



上海财经大学

SHANGHAI UNIVERSITY OF FINANCE AND ECONOMICS

系统设计、实施、运行和维护

饶艳超 上海财经大学会计学院

raoyanchao@qq.com





学习目标



上海财经大学
SHANGHAI UNIVERSITY OF FINANCE AND ECONOMICS

了解并掌握

- 概要设计的主要任务
- 概要设计说明书的主要内容
- 详细设计的主要任务
- 详细设计说明书的主要内容
- 系统实施阶段的主要任务和相应的文档资料内容
- 系统运行和维护阶段的主要任务和相应的文档资料内容
- 系统文档资料的建立



- 怎样将用户的具体要求转化为一个真正的电算化系统，这就是系统设计阶段所要解决的问题，也就是“**怎么做**”的问题。
- 系统开发人员必须根据要开发的信息系统的逻辑模型来选择所需要的硬件和软件设备。只有确定了新系统所采用的硬件、软件设备之后，才能进行进一步的设计工作。



- 系统设计
 - 系统设计分为总体设计和详细设计两部分。
 - 总体设计又叫概要设计，主要是完成模块分解，确定系统的模块层次结构。
 - 详细设计则涉及到系统的代码、数据库文件、输入、输出等多方面较细致的设计。



- 一、几个概念
- （一）模块
- 一般认为，模块是指一段程序，可以是一个独立的程序，也可能是其中的一段，或几个程序的结合。
- （二）模块间的联系和模块内的联系
- 1、耦合
 - 耦合反映模块间相互联系的紧密程度。
- 2、聚合
 - 聚合反映一个模块内部各个元素彼此结合的紧密程度。



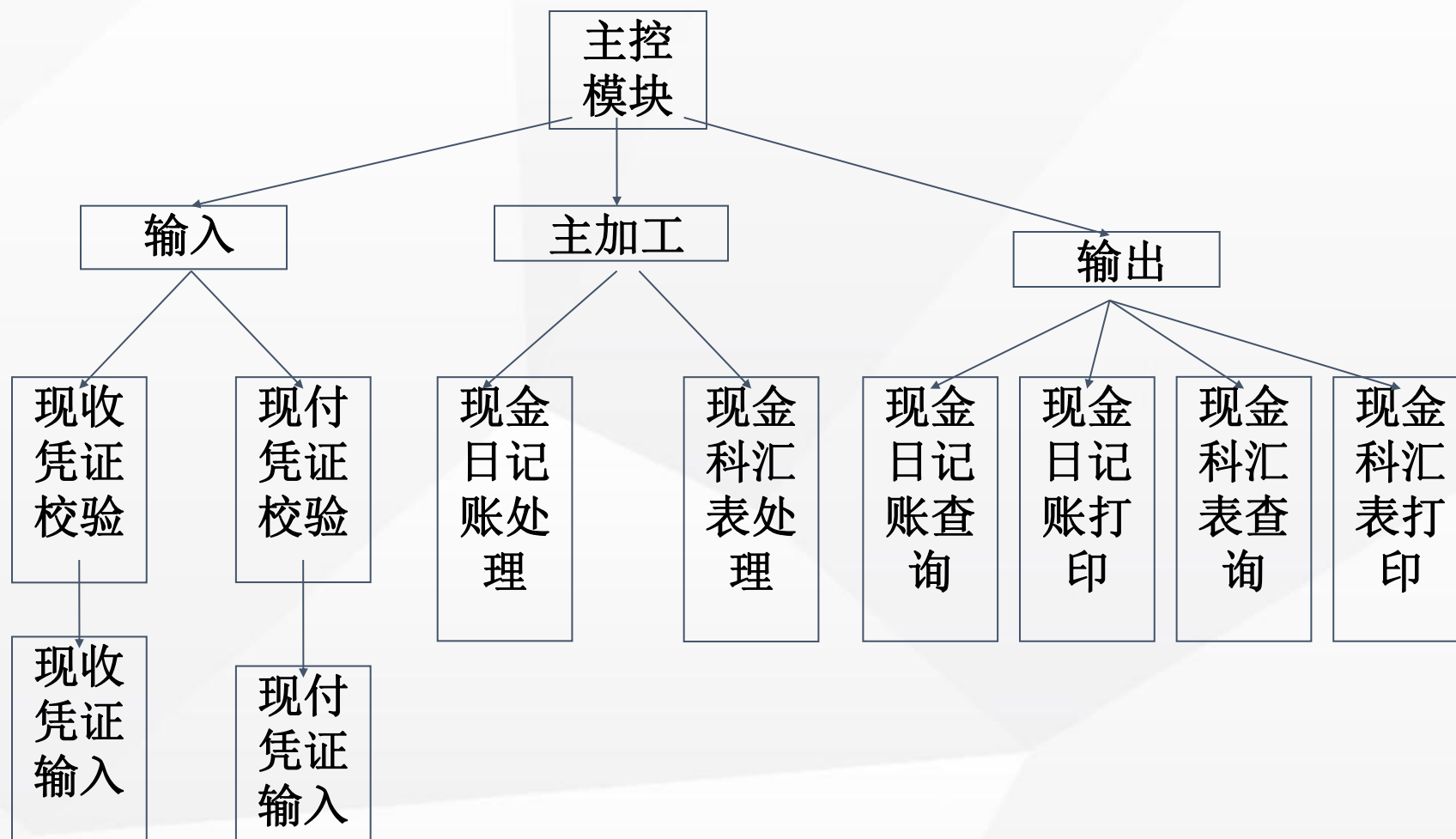
概要设计



- 二、启发式规则
 - （一）程序结构与问题结构相对应
 - （二）功能型的模块的组成
 - （三）消除重复的功能
 - （四）模块的作用范围应该在控制范围之内
 - （五）模块规模应适中
 - （六）深度、宽度、扇出和扇入要适当

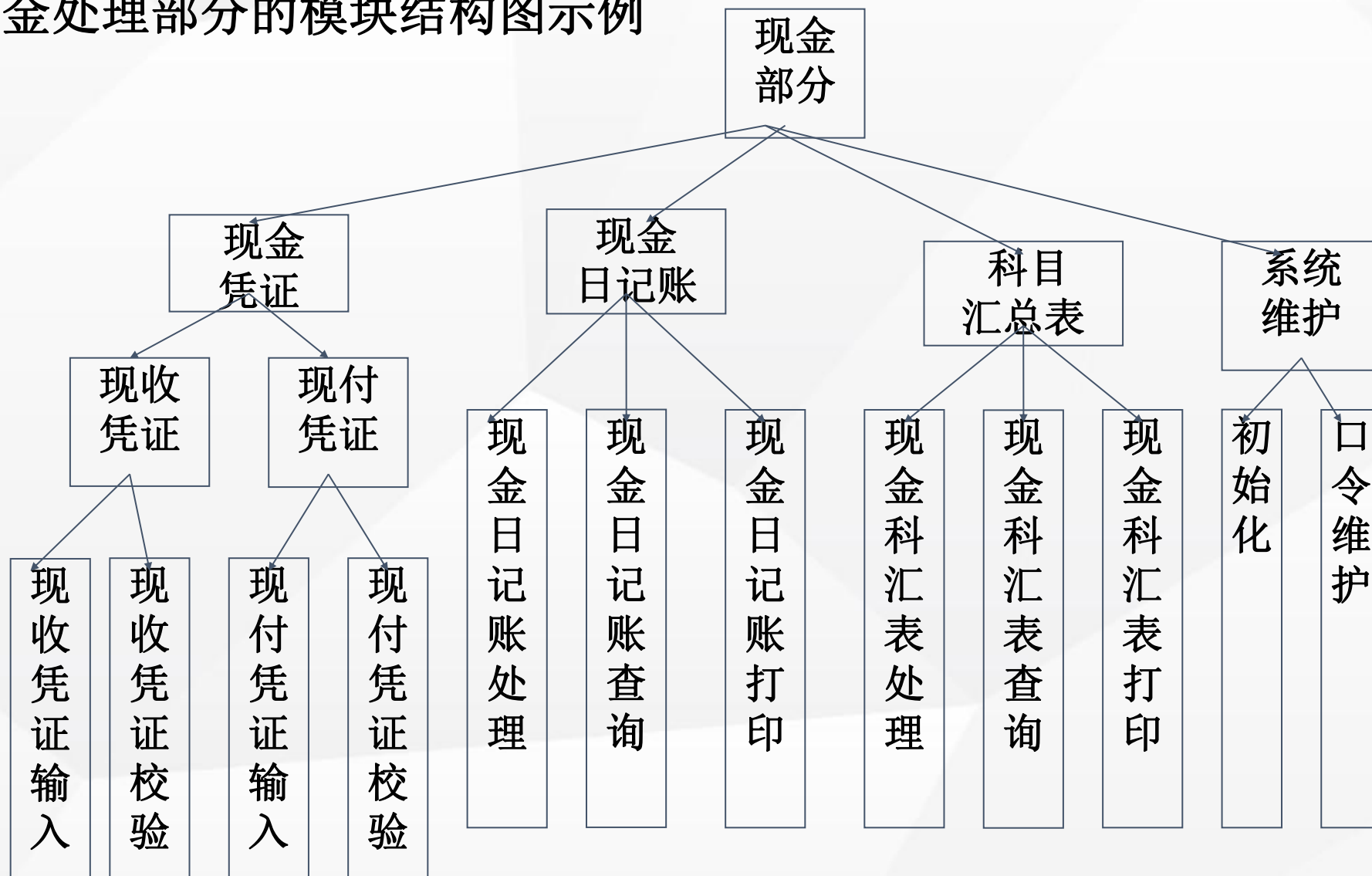


- （三）现金处理部分的模块结构图示例





(三) 现金处理部分的模块结构图示例





- 代码（或称编码）是指用来表示事物的名称、属性、状态等的符号或记号。
- 一、代码设计的目的
 - （一）为了识别事物（如学号、客户编号等）。
 - （二）为了高效率地进行数据的输入和处理
 - 代码一般较简单，节约输入时间（如科目代码、工号）
 - 编码规则利于计算机进行分类、检索、排序等处理。



- 二、代码类型
 - (一) 数字型代码
 - 1、连续的数字型代码 (员工编号001, 002, 003)
 - 2、分离的数字型代码 (分公司编号: 1-9, 10-19)
 - 3、组合的数字型代码 (分公司+员工: 10001, 10002)
 - (二) 字母型代码
 - (A-会计部, H-人力资源部)
 - (三) 混合型代码
 - (A001-会计部员工, H001-人力资源部员工)
 - (四) 层次型代码
 - (图书书目编码系统, 主要分类.次要分类, F.b01)
 - (五) 助记型代码



• 三、代码设计的原则

- （一）代码的标准性和统一性
 - 如财政部统一规定的一级会计科目代码，企业必须统一遵循该编码体系。
- （二）代码的唯一性
 - 一个代码只能标志一个事物，如一个会计科目代码只能对应一个会计科目。
- （三）代码的精简性
- （四）代码的易记性
- （五）代码的可扩充性
- （六）代码的稳定性
- （七）代码的易于校验



• 四、代码设计的步骤

- （一）调查、分析需代码化的对象
 - 客户
 - 供应商
 - 存货
 -
- （二）研究代码化对象的特征
 - 数量、区域、规模、重要性.....
- （三）编制代码设计说明书



- 系统数据库文件设计

- 一、需要建立的文件

- 主文件

- 反映数据项经过加以数据修改后当前状态的文件。

- 事务文件

- 存放日常交易数据，用于更新主文件。

- 参照文件

- 用于存储各种代码信息，并对代码进行解释。



- 系统数据库文件设计

- 一、需要建立的文件

- 控制文件

- 用于存储信息系统中各种参数设置的表文件，如账套参数表、人员权限表。

- 日志文件

- 用于记录任务处理或程序执行期间所使用的命令、执行日期、时间等资料的文件。



- 系统数据库文件设计

- 二、数据库文件结构设计的考虑因素

- (一) 数据的可扩充性
 - (二) 数据处理的效率
 - (三) 数据的一致性及完整性
 - (四) 数据的安全性



- 系统数据库文件设计

- 三、数据库文件设计步骤

- （一）用户需求分析

- DFD

- （二）建立概念数据模型

- ERD

- （三）逻辑设计

- 将ERD转化为子模式

- （四）物理设计

- 内模式：文件结构、各种存储路径、存储空间的分配、记录的存储格式等等。



- 系统输入设计
 - 一、输入方式的选择
 - 类型
 - 联机输入
 - 脱机输入
 - 特殊设备输入
 - 选择输入方式需要考虑的因素：
 - 速度
 - 准确性
 - 成本
 - 数据量
 - 可靠性
 - 灵活性



- 系统输入设计
 - 二、输入设计的内容
 - 确定输入数据的内容
 - 确定数据的输入方式
 - 确定数据的输入格式



- 系统输入设计
- 例：输入格式的设计
 - 输入格式的设计首先要考虑到用户的要求。另一方面，还必须考虑现有设备、也就是企业拥有的计算机显示器的条件。
 - 具体来说，在账务处理中的输入主要指记账凭证的输入。

转账凭证 年 月 日

摘 要	借方科目代码	贷方科目代码	金 额
合 计			



- 系统输入设计

- 三、输入设计的原则

- （一）操作方便
 - （二）复核、修改方便
 - （三）输入响应迅速

- 四、需要重点关注的因素

- （一）将输入数据进行分类
 - （二）解决同一数据多重入口问题
 - （三）输入数据要具有统一性、规范性
 - （四）输入数据要精简



- 输出设计
 - 一、输出方式的选择
 - 输出方式直接影响着会计数据输出的质量，在设计时应根据各种输出方式的特点、适用范围等因素来考虑。
 - （一）显示输出
 - （二）打印输出
 - （三）磁化输出
 - 选择输出设备应该考虑的因素
 - 保存要求
 - 数据量
 - 速度
 - 再处理要求
 - 成本



- 输出设计

- 二、输出设计的内容

- 确定输出内容
 - 确定输出设备
 - 确定输出格式



- 输出设计

- 三、输出设计的原则

- （一）输出格式尽可能符合用户的需要。
 - （二）无论是打印输出还是显示输出，尽可能以物理规格为一单元，不致显得支离破碎。
 - （三）格式除了实用，还须考虑标准化设计，这样，可以用较少的模块（公共子程序）适应较多的输出格式。
 - （四）格式的设计要考虑系统发展的需要，在设计时，一般都在原格式中留有备用项目和可扩展数据长度。



- 概要设计说明书编制指南
 - 概要设计说明书又可称系统设计说明书，这里所说的系统是指程序系统。
 - 编制的目的是说明对程序系统的设计考虑，包括程序系统的基本处理流程、程序系统的组织结构、模块划分、功能分配、接口设计、运行设计、数据结构设计和出错处理设计等，为程序的详细设计提供基础。
 - 参见计算机软件产品开发文件编制指南 GB 8567-88



详细设计说明书



上海财经大学
SHANGHAI UNIVERSITY OF FINANCE AND ECONOMICS

- 详细设计是对总体设计中划分的每个模块再进行详细定义和说明。
- 它包括定义每一模块的详细功能、输入数据、使用文件及使用方式，确定输出内容及格式，模块实现的详细算法，每一模块的程序构成等。详细设计是系统功能、结构实现方法的最详细说明，是程序设计的依据。
- 详细设计说明书又可称程序设计说明书，编制目的是说明一个软件系统各个层次中的每一个程序（每个模块或子程序）的设计考虑。
- 如果一个软件系统比较简单，层次很少，本文件可以不单独编写，有关内容合并入概要设计说明书。



系统实施



上海财经大学
SHANGHAI UNIVERSITY OF FINANCE AND ECONOMICS

- 系统实施就是要将理论设计变化成一个实际的、可操作的系统。
- 具体工作包括：硬件准备、软件准备、人员准备、数据准备。



- 硬件准备
- 硬件准备指物理系统的实施，是计算机系统和通信网络系统设备的订购、机房的准备和设备的安装调试等一系列活动的总和。
- 购置计算机系统应该考虑系统的设计要求、计算机系统的性价比、可扩充性、售后服务和技术支持等方面。
- 网络系统的实施主要是通信设备的安装、电缆线的铺设及网络性能的调试等工作。



- 软件准备

- 系统实施阶段的一项主要任务是程序设计即代码编写。
- 结构化程序设计是一种设计程序的技术，它采用自顶向下逐步求精的设计方法和单入口单出口的控制结构。
- 在结构化程序设计中，基本的控制结构是“顺序”、“选择”和“循环”。
- 实践证明，结构化程序设计技术可以显著提高软件开发工程的成功率和生产率，程序容易阅读和理解，容易诊断和修改。



系统实施



上海财经大学
SHANGHAI UNIVERSITY OF FINANCE AND ECONOMICS

- 编写源程序代码
 - 根据用户和系统的要求，选定一种编程语言，用这种程序设计语言将设计的结果表达出来。
 - 程序设计自动化。



- 测试
- (一) 单体测试
 - 对模块的测试主要侧重于语法检查和逻辑检查。
- (二) 总体测试
 - 总体测试着重于测试系统中各模块及组成的子系统之间接口关系的正确性，系统逻辑关系的正确性，以保证数据传送及调用关系的正确性。



- 测试
- (三) 系统测试
 - 把已经过确认的软件纳入实际运行环境中，与其他系统成分（计算机硬件、外设、某些支持软件、数据和人员等）组合在一起，对计算机系统进行的测试。
 - 1、 α 测试
 - 是用户到软件开发的场所来进行的测试，是在软件开发人员的指导下进行的，开发人员负责记录测试发现的问题，所以 α 测试是在一个受控环境下的测试。



- 测试
- (三) 系统测试
 - 2、 β 测试
 - 是由软件的多个用户在实际使用环境下进行的测试。这些用户返回有关错误信息给开发者。测试时，开发者通常不在测试现场，因而， β 测试是在开发者无法控制的环境下进行的软件现场应用。在 β 测试中，由用户记下遇到的所有问题，包括真实的以及主观认定的问题，定期向开发者报告。



- 调试
 - 测试的目的是尽可能多地暴露程序中的错误，发现错误的最终目的是改正错误。
 - 进一步诊断和改正错误，这就是调试的任务。
 - 试探法：分析错误征兆，猜想故障的大致位置，在程序中加进许多显示语句，获取程序中被怀疑的地方的信息。这种方法效率较低。



- 一、文档的建立

- （一）开发性文档

- 与系统的开发和维护有密切关系的文档资料。如
 - 数据流图、模块结构图、系统需求说明、设计说明书等
- 开发性文档应该在信息系统的建立过程中同步生成，并且随着信息系统的修改进行及时修正。

- （二）使用性文档

- 指用户手册（非技术性）、操作手册等用于指导对信息系统的操作使用的说明性资料。
- 在信息系统交付用户的同时，也需要提供使用性文档。



• 二、员工的培训

• （一）培训的重要性

- 当企业建立或者开发了一个新的信息系统时，或者当企业对现有系统进行重大修改以后，或者当企业聘用了新的员工时，或者当员工的工作在企业内部调整时，等等，在这些情况下，都表明企业需要开始员工的培训工作了。

• （二）怎样进行培训

- 如果能够在信息系统建立时，尽可能早地将员工结合进来，使得员工对将要发生的变化有所了解 and 准备，可能会促进员工对系统的理解和支持。
- 具体的培训方法可以采用课堂培训、电脑培训、案例研究的方法、阅读参考资料等手段。
- 对不同管理等级的员工进行培训时的具体内容也有所不同。



• 三、数据文件的转换

- 在信息系统正式运行之前，需要将企业已有的数据文件转换为信息系统可以读取的方式。这也是我们熟悉的初始化工作。
- 在进行数据文件转换之前，首先要检查和确认原有的数据文件是否是完整的、准确的。
- 转换前后要对数据进行检查，保证转换过程没有导致数据丢失、出错。



• 四、新旧系统的交接

- 应该保证新旧信息系统的平稳过渡，尽量减少对业务处理的影响。
- 直接转换
- 并行转换
- 分阶段转换
- 试点转换



- 一、系统正式运行阶段的主要工作
 - （一）进行会计核算软件要求的初始化设置
 - （二）日常的输入工作，特别是凭证的输入工作
 - （三）输入数据的复核和修改
 - （四）会计数据的处理
 - （五）会计信息的输出
 - （六）报表的编制和输出
 - （七）尽可能利用软件提供的功能



- 二、正式运行阶段应注意的问题
 - （一）工作程序问题
 - （二）输入的准确性
 - （三）会计数据的安全



- 会计信息系统系统移交给用户使用以后，可能会出现一些问题需要解决，这就是维护工作。
- 一、会计信息系统维护的原因
 - （一）会计软件开发后期的测试阶段未能发现的缺陷，包括编程技术方面的某些错误和对用户需求理解方面的某些偏差，在软件交付使用后逐渐暴露出来。
 - （二）会计软件用户环境的改变。例如，操作系统版本的提高，硬件设备的更新，由单机环境改为网络环境等。适应原有环境的软件可能无法在新环境下有效工作。



• 一、会计信息系统维护的原因

- （三）由于用户知识技能水平的提高，而对原有软件的用户界面不太满意，要求得到改善。
- （四）随着使用单位的管理信息系统的建立或完善，要求会计软件作为一个子系统与其他子系统有良好的接口，以便协同工作。
- （五）使用单位经营状况的改变，导致会计人员需要处理一些新的经济业务或者淘汰一些不再经营的经济业务，这时需要对会计软件改造设计，完善功能。



- 二、会计信息系统维护工作的内容
 - （一）改正性维护
 - 发现程序错误，进行诊断和改正错误。
 - （二）适应性维护
 - 为了和变化了的环境适当地配合而修改软件。
 - （三）完善性维护
 - 为了扩充会计信息系统的功能，提高原有性能而进行的修改。
 - （四）预防性维护
 - 为了给未来的改进奠定更好的基础而修改软件。



• 三、提高系统的可维护性

- 所谓软件的“可维护性”，通常包括：对软件易阅读、易发现和纠正错误、易修改和扩充等含义。
- 软件的可维护性能越好，软件的维护工作就可以得到简化和减轻。
- 支配软件工程方法论所有步骤的关键目标就是提高软件的可维护性。
- 决定软件可维护性的因素主要就是软件的可理解性、可测试性和可修改性。



- 三、提高系统的可维护性
 - （一）软件“可维护性”的重要性
 - 软件维护阶段的花费要占整个软件生命周期花费的70%左右。
 - 软件维护工作是相当繁重和困难的。如果没有成熟、规范化的软件维护，系统可能瘫痪从而前功尽弃。
 - 随着软件规模的扩大和复杂性的增加，怎样改进软件维护的问题已显得越来越重要。



- 三、提高系统的可维护性
- (二) 影响软件“可维护性”的因素
 - 1、程序及与其相关的文件所用的记法、记号及与内容是否协调一致。
 - 2、能否较容易地选择和利用软件所具有的功能和设施。
 - 3、对软件的输入和输出信息是否灵活方便。
 - 4、程序结构是否符合结构化设计的要求。
 - 5、程序内的解释、符号名的命名是否能清楚地表明它的功能、结构、使用及输入输出信息等特性。
 - 6、提供的信息是否都是必要的，而无多余信息。
 - 7、程序编写得是否具有良好的风格，是否便于阅读和理解。
 - 8、是否为进一步扩充创造了有利的条件。



- 三、提高系统的可维护性
- (三) 提高软件“可维护性”的方法
 - 1、对软件需求分析的可维护性考虑
 - (1) 系统结构上的可扩充性
 - (2) 出错处理的考虑
 - 2、对系统设计的可维护性的考虑
 - (1) 对功能模块划分的要求
 - 应尽可能强调模块的低耦合、高内聚，提高模块的独立性
 - (2) 对设计规范化的要求
 - 规范化考虑的主要内容有：命名规范化、数据类型统一化、数据代码标准化等等。



- 三、提高系统的可维护性
- (三) 提高软件“可维护性”的方法
 - 3、对程序设计的可维护性的考虑
 - 必须要求源程序的逻辑、语法正确，还要求程序具有清晰的结构，有较好的可读性和易修改性。程序设计应考虑采用结构化程序设计方法、参数使用文件化、程序的设计风格易于理解等。
 - 4、对文档资料的考虑
 - 这些文档资料必须是统一的，是和程序代码完全一致的。当然，维护过程中也要保留有完善的文档。



系统评价



- 在会计信息系统的整个存续期内，都需要对软件进行评估和监控，及时了解软件的实际运行状况。
- 系统评价工作的目的是通过对新系统运行过程和绩效的审查，来检查新系统是否达到了预期目的，是否充分地利用了系统内各种资源(包括计算机硬件资源、软件资源和数据资源)，系统的管理工作是否完善，以及指出系统改进和扩展的方向是什么等。
- 系统评价的主要指标



- 阅读材料：打造“现代神农”——神农K3系统实施案例
 - 神农公司信息化前面临的问题有哪些？
 - 神农公司信息化的目标和实施思路是什么？
 - 神农公司信息化实施效果分析？



上海财经大学
SHANGHAI UNIVERSITY OF FINANCE AND ECONOMICS

THANK YOU

