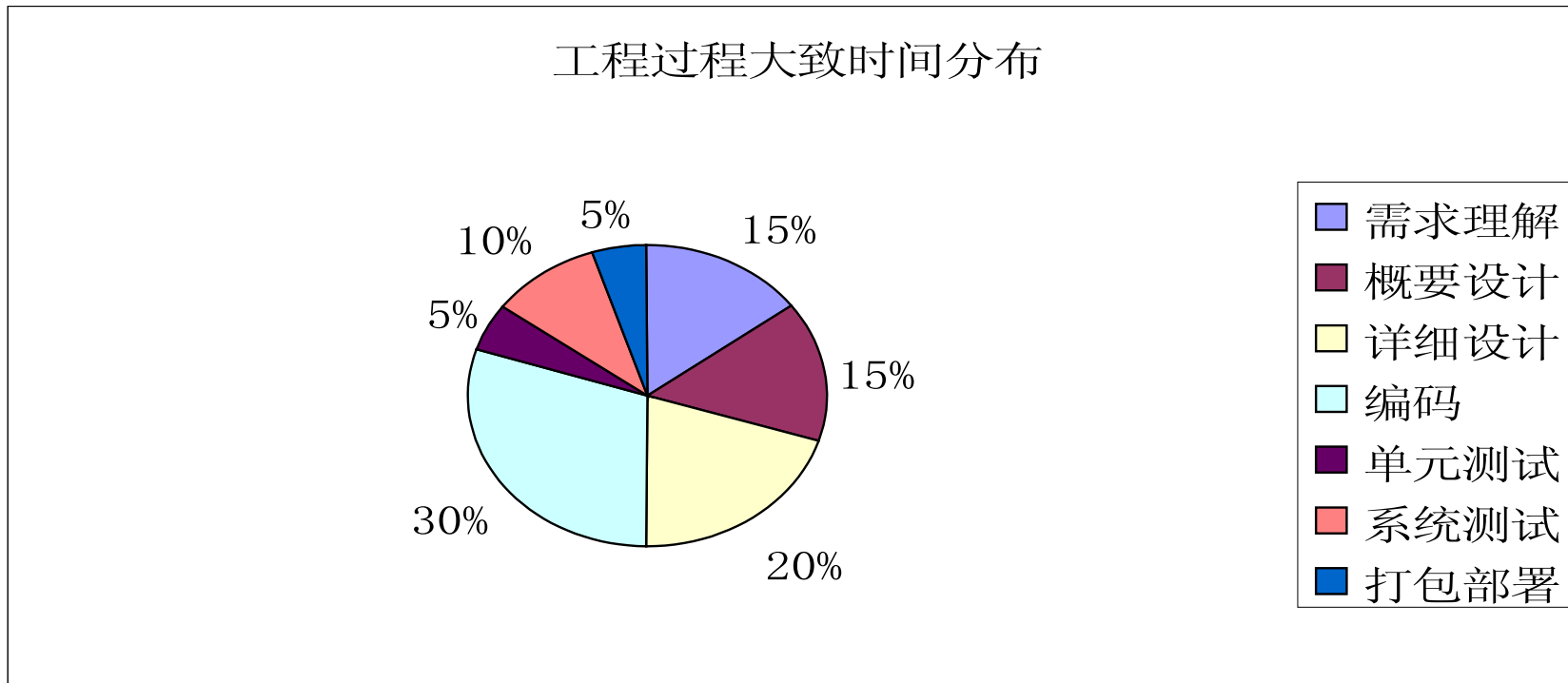
# 需求理解

## 需求理解

- 此时正式进入项目的展开阶段。有道是“一年之计在于春”。一个项目的成败在于所迈出的第一步。项目需求理解就是这第一步，“成败乃在此一举”。
- 前期下发给学员的需求规约只是业务描述，而非技术描述。要将业务需求变成软件产品，中间有巨大的差异，形成一个很大的台阶，切不可一步跨下来，那必将粉身碎骨。但若分解成一个个小台阶，这样走下来，落差很小，则平安无事。需求理解就是这样第一个小台阶（后面还有多级台阶）。经充分的业务理解后，达到技术理解，进而产生概要设计。因此这一步切不可匆忙了事，务求踏实务实。

# 需求理解

## 工程过程时间大致分布



# 需求理解

## 需求理解

任务描述	<ul style="list-style-type: none"><li>1、理解系统需求规约。</li><li>2、理解或做出系统业务流程图。</li></ul>
下发文档	<ul style="list-style-type: none"><li>1、《Q&amp;A 表》。</li><li>2、《同行评审记录》。</li><li>3、《需求规约》。</li></ul>
成果物	<ul style="list-style-type: none"><li>1、系统业务流程图。</li></ul>
培训	<ul style="list-style-type: none"><li>1、使用 Visio 画业务流程图。</li></ul>



# 需求理解

## 需求理解

### 操作步骤

- 1、**人员分工**：PL 将项目大致分成几块（这不是模块），然后主持会议对团队成员分工，原则上采用“个人自告奋勇，PL 统筹平衡”的方式，落实团队成员对“块”的任务分工，形成“块”与成员的相对应关系。
- 2、**仔细研读**：由团队成员对所领受的块仔细研读，认真领会业务，并负责画出自己所负责部分的业务流程图，对不确认的问题记录 QA 表。  
然后由 PRL 召集全体团队成员开会，对每人作出的“理解”进行评审。使正确的理解得以确认，理解错误的予以否决。通过讨论，再补充遗漏的“需求点”。总之，要达到将项目中所含有的全部“需求点”都一一挖掘出来，并记录在案。所谓“全部需求点”是尽力追求的目标。（**注意：此时仅是确定“需求点”及其分布，不要研究如何实现。挖掘务求彻底，切不可匆忙了事。而且要将悬而未决的问题记入 QA 表。PRL 此时展开工作，对本组的讨论结果把关。**）
- 3、**需求整合**：每个成员所负责块的需求全部理清之后，由 PL 组织成员对整个系统的流程整合到一起，体现形式可以任意，只要能表达清楚需求即可，例如为页面原型草稿或者业务流程图细化。
- 4、**需求评审**：实训经理对每组的需求理解进行点评，确认。

### 评审要点

- 1、整体业务流程是否正确，完整？

## 四、概要设计

# 概要设计

## 概要设计

- 这是个重要的承前启后的环节。此阶段的工作力求准确，但切忌过细，只建立起能与需求相吻合的模型即可，以便于团队成员对项目的整体把握。这正体现了“自顶向下，逐步加细”的设计原则。

# 概要设计

## 概要设计

任务描述	<ul style="list-style-type: none"><li>1、在已经充分理解需求的基础上，组内展开讨论，将整个项目划分成“模块”。</li><li>2、确定模块成员。</li><li>3、完成页面流程图。</li><li>4、完成页面原型。</li></ul>
成果物	<ul style="list-style-type: none"><li>1、页面流程图。</li><li>2、页面原型。</li></ul>
下发文档	<ul style="list-style-type: none"><li>1、无</li></ul>
培训	<ul style="list-style-type: none"><li>1、页面流程图画法（课件待开发）。</li><li>2、模块划分方法（课件待开发）。</li></ul>

# 概要设计

## 概要设计

### 操作步骤

- 1、**模块划分**：在充分理解业务需求的基础上，按照实际的业务对项目进行模块划分。  
划分模块原则尽量做到“高内聚，低耦合”。
- 2、**分工微调**：确定项目模块之后，PL可以根据模块任务 / 前期需求理解分工 / 组员状况，对人员分工进行微调。
- 3、**页面整体流程设计**：组员分工确定好之后，由其各自完成本人部分的页面整理流程，体现形式随意。
- 4、**页面原型框架设计 & 培训**：可由组内比较擅长前台页面设计的成员先进行页面框架设计和整体风格设计，其他成员可以进行相应的技术储备。（**注意**：页面整体风格大方，美观，易用是目标，但切记不要过度占用太多时间，以免耽误下阶段工作。），页面框架设计完成之后，请实训经理给出意见，然后由设计人员在全组范围内进行培训。
- 5、**实现页面原型**：各自完成本人负责部分的页面原型。
- 6、**整合 & 评审**：各自负责的页面完成之后，链接到一起，并由实训经理评审。

# 概要设计

## 概要设计

### 评审要点

- 1、页面原型是否符合“业务流程图”，组内页面规范是否一致。
- 2、页面重要元素是否具备（不重要）。

## 五、详细设计

# 详细设计

## 详细设计

- 此阶段工作是在概要设计的基础上进行细化的过程，主要是确定完成概要设计阶段所描述的页面原型功能需要的类以及方法。
- 在细化类时，要细化到类中的成员和方法，确定方法的入口参数，出口参数。
- 另外，还应进行数据库的详细设计。
- **注意：**此项工作是进行编码之前的又一次细化的过程，同时也是发现前期工作中存在问题的阶段，一定要仔细思考，切不可忽略此阶段，草率行事，那样后期更改的工作量成本会更大。



# 详细设计

## 详细设计

任务描述	<ul style="list-style-type: none"><li>1、技术框架培训。</li><li>2、在概要设计的基础上进行设计类及类间的关系，可用类图表示。</li><li>3、细化类中的成员和方法。</li><li>4、描述重要方法的处理逻辑（可用时序图或者流程图表示）。</li><li>5、进行数据库的详细设计，明确需要用到的数据库对象，明确对象之间的关系。</li><li>6、评审详细设计。</li></ul>
成果物	<ul style="list-style-type: none"><li>1、设计报告。</li></ul>
下发文档	<ul style="list-style-type: none"><li>1、设计报告。</li></ul>
培训	<ul style="list-style-type: none"><li>1、技术框架培训。</li><li>2、类图，时序图，流程图画法。</li><li>3、如何进行数据库设计（课件待开发）。</li><li>4、使用 powerdesigner 进行数据库设计。</li><li>5、编码规范（类，方法设计部分）。</li><li>6、设计报告填写。</li></ul>

# 详细设计

## 详细设计

### 操作步骤

- 1、实训经理对项目采用技术框架进行全员培训。
- 2、学员根据框架进行类的设计。
- 3、细化类成员，方法，入口参数，出口参数。
- 4、描述关键方法的流程。
- 5、完成数据库对象设计。
- 6、完成设计文档。
- 7、评审。

### 评审要点

- 1、设计是否实现了“概要设计”设定的类功能要求？
- 2、数据成员的设计是否满足类的功能需求？
- 3、方法成员是否能满足类的功能需求？
- 4、程序中的各种元素的命名是否规范？
- 5、流程图的表述是否清晰、准确、合乎规范？
- 6、数据库设计是否能满足功能的需求？

## 六、编码



# 编码

## 编码

- 到此阶段就可以正式进行编码了，因为前期对系统已经进行了足够的分析设计，因此，本阶段应该相对会比较轻松。
- 这个阶段 PL 要注意团队成员的编码进度，提携有困难的成员，把握项目的整体进度。
- 编码时优先实现别人要用到的方法，以免耽误别人的编码进度，对一时难以实现的功能复杂的方法，要暂时用返回一个期望值的权宜之计处理，过后再慢慢实现。

# 编码

## 编码

任务描述	1、在详细设计的基础上实现类中的方法。
下发文档	1、checkList。（需裁剪）
成果物	1、源代码。
培训	1、高效率编程（课件待开发）。 2、编码规范（方法实现部分）。

# 编码

## 编码

### 操作步骤

- 1、由框架设计人员在 cvs 上搭建项目框架，放入必要的 jar 包。
- 2、团队成员检出项目，负责实现自己的部分。
- 3、使用 checkList 进行代码自查或组内检查。
- 4、代码评审。

#### 注意：

- 1、因为是团队开发，所以一定要保证各种文件的版本正确性，要求大家在修改完文件之后一定要及时提交，修改之前要 update 。切不可把编译有问题的文件提交。
- 2、方法注释一定要完整，包括方法作用，每个入口参数、出口参数说明，对于是集合的对象，要说明集合中的具体对象类型。

### 评审要点

- 1、方法注释是否完整？
- 2、是否符合编码规范？
- 3、方法实现是否优化？
- 4、方法或成员访问权限是否过多？
- 5、类的功能是否过多？

## 七、单元测试

# 单元测试

## 单元测试

- 单元测试是对上一个阶段编写的代码进行的代码级（而非系统级）检验，验证其合格度（而非正确性）。不同于上一个阶段的“代码评审”，那时只看代码书写格式。单体测试则要运行程序，检验功能。编码和测试都是针对代码的，但思路完全不同，编码是千方百计写出正确的代码；测试则是千方百计找出代码中的错误。



# 单元测试

## 单元测试

任务描述	1、编写单元测试类。 2、执行测试类方法。
下发文档	1、无。
成果物	1、单元测试文件。
培训	1、使用工具进行单元测试。
操作步骤	1、编写测试用例类。 2、运行代码，进行逐行跟踪，观察测试点的运行情况。 <b>测试用例类的设计要注意以下几点：</b> 1、程序中所有的分支点都应设为测试点，不能遗漏。 2、测试要分为“正常系”和“异常系”，“异常系”测试点应多于“正常系”，大约 3:2 比例。 3、对变量 / 对象的测试要测其非法值和临界值。 4、方法有多个参数时，应各自设点，不可合为一个。

## 八、系统测试



# 系统测试

## 系统测试

- 与上一个阶段的“单元测试”不同的是，集成测试的侧重点在于程序的宏观整体和实现逻辑，不看微观的内部实现。是站在用户的角度上，从外观界面上检验其功能。
- 分为：1. 功能的逻辑测试； 2. 输入输出操作测试。
- 测试依据：需求规约。
- 测试性质：黑盒测试。
- 指导思想：以外行人的使用可能检验程序，以程序员的专业眼光发现错误。

# 系统测试

## 系统测试

### 任务描述

- 1、根据项目开发计划编写测试计划。
- 2、编写测试用例并评审。
- 3、执行测试并填写测试问题卡。
- 4、修改问题，验证测试问题。
- 5、测试总结。

### 成果物 & 下发文档

- 1、测试计划。
- 2、测试用例。
- 3、测试问题卡。
- 4、测试总结。

### 培训

- 1、各种文档的写法。

# 系统测试

## 系统测试

### 操作步骤

- 1、由实训经理指导 TL 编写根据项目开发计划编写测试计划。
- 2、由 TL 分配组员根据需求规约编写测试用例。
- 3、测试用例评审。
- 4、发布可测试代码。
- 5、根据测试用例执行。
- 6、填写测试问题卡。
- 7、问题确认，修改，再确认。
- 8、测试总结。

# 系统测试

## 系统测试

- 测试用例的设计要注意以下几点：
  - 1、测试用例的设计应描述的详细、准确，杜绝词语含糊概括。
  - 2、“测试点”应对程序中的所有的“功能点”都覆盖到，不可遗漏。
  - 3、应对用户所有可能的输入（包括合法与非法）都设立测点。
  - 4、可以自测，也可以交叉测试（即请别的团队成员参与本组测试）。
  - 5、凡是发现 bug，修改后定要重新全测，不可只测修改的那个局部。

## 九、部署



# 部署

## 部署

- 到此阶段项目已经完成开发和测试任务，本阶段的主要任务是完成应用程序的编译，安装，配置工作。

任务描述	1、编写 build 文件 2、编写软件安装配置报告。
成果物 & 下发文档	1、软件安装配置报告。
培训	1、各种文档的写法。 2、Ant 的使用（课件待开发）
操作步骤	1、编写 build 文件 2、编写软件安装配置报告。



# 十、项目总结

# 项目总结

## 项目总结

- 项目总结是实训项目的“收官”阶段，至此项目已近尾声，需要作的是整理总结。
- 项目总结的前提是：
  - bug list 中的 bug 全部“关闭”；
  - vss/cvs 上的文档、代码都是最终版，且已全部 check in, 它们应被设为“只读”状态。

# 项目总结

## 项目总结

任务描述	<ul style="list-style-type: none"><li>1、每个成员填写个人总结报告。</li><li>2、PL 填写项目总结报告。</li><li>3、PL 指定人员填写归档文件记录。</li><li>4、CML 检查本组实训阶段所有成果物是否为最新版本。</li></ul>
下发文档	<ul style="list-style-type: none"><li>1、个人总结报告。</li><li>2、项目总结报告。</li><li>3、归档文件记录。</li></ul>
成果物	<ul style="list-style-type: none"><li>1、个人总结报告。</li><li>2、项目总结报告。</li><li>3、归档文件记录。</li></ul>
培训	<ul style="list-style-type: none"><li>1、各种文档的写法。</li></ul>

# 十一、项目答辩



# 项目答辩

## 项目答辩

- 该阶段主要用来检查每组实训前的制定的目标是否达成。
- 另一层目的分享实训经理带项目的经验。
- 目前答辩时主要考核的维度及权重参见《项目答辩记录》表，每次实训项目可调整，但在实训之前应确定具体的考核指标。
- 通常项目答辩要由多个评委参加。

# 附 1 - 团队介绍

## 团队介绍

- 团队是按项目开发需要组成的人员集体，其规模随项目的需要而定。
- 若只是把人们简单地凑在一起，各做各事，那叫团伙，顶多叫集体。
- 只有成员们目标一致、相互了解、相互信任、相互支持、通力合作、同心协力、默契配合、荣辱与共的命运共同体才真正称得上团队。
- 一个团队应有核心领导，他们负责掌控团队的各阶段活动，建立并强化团队的凝聚力，及时发现问题并予以纠正，指挥协调团队成员的工作。



# 附 1 - 团队介绍

## 团队介绍 (续)

**PL**  
(项目负责人)

- 1、带领团队按开发流程和进度要求，保证质量地完成每个步骤，最终完成整个项目的开发。
- 2、及时发现问题，调配组内力量，主持会议，检查日报，撰写周报，汇集项目文档，督促团队成员的工作，还要及时与实训经理沟通。
- 3、PL 相当于一艘船的船长，他在组内应拥有足够的管理权威，令行禁止，以带领团队成员步调一致地展开活动。
- 4、应由团队中具有管理能力有威信者担任。

**PRL**  
(评审负责人)

- 1、在团队开发的各里程碑，对相关文档审查把关，具体承担保证各阶段开发质量的责任。
- 2、组织本组各阶段的评审，并对评审结果把关，以及修改结果的跟踪。
- 3、PRL 象舰船上的舵手，负责协助 PL 把握方向，修正错误。
- 4、应由团队中有较高技术能力者担任。

# 附 1 - 团队介绍

## 团队介绍 (续)

### CML (配置管理)

- 1、按照既定的规范，使用开发管理工具 VSS/ CVS 对团队成员的开发流程和阶段成果物进行技术支持服务，保证整个项目的开发秩序。
- 2、及时督促检查每个人的成果物更新。
- 3、在编码开始以后的各阶段。尤其重要的是对代码版本的管理。
- 4、应由团队中认真仔细，有强烈责任感者担任。

### PE (团队成员)

- 1、在团队负责人的领导下，运用自己的知识，发挥个人才华，按照既定的规范，及时完成文档书写、代码编写以及调试等份内的工作。
- 2、积极与团队的其他成员沟通，团结合作，参加各阶段评审和各种会议，并发表意见。
- 3、积极学习各种新知识，新技术。



# Neusoft

Beyond Technology

Copyright © 2008 Neusoft Corporation