

系统开发过程

□ 五个阶段

各种系统开发方法学在范围、复杂性、完善程度以及方法上有很大的不同。尽管有的方法学分三个阶段，有的分 15 个阶段，但是每个方法学所描述的要完成的活动基本上是相同的。本章要阐述的最重要的一点是：最好的方法学是那些始终把用户考虑进去的方法学。过去的情况是，用户管理人员与信息服务开发组合作来完成系统的一般功能说明书，然后，由信息服务人员来进行系统开发。现在，系统开发是各占 50% 的比例；因此，用户管理人员应该非常熟悉系统开发的大体过程，特别应该熟悉他们单位自己使用的方法学。

系统开发过程可分为五个阶段来描述。这五个阶段是：

1. 第 I 阶段—系统开始和可行性研究
2. 第 II 阶段—系统分析和设计
3. 第 III 阶段—程序设计
4. 第 IV 阶段—转换和实现
5. 第 V 阶段—实现后的评价

第 I 阶段—系统开始和可行性研究是在为开发一个建议的系统提供人力和资源之前完成的。第 I 阶段多数的工作和编写的资料是第 II 阶段的输入。在第 II 阶段—系统分析和设计期间，系统分析员与用户一起工作以编写详细的功能和系统的说明书。将这些说明书交给程序员，然后开始第 III 阶段—程序设计。在第 IV 阶段—转换和实现期间，一旦软件开发出来，则建立数据文件，转换现有系统，并且实现新系统。第 V 阶段—实现后的评价。在开始了系统寿命期中的生产阶段之后，提出(经常被忽略的)实现后的评价要求。

□ 具体开发过程

下面将逐步地描述系统开发过程。至于具体的细节、相互的影响、方法、形式等，用户管理人员应该与信息服务经理联系，与他们讨论公司当前使用的方法学，同时再看看公司内部描述方法学的手册。

1. 第 I 阶段—系统开始和可行性研究

在第 I 阶段的活动中很少有与其他四个阶段的活动相一致的。此处所提供的方法包括对于受拒绝后的再次服务请求的方法以及将技术转移可能性的研究合并到诸过程中这些内容。第 I 阶段最终的产品有两个部分。第一部分是实际的可行性研究报告，它包含对建议的或改进的系统的描述以及利润/成本分析。第二部分是系统的初步设计。它对于估价成本和利润是必要的。该初步设计是第 II 阶段—系统分析和设计的直接输入。

将系统的初步设计并入可行性研究的依据是，多数可行性研究是以概念而不是以设计为基础的。如果在描述系统目标上花的时间太少，那么成本估计，甚至利润估计将是错误的。用概念来指导可行性研究注定会导致成本过高，而且用户不满意。在系统初步设计上所花费的时间是值得的，即使拒绝可行性研究也是如此。因为所编写的资料将必然会被证实其他项目中是有价值的。

下述编号的活动与表 20.9.2 的系统开发责任矩阵相对应。

- (1) 提交服务请求

图 20.5.1 说明了包括对受拒绝的请求再次请求处理的一种方法。所请求的服务毕竟是由用户做的，因此，应该由用户着手进行。我们鼓励用户管理人员请求信息服务人员的帮助，但是应该再一次强调，业务领域的管理人员应该对各种大小的服务请求都提供合适的资料。

- (2) 估价服务请求

正如在责任矩阵中所注释的那样，信息服务管理人员只能承诺小的项目(由公司的方针所确定的小项目)。

- (3) 指定可行性研究组

信息服务经理和用户经理共同来指定适当的混合的人选以组成可行性分析研究组。该组至少由一名系统分析员和一名用户代表组成。可行性研究组的大小取决于可行性研究的范围和时间限制。

用户代表应该熟悉当前专业领域的所有工作，用户经理、总经理助理，或专业领域分析员是合理的候选者，用户的系统分析员，具有计算机信息处理基础知识的情况已经越来越普

遍了。

必须指定一个人担任可行性研究组的组长,哪怕只是两个人的可行性研究组也需要一个组长。直到 1980 年为止,多数的可行性研究组和项目组是由一个高级系统分析员或一个项目负责人来领导的。在信息服务部门中,这两种人是固定分工做这项工作的。目前越来越多的公司采取这样一种政策,即由用户担任项目组组长。这种将主要责任下放给最终用户的做法将进一步鼓励用户参与系统设计。在这种政策上取得成功经验的那些公司已经指派了一些具有杰出管理经验和具有某些计算机和信息处理知识的用户人员担任项目组组长。在任何情况下,组长必须对该组的工作有一个总的安排。如果要求一个用户代表既作为可行性研究组或项目组的组长而同时又要求他继续履行业务领域的职责,那么该项目是肯定要失败的。有好些公司已经采用了一种政策,即自动地指派受系统影响最大的业务领域的经理作为可行性研究组和项目组的领导以后该经理将从原来的工作职责中解脱出来,而用他(她)的全部时间管理可行性研究(或项目)组。这种人事安排已经成为当今的主流,其困难是用户经理需要离开原来主管的业务部门少则两个月多则三年后才能回他原来的工作岗位上。

(4) 标列约束条件

在系统开发的过程一开始,可行性研究组与信息服务人员和用户经理密切合作标列出设备、成本、进度、规程、软件以及操作上的约束条件。它们可能限制建议的系统的定义和设计。

(5) 整理现有系统的资料

整理现有系统资料的主要理由是:如果可行性研究组不了解现有系统,那么他们就不可能有效地完成所建议的系统的初始设计。已经建立起来的多数人工系统并没有经过真正的设计。在这些系统中,必须从手稿整理出资料。如果一个建议的系统是改进一个现有的计算机信息系统,那么可行性研究组只需要保证现有资料的完整性和保持最新版本就行了。

现有系统所形成的任何资料将给设计阶段提供有价值的输入(如果批准开发该系统)。即便建议的系统遭到拒绝,也能对现有系统提供基本的资料,并且可能透彻地理解现有系统。现有系统的资料由四部分组成:①系统报告和资料;②系统数据文件;③系统数据元以及④说明现有系统的数据、信息和工作流程的图表。前三部分(报告、文件和数据元)可分类如下:

- ①当前使用的,而且在建议的系统中以目前的形式保留下来;
- ②当前使用的,但是修改后才在建议的系统中使用;
- ③当前使用的,但是在建议的系统中将被删除而不再保留的。

例如,列出所有现有的报告和标准的资料,并按上述分类给定一种状态。在报告上将标明相对周期(如,每天,每周)以及分发范围。

对于现有系统的所有数据文件都标明有关的存储介质(如,3×5 的卡片,磁带,马尼拉折纸机,磁盘等等)以及存储方式。例如,一个名字—地址文件可以存储在许多张 3×5 的卡片上,并且按名字的字母顺序排列。一个人工系统所保存的文件数总是令人吃惊的,即便对于业务领域管理人员也是如此。为了完善现有文件的资料,将每个文件的记录的样式和简单描述附在文件表中。

系统数据元(即,社会保险号,顾客名,货号等等)是直接列出的,而不必关系有关的文件。数据元经常几个文件中重复出现。除了状态指示符之外,如果数据的名字不能自我说明,则必须对每个数据元进行描述。有关数据元的其他信息还包括更新要求(如,每天,每周,每月,或根据需要更新等等)、来源(如,代办处,资料,系统,工作人员等等)以及职责(如,部门名和负责更新者的职务)。图 20.9.3 说明在整理现有系统资料时数据元可能采用的一种典型格式。

报告标题			系统数据元				日期	
系统标题			医疗信息系统				标识 MIS	
编号	标题	描述	目前的	需要修改	需要删除	更新要求	来源	职责
1	名字	患者	x		x	--	住院表	住院处
2	首字母	"	x		x	--	"	"
3	姓	"	x			AR	"	"
4	子女数	"			x	AR	"	"
5	性别	男或女		x		—	"	"
6	婚姻状况			x		AR	"	"

AR - 当需要时

图 20.9.3 整理现有系统的资料:系统数据元

我们通过将系统简化为输入、处理和输出等几个基本组成部分来表示整理现有系统资料的工作过程。然后用图形描绘出各部分之间的逻辑关系。有多种图像表示技术来做这件事。最为流行的(尽管不一定是最好的)是流程图。其他的更为结构化”的技术还有: IBM 公司的层次化输入-处理-输出图(HIPO), 汽泡图, 数据流框图, 南茜-斯奈德曼(Nassi-Shneiderman)图, 渥尼尔(Warner)框图以及判定表。当前工作过程的图像描述提供了系统的数据、信息和工作流程的一个概貌。它着重强调系统中控制工作流程的那些数据元。这些图应该刻划人工和计算机的处理步骤, 并且以适当的顺序安排每一处理步骤。通常以能最好地显示出工作过程的方式来组织和提供这些图。它们可以由一些随机事件、功能或按小的和大的周期来驱动的系统, 也可以是若干子系统; 既可以是层次的, 也可以是混合的。很少有几个系统是完全顺序的, 因此, 在多数情况下可以应用模块方法。

(6) 调查研究技术转移的可能性

为了更好地利用现有的技术, 许多公司正在进行将有关技术转移到他们的系统开发方法中可能性的调查。鼓励调查技术转移的可能性和(或)可行性的政策必将带来人力资源的大量节省。特别对程序员和分析员更是如此。合适的技术转移将使这些人的工作集中于还没有现成软件的特定行业的应用领域。

技术转移可能性的调查是从走访那些已经实现的, 而且与所建议的系统有类似规模和工作量的系统。可行性研究组还应该调查商品软件目录, 以便找到适合的可应用的软件。如果认为技术转移是可行的, 则可行性研究组说明怎样使用这些技术以及为适应现有环境所要求的修改范围。

如果使用标准的方法来进行技术转移潜力调查, 那么提出要求的公司应该采取与具有类似要求的其他公司合作的政策。

(7) 完成建议系统的初步设计

可行性研究组要走访专业人员以获得一般的系统要求, 然后, 将这些要求转换成初步的系统设计。设计过程是交互的, 用户经理和可行性研究组需要经常就设计思想和方法等交换意见, 用生动的文字和图形说明来形成建议的系统初步设计的资料, 这些生动的文字(用非技术词汇)描述了所建议的系统的基本工作过程, 而且常常同时附有图形说明。这些文字图表也将列举出那些大大违背现有工作方式而建议的系统所期望的手续、手段和方法。这些文字图像也将描述建议的系统与人工系统以及建议系统必须与之兼容的自动系统之间的关系。

图形说明将建议的系统过程简化为它们的组成部分, 同时强调各部分之间的逻辑关系。

(8) 确定项目范围

可行性研究组与信息服务人员以及用户管理人员合作估计初步设计中所刻划的系统的复杂程度。并对开发项目今后的每一个阶段进行人力资源要求的估计(用户, 信息服务人员

及其他人员)。此外，还注意到培训和计算机机时要求。

(9) 准备利润/成本分析报告

一旦完成初步设计并且确定了项目的范围，则可以开始利润/成本分析。不幸的是，由于用户和信息服务管理人员都希望加快可行性研究阶段，所以，一些关键的步骤被省略了，因此造成在利润、成本估计上的错误。仅仅根据一种概念是不可能精确的反映出利润和成本的。设计中的某些步骤是必不可少的。

另一种在形成公司决策过程中所隐含的错误将不可避免地把那些难以确定的利润也算成资金收入。当今许多复杂的、综合的系统为公司的利益做出了重大的贡献，而做到这样程度是因为它们经历了漫长的、不可捉摸和难以预见的道路。评价信息服务项目的好处和价值是一个主观的过程，它要求具有成本和利润方面的实际的知识。此外，决策者对于正的和负的不确定的利润要有透彻的理解。使用美元作为所有成本和利润的统一的计量标准大大地简化了评价工作。那种把不确定的利润引入盈利图表(为了“建立更好的顾客关系”或“提高威信”)的作法会造成在“底线”中复合的错误。底线经常被盲目地接受作为一种信条。事实上，在那种情况下，估价是取最好的情况(理想的)和最坏的(荒谬的)情况之间。然而，如果将不确定的利润化成美元，那么决策者将以更好的判断代替那种不准确的估计。

估价建议的信息系统的最好途径是针对系统净值(收入减去成本)估量正的和负的不确定利润。为了便于理解不确定利润(例如，增加服务，减少发票上的错误，加快周转期等)，应该产生一个成本和收入的一览报表。

表 20.9.4 说明如何使用最少的成本类别来表示一次性的和重复使用的成本。这些成本可由预算中心提出，并且把公司作为一个整体来考虑。成本类别有：劳力，材料和设备，旅差以及其他各种成本。对于每一类，在第一列指出一次性成本估计(开发)，而在系统寿命期的水平线上指出可重复使用的成本估计(生产)。公司项目在净值可以从估计收入中扣除成本计算出来，并且根据公司政策对流动资金打折扣。

表 20.9.4 成本一览表

预算中心			项目标题和编号							
成本项		一次性成本	每年重复使用的成本							
		年	第 1 年	第 2 年	第 3 年	第 4 年	第 5 年	第 6 年	第 7 年	第 8 年
劳 力										
材料和 设备	材料									
	设备									
旅 差	每日开销									
	交通费									
其 他 开 支										
总成本										

(10) 根据可行性研究做出决策

完成可行性研究后，除了技术补充之外所有报告和资料全部交给信息处理政策委员会以便实施。技术补充包括准备可行性研究所要求的背景信息。它还包括一般的系统设计和开始第 II 阶段(系统分析和设计)的一个框架。信息服务政策委员会感兴趣的主要是初始服务请

求、范围、图解说明和利润/成本分析。

信息服务政策委员会能对可行性研究施加影响。信息服务政策委员会能够：

①拒绝建议。

②批准建议并对该建议的开发和实现指定一个最高优先数。

③批准系统并给它指定一个比最高优先数小的优先数，同时将请求放在所有建议的系统队列的适当位置(定期检查队列，当所请求的资源可用时，委员会给当时是最高优先数的项目发出通行命令)。

2. 第Ⅱ阶段—系统分析和设计

很少有几个项目能在批准可行性研究后立即实现。在得到批准和项目开始之间的估计时间可能是两年或两年以上。一旦项目获如通行命令，则开始第Ⅱ阶段—系统分析和设计。在第Ⅱ阶段，将描述所有输入/输出的格式和内容，并且完成详细的系统设计。第Ⅱ阶段的最后一步活动是准备程序说明，其中包括各种程序模块的说明书。重要的是牢记在第Ⅰ阶段和第Ⅱ阶段不编制程序。一个普遍容易犯的错误(经常与系统的质量和运行维护的水平密切相关)是压缩第Ⅱ阶段，使它提前完成以便开始第Ⅲ阶段—程序设计。粗糙的系统设计必将成为、甚至三倍地增长项目所要求的程序设计量。

(11) 指定项目组

与可行性研究组一样，项目组也应该有一个或多个系统分析员和一至多个来自所建议的系统范围内各业务方面的用户代表。如果可能的话，还要给项目组指派一名信息服务审计员，他不作为专职人员，而作为安全和控制方面的顾问。因为在第Ⅱ阶段结束之前程序员实际上并不参与进来，所以可以将指定程序员一事推迟到第Ⅱ阶段结束时再进行。可行性研究组的成员不一定是项目组成员。在第Ⅰ阶段结束到第Ⅱ阶段开始之间的这一段时间里，通常委派他们到其他项目去。然而我们建议，只要可能则尽量将原有可行性研究组的人员指派到项目组。项目组的组长可以是信息服务人员，也可以是用户。

某些单位有按业务领域组织的固定的项目组。例如，某个项目组专门负责人力资源开发方面的老的系统的维护和新系统的开发，而另一项目组则负责会计和财务方面等等。另一种办法是项目组必须由信息服务人员和用户专业人员共同组成，而且是以项目为基础来指定项目组。究竟怎样组成项目组为好，显然要进行权衡。按专业组成的项目组很难预料在任务过多时或任务不足时由于人员不足或过剩所带来的损失。然而，这种项目组织使得项目组成员有更多的机会积累开发专业领域应用的经验。信息服务项目组织的最好方式或许是既按专业领域组织而同时又保持一定的灵活性，使得项目组成员能在各项目组织之间流动，以便达到饱满的工作负荷。

根据项目的复杂程度和涉及范围的大小，每个项目组都有不同的最佳人数。项目组长的能力是一个重要的因素。有些地方，一个经理能有效地管理 20 个以上的人员，而另一些经理却连管理 3 个人都有困难。项目组的大小以及相对进度这些是用户、信息服务人员以及公司的经理感兴趣的问题。许多公司的经理人员有一种错误的概念，即如果将项目组人员增加一倍，那么完成项目的时间就应该减少一半。实际情况并非如此。一个能够直接分成若干个相同大小模块的简单项目，用两倍的人力，可以在原定的一半时间里实现。然而，绝大多数的项目是复杂的，有的甚至是极为复杂的，这就要求在所有项目组成员间进行内部协调。

图 20.9.5 说明增大项目组的规模时，将会发生的情况。在某确定的数目之前，每增加一个指派到项目组的人员都增大了对项目的贡献。在这之后，每增加一个人实际上减少了项目组每个人对项目工作的贡献。图上有一点是增人员的反射界线，超过那一点，再增加人对于项目的目标来说反而起相反作用。由于项目成员之间的关系复杂，因而使得生产效率降低。在为了满足项目限期而采取紧急措施的情况下，有时经理人员要求将所有资源转移到紧急的项目上，图 20.9.5 形象的说明了当一个项目组人员太多时，将会出现的情况。这时将不可能进行内部协调。当头都不知道尾在做什么的时候，即使每一个成员都忙于从事某种与项目有关的工作，项目的进度还是要停顿下来。

对于每一个确定的项目组都有最佳规模。与项目有关的所有经理和公司行政人员都应当很好地掌握这样一个格言：与其过分地扩大项目组织规模，造成欲速则不达的局面，还不如推迟项目的实现时间。

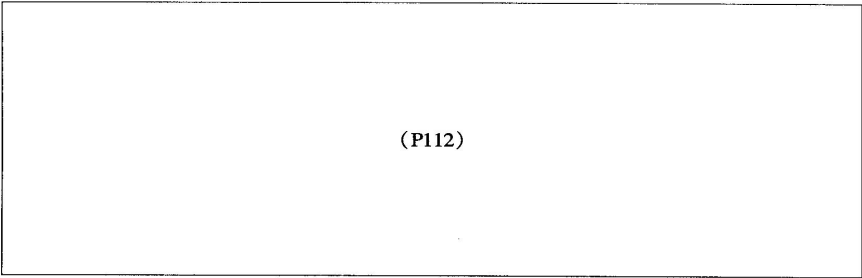


图 20.9.5 项目组的规模

(12) 估计人员要求并进行人员委托

一个项目的成功与否在很大程度上依赖于用户与公司经理、其他专业领域人员以及某些范围内信息服务人员(如，数据库管理员，联系用户的人员等等)。由于某人(或某部门)忘记或不承认以前的口头上的委托，会使得许多紧急项目被延误。因此有必要签署一个书面的人员委托书。应该造表列出在系统开发过程中所直接参与到的项目组的人员和其它人员(如访问用户人员、收集数据人员等)，并同时列出在每一阶段对他们的相对的时间要求(见表 20.9.6)。项目的人力要求来自于可行性研究报告。

报告标题:估计人员要求		日期:12月8日		
系统标题:市场分析系统		标识: MARS		
部门	业务头衔	时间百分比		
		第Ⅱ阶段	第Ⅲ阶段	第Ⅳ阶段
管理信息部	系统协调员	60	30	80

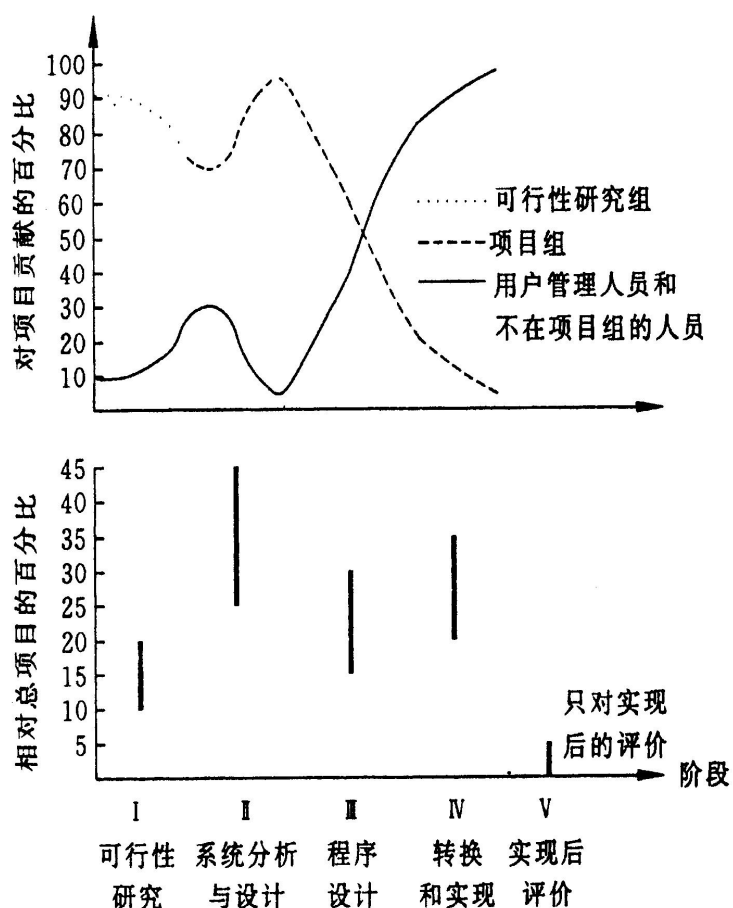
管理信息部	高级系统分析员	100	10	80
管理信息部	高级系统分析员	100	10	80
管理信息部	高级系统分析员	60	0	0
管理信息部	高级程序员	0	80	20
管理信息部	程序员	0	100	100
市场部	经理	10	20	30
市场部	经理助理	10	20	30
市场部	文职人员	10	10	40
市场部	文职人员	10	10	40

图 20.9.6 估计人员要求

没有书面人员委托而进行的项目肯定会产生不必要的延误，甚至可能失败。本书把项目开发的重要性放到一个恰当的位置。在项目中所涉及到的许多人并不在项目组内。由于这些的多数都理解他们的例行活动比项目所涉及的任何外部事物更为重要，所以一个书面委托是必不可少的。不幸的是，项目委托有时超过了他们按常规分配的工作负荷。在这种情况下，需要经理直接参与、定期督促和采取干预措施。

图 20.9.7 对于在各个阶段人员委托的相对要求上给读者一个感性的认识。图 20.9.7 的底部描绘了在系统开发的每一阶段占总的项目工作量的百分比，对每一阶段提供了项目工作量百分比的一个范围。公司的政策以及系统开发方法学将影响到相对百分比。例如，一种强调设计阶段(III)的方法学将必定有更为清楚定义的程序功能说明书。因此减少了程序设计工作所要求的时间。作为一个规则(到目前为止)，花在第II阶段(系统分析和设计)上的工作量是与花在第III阶段(程序设计)上的工作量成反比的。在一个设计良好的系统中，第II阶段将具有比第III阶段更大的工作量。

D:\HFZDCPC\HF4907.PS2 4/9/1997 13:20:19



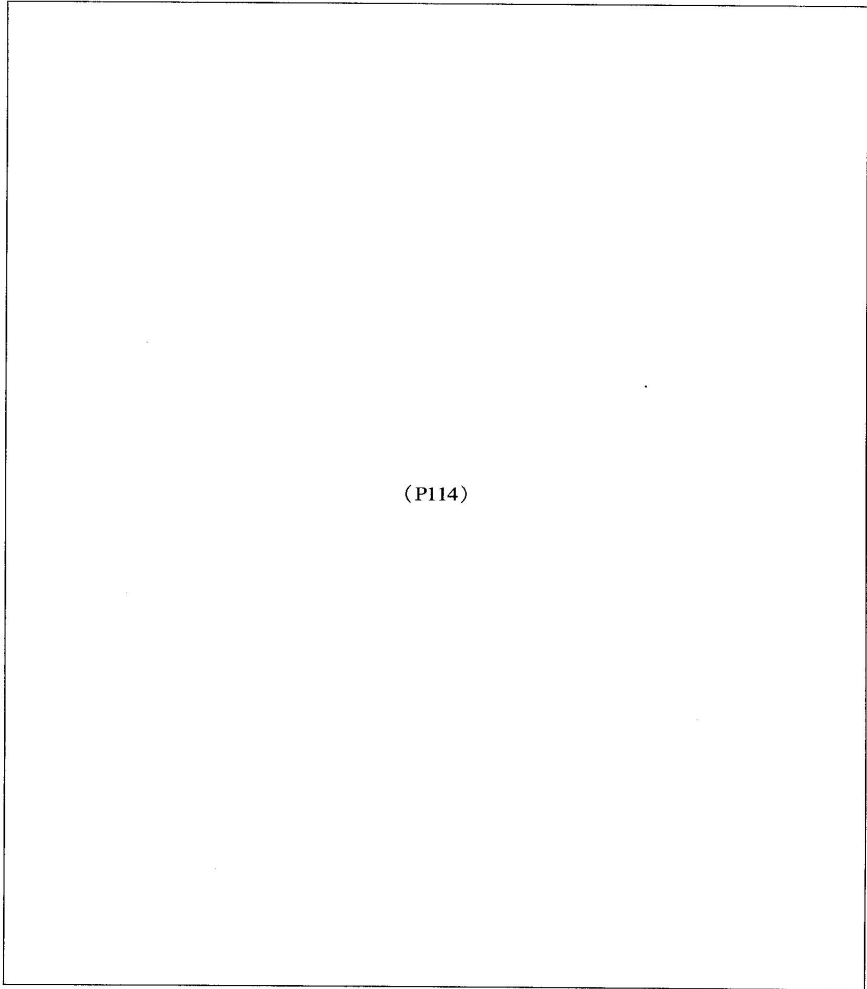


图 20.9.7 相对的项目工作量

图 20.9.7 的上端说明了由项目组(用户和信息服务人员)和非项目组成员的用户对项目工作贡献的相对百分比。注意,在第Ⅱ阶段期间,30%的工作量是由不在项目组的用户做的。在第Ⅱ阶段(系统分析和设计)期间,项目组必须不断地在每一级与用户进行通信。在程序设计期间,仅仅在外围才涉及到用户。在第Ⅳ阶段(实现和转换),在培训、测试、数据转换和并行操作中都涉及到用户。在第Ⅳ阶段中项目组和用户肩并肩工作,直到实现系统。在第Ⅴ阶段,将系统转交给用户。

(13) 人员培训

为了在系统开发过程中进行有效的交流,可能要求对于在设计数据库时所涉及的用户以及在生产调度中所涉及的信息服务人员培训。根据经验,信息服务人员负责信息系统方面的培训,而用户则负责专业领域的培训。

这个活动的产品是一张表,表中列出要求某种培训的人员的名字和头衔。每行表中都注明那种培训的简单描述,包括地点、负责人以及计划的时间等。有些培训将要求马上进行,而另一些培训(比如数据录入)将推迟到项目接近实现时进行。

(14) 建立详细进度表

通过使用一种标准的系统开发方法,管理人员可以建立阶段标志(见表 20.9.2 的活动 5, 10, 19, 23, 27, 29, 32, 33, 和 36),然后,利用历史统计数据 and 经验来估计中间和最后活动完成的日期。项目组组长必须与信息服务人员以及业务领域的管理人员密切合作以保证在系统开发过程中在各关键点有足够的人员。

系统开发过程本质上是线性的(一个活动接着一个活动),而且是不难用适当的准则(方法学)和合理的估计来监视的。表 20.9.8 说明了一个典型的信息系统项目进度表。在活动点

上加上三种标志之一以指出该活动的状态。如果情况表明该活动是不必要的，则在活动号上加一个圆圈。如果一个特定的活动正在着手进行，则在相应的活动号上划一个对角线。一旦活动完成则将对角线改成交叉线“×”。有时也用甘特表来给出项目进展的图形轮廓。

在开始一组有阶段标识的活动之前，要准备一个更为详细的进度表，来单独安排这些中间活动。对于要求多于两周时间的那些活动将以两周为增量来安排进度。表 20.9.9 说明了对具有阶段标志 E 的那些活动的一个详细的信息系统项目进度表。

表 20.9.8 信息系统项目进度表

阶 段

+具有阶段标志完成的活动

阶段 标志 活动

估计的 开始时间 实际的 开始时间 提前或 推迟的 天数 估计的
完成日期 实际完 成的日期 提前或 推迟的 天数

A 1 2 3 4 5 198W.9.1 198W.9.1 DS 198W

.10.1 198W.10.15 12B

B 6 7 8 9 10 198W.10.1 198W.10.20 14B 198W.11.1 198X.12.1 22B

C 11 12 B13 14 15 16 17 18 19 198X.9.15 198 Y.9.1 13A
198X.12.25 198X.12.20 3A

D B20 21 22 23 198Y.1.15 198Y.1.15 DS 198Y.2.15

E 24 25 26 27 198Y.3.1 198Y.6.30

F 28 29 198Y.6.1 198Y.7.15

G 30 31 32 198Y.6.25 198Y.9.10

H 33 198Y.10.1 198Y.10.31

I 34 35 36 198Y.11.1 198Z.2.1

1 =已开始的活动

×=已完成的活动

0 =不要求采取措施

+对应于图 20.9.3 中的方法学

*直到实现可行性研究之前，并不进行第 II 阶段活动 V 的估计

A =提前的工作天数 B =推迟的工作天数

DS=正在进行

表 20.9.9 信息系统项目进度表具有阶段标志 E 的活动

阶段标志 E—细节

活 动 估计的开始 时间 实际的 开始时间 提前或 推迟 天数
估计的完 成日期 实际的完 成日期 提前或 推迟 天数

24 指定程度组长 198y, 3.1 198y, 3.8

25 安排顺序和分配 程序 198y, 3.5 198y, 3.12

26 安排程序准备 进度 198y, 3.15 198y, 3.25

27a[KG*2]编定、测试程序 并编写程序资料 198y, 4.1 198y, 4.11

27b[KG*2]同上 198y, 4.15 198y, 4.30

27c[KG*2]同上 198y, 5.1 198y, 5.14

27d[KG*2]同上 198y, 5.15 198y, 5.31

27e[KG*2]同上 198y, 6. 198y, 6.14

27f[KG*2]同上 198y, 6.15 198y, 6.30

*以阶段标志 D 的活动 A=提前的工作天数

B=推迟的工作天数

实际开始时间为准 os=正在进行

下面的方法可以用来估计价格、人员以及相应的时间要求。这种循环使用的方法使得一组人能意见一致，而且对于信息服务项目特别合适。我们假定参与估计的那些人能够提出问题或具有任务方面的知识，而且能够提出支持自己意见的重要的理由。参与建立信息系统项

目进度表的人可以包括项目组长、起作用的用户经理以及其他有经验的信息服务人员(他们不一定与本项目有关)。我们通过以下几个步骤来描述进行合理估价的方法。

- ①项目组长介绍任务(例如,确定项目进度表的阶段标志的日期)和相应的背景信息。
- ②每一个参加者提交一个书面估计(成本、人员要求或时间)。
- ③项目组长(以线性比例)绘出该组每个成员的估计。
- ④计算上、下四分点和中点,并且标上迟度。
- ⑤要求其估计低于上、下四分点的那些参加者解释他们低或高估计的理由。
- ⑥项目组长就所标绘的估计召集一次公开的讨论会。
- ⑦重复步骤②至⑥,直到达到精确性要求不需要再循环为止。通过每一次循环,将降低估计的误差。
- ⑧估计是取中间值或(在适合时)取平均值。估计的误差是包含危险的一种标志。

(15)与用户人员交谈

与用户交谈的过程从本活动开始。为了解决问题和确定系统要求,项目组成员定期与有关用户见面。与用户交谈及反馈的过程贯穿于系统开发的全过程。

对于详细设计的基本输入是:(A)初始设计(来自可行性研究),(B)对现有系统及其成分的评价(也是来自可行性研究)以及(C)输入、处理以及输出的要求(由用户提供)。

①项目组与有关的用户人员检查在可行性研究的初始设计中所描述的输入/输出要求和频率,并根据需要及价值对每一种输入/输出进行评价。许多输出是“有了更好”,但是却不值得去产生它们。还可以根据周期和时帧来估计输入/输出。通过估计频率/价值比的平衡来优化周期的输入和输出。例如,如果每周情况报告可以满足需要,那么就没有必要再产生每天的情况报告。在联机系统中,检查响应时间要求以确定这种时间要求是否太紧迫,能否适当放宽要求而又致于对运行效率产生较大的影响;或者确定这种响应时间的要求是否不能满足。

②目前系统的资料对设计提供了有价值的输入。现有的报告、表格、原始资料等等,实际上能够追踪最终用户以便确定该资料是否合适,是否及时等。如果是,还能做哪些工作来改进它们?项目组负责对现有的所有输入和输出进行修改。通过合并类似的输入和(或)输出以及消除多余的信息尽可能地减少重复。

③初步交谈的一个直接结果是对所建议的系统所有的输出一般的描述(报告,显示或事务)。根据周期、初始用户、输出介质、内容以及分布来描述每一种输出。

(16)说明数据库要求

数据库用来支持系统的处理,特别是支持系统的输出。在目前系统的资料中包含了可继续使用的数据元。许多现有数据元的格式肯定是需要改变的,还需要将支持系统功能要求所需要的其他数据元标列出来。

项目组设计和编制数据字典,在一部数据字典中所列出的数据具有维持每个数据元的基本信息,而它们与数据库或文件的组织形式无关。在表 20.9.10 给出的数据字典的例子中,包括对每个数据元指定了一个各自的前后参照号、标题、描述(如果必要的话)、是否被编码、程序设计标识、存储单元(字符)数、格式和存储器大小(程序最初使用的)以及职责等。用户必须给出负责的人或部门、存储单元以及是否对数据元编码等事项。表 20.9.10 的数据字典形式,也可以用来交叉引用在所有原始资料、报告、文件以及数据库中出现的每一个数据元。

在标列出所有的数据元之后,项目组与数据库管理员合作来进行记录格式和文件的设计,或者,在数据库环境下,他们设计数据库的模式。此活动的输出是数据字典以及有关文件和(或)数据库模式的一份详细的技术描述。

表 20.9.10 数据字典

报告标题	数据字典	日期
系统标题	标识	
编号		
标题	描述	编码否
标记	字符数	字形/格式
存储	职责	
原始资料(S)、报告(R)、文件(F)、或数据库(D)		
工资支票(R)	工资登记簿(R)	工资主文件(F)
会计文件(F)	工时卡(S)	
1 社会保险号	职工	否

```

99999 P 人事
× × × ×
2 姓 否
LNAME 13 ×(13) E 人事
× × × ×
3 名字 职工 否
ENAME 10 ×(10) E 人事
× × ×
4 名字首字母 职工 否
MI 1 × E 人事
× × ×
5 部门 职工亲属 是
DEPT 3 ××× E 人事
× × × ×
6 性别 男或女 是
SEX 1 ×E 人事
×
7 工资 月工资 否
SAL 6 9999 P 人事
× × ×

```

(17) 建立控制和后援的方法

为了保证信息系统的正确性、可靠性和完整性，在设计时就要考虑加进控制手段。项目组将说明在系统设计时要嵌入所有物理上的和行政管理上的控制。在系统的输入、处理和输出阶段用以控制系统的技术的范围是广泛的。在处理之前核对输入，在处理期间使用诸如合理性检查以及数字位检查等技术以便最小化或消除在计算或处理中的过失误差，记录计数和长度核对是用来保证输出正确性的许多技术的代表。

为了避免在系统故障期间造成破坏，需要确定后援(备份)和校验点/重新启动的方法。这些方法描述了包含在系统中的克服故障的额外处理，在系统故障的情况下，利用备份文件和(或)备份事务日记从上一个“校验点”来重新建立处理。在上一个校验点“重新启动”系统，并重新开始正常的运行。在系统处理周期期间，定期地建立校验点将会使系统及时地保留在该点的所有处理，而且不会被破坏。

(18) 完成详细设计

详细的系统设计是分析输入/输出、处理、控制和后援要求的结果。系统初步设计或系统一般设计只描绘了各主要处理活动之间的关系，而系统详细设计则扩展到包括所有处理活动和有关的输入/输出。这是系统开发过程的基础活动。正是这一步，将功能说明书与技术上和方法上的新设施结合在一起以实现一个系统。详细设计是前面所有工作的归宿。此外，它也是该项目今后所有活动的一张蓝图。

在活动 5 中提到了用图形说明系统设计所使用的若干技术(但没有详细讨论)。这里我们简单地讨论其中三种技术—流程图、HIPO 以及渥宁(Warnier)图。用来形象地描述工作流程和总的系统设计的最流行的技术是流程图。流程图使用刻画系统逻辑的一些专用符号并通过流线把这些符号相互连接起来以说明工作流程和数据流程。图 20.9.11 给出了系统流程图符号的一个子集。在图 20.9.12 中，用流程图描绘了一个已投入运行的工资系统的一部分。流程图有一定的缺点。不像前面所讨论的其他两种技术，流程图并不鼓励分析员使用系统设计的自上而下或模块化的方法。因此，用流程图方法来设计系统，不仅难于设计，而且设计出的系统也难于理解和维护。流程图之所以较为流行，主要是由于它是最早出现的设计方法。

层次式输入—处理—输出法(又称 HIPO 法)是在一层次体系中将系统设计按其详细程度分层，依次地说明所有的输入、处理和输出的一种方法。图 20.9.13 说明了一个工资系统的 HIPO 卷内容表(VTOC)。VTOC 是在 HIPO 设计方法中所使用的几种标准形式之一。整个系统被划分成由若干逻辑模块所组成的一个层次体系，并用 VTOC 来描绘。此后，利用粗框图和细框图还可以将这些模块进一步划分成更细小一层的输入—处理—输出的细目。通常由若干个

VTOC 将设计的层次体系统推进到依次的细目层。从 HIPO 结构化方法所得到的好处往往被编写系统资料所需要的大量繁琐的文书工作所抵消了。

Warnier 框图(在图 20.9.14 中说明)可以用来设计整个系统、数据结构、报表内容以及数据元的编码。使用 Warnier 框图的依据是：应该围绕着数据结构来设计系统。Warnier 框图的最大优点是对各种环境的适用性。图 20.9.15 中的例子是一个扩展项判定表，它是许多判定表中的一种，一个判定表有一个条件分叉(在表的左上方)和活动分叉(在表的左下方)，一个条件项(右上方)以及一个活动项(右下方)。判定表并不是一个说明数据流和工作流的有效工具，最好把它作为其他设计方法的补充。判定表的主要好处是必须考虑到每一种可能的替换者、选择、条件、变元等。与流程图，HIPO 图以及其他设计方法不同使用 Warnier 框图法，系统分析员不必考虑细节。

图 20.9.11 部分系统流程图符号

图 20.9.12 简化的工资支付系统流程图

工资系统 系统开始 每月处理 月初 每周处理 提交时间卡片 数据录入 按工时处理
职工记录 工资支票 工资联单 更新工资文件 月末 按月薪处理职工记录 工资支票
工资联单 更新工资文件 系统结束

图 20.9.14 Warnier 框图

图 20.9.13 HIPO：卷内容表

上面讨论的分析工具代替了一大段解说词，而通常对解说词的理解容易产生混淆。然而，精心设计的解说词可以而且应该用来支持图形设计技术。

没有一种分析和设计的技术是最好的，最好的分析和设计技术是适合一个公司具体情况的各种技术的组合。总之，模块化的自顶向下方法是当今必不可少的。按自顶向下方法进行设计时，通过最高一级的管理者来建立基本的系统目标，然后根据在公司每一级收集的输入数据，在设计中增加后继的细目层。由于作为一个整体概念多数系统过于复杂，所以将系统分成若干个更容易理解的模块。模块化的主导思想是“各个击破”，而这是行之有效的。

(19)指导用户或信息服务部门预演。

表 20.9.15 一张判定表

支付类型	工资	按工时处理	佣金			
时帧	周末	月末	周末	月末	周末	月末
打印工资支票	×	×	×	×	×	×
打印工资联单	×	×	×	×	×	×

结构预演是一种预测评价方法，它能有效地减少某些被忽略的或作错的事情。它也给预测者提供一个机会来评价那些业已建议的事情(如系统设计)，从而有可能给出一些建设性的建议。预演的目的是给项目组提供有价值的反馈信息，而不是对系统的质量下判决性的结论。

项目组长应考虑何时开始结构预演。通常预演是在系统设计以及系统开发过程中其他一些关键点(如，测试计划、程序描述等)完成之后才进行。

参与结构预演中的人有：若干项目组成员，一个协调员，参加者，一位秘书，或许还包括一位不属双方的“中立的”经理。项目组的某个成员或所有成员扮演“推荐者”的角色，并且解释他们所承担设计的系统的那一部分。协调员负责组织预演和协调“推荐者”与“参加者”之间的相互配合。根据对所提出的课题的知识和兴趣来选择“参加者”。这些人应该没有直接参与本项目的。秘书将对一些要点作书面记录。通常邀请一个“中立的”经理参加第一次预演。中立经理的出席将促使参与预演的每一个人专心于他的工作(这一点有时是预演的一个问题)。

结构预演的方法是简单的。在进行预演的前几天将需要审查的材料(即系统设计)分发给参加者，协调员负责跟参加预演的所有人联系和通信。在实际的预演期间，推荐者解释系统设计以及有关的资料。这是通过一步一步地预演系统来进行的，有时可能还借助于某种设计工具。参加者提供出讨论的建议，而秘书则记录下来以形成资料。通常一次预演持续的时间不应超过一个半小时。如果超过了这个时间限制，那么一次预演会议将变得没有实际效果。如果必要，可以安排几次会议来完成预演。

项目组评价所有的建议，并且把所有有价值的建议并入到系统设计中。预演是有价值的，它使得设计者在系统实现之前获得重要的反馈信息。

(20)选择硬件

如果正在开发的系统要求额外的硬件支持,则需要选择适当的硬件并进行订货。获得硬件的过程通常是信息服务经理的责任。

(21)准备输出格式

在系统开发过程中,到目前这一阶段为止,我们已经提及了输出并描述了其有关的内容,但是程序员需要知道具体的输出形式(即应该怎样在输出设备上出现)。这种详细的输出说明称之为输出格式。项目组产生出显示屏(VDU)格式,这种格式规定了诸如题目、标题、输出形式等项,有时还应包括输入形式。

某些硬拷贝报告和资料要求事先打印好的表格纸,项目组与表格纸厂商的代表合作设计这种事先打印好的表格纸(例如,工资支票和短线)。

项目组还负责设计和满足在系统范围内所有人工产生的报告和资料,同时与受有影响的用户经理相配合进行修改、增加或删除。

(22)描述数据项的说明书

数据项的说明书详细规定了什么数据将输入到系统以及它们怎样被输入到系统中。

(23)准备程序描述

系统开发进展到目前这一步,我们已经对现有的系统作了详尽的分析。它的功能已经并入建议的系统的设计中,我们已经完成了建议的系统及其支持的数据库的设计,并且还准备了所有输入/输出详细的说明书。现在项目组可以着手标列和确定所有的程序,而这些程序是使得建议的信息系统运转所要求的。系统的图形表示(流程图、HIPO图和其他)是标列所要求的程序的初始输入。对每一个程序,项目组编辑下述的资料:

- ①程序语言的种类(例如, COBOL、BASIC、FORTRAN)
- ②程序解说词的描述—描述要执行的任务。
- ③由程序所产生的各种输出的描述和格式
- ④处理频率(例如, 每天、每周、联机等)
- ⑤界限和限制(例如, 输入数据的顺序, 容量的限制, 响应时间, 最大值, 最小值等)
- ⑥详细说明书(例如, 排序, 编辑的标准, 特殊的计算和逻辑操作, 各种表格等)。

3. 第Ⅲ阶段—程序设计

项目组现在可以着手开始与计算机通信了。这种通信(或与计算机的接口)是采取指令形式来进行的,而这些指令被编进计算机程序中。这些计算机程序包括系统运转所必需的软件。在第Ⅲ阶段—程序设计阶段将开发支持信息系统所要求的全部软件。

用户的介入集中在系统开发的过程前段(第Ⅱ阶段)和后段(第Ⅳ和Ⅴ阶段)。如果正确地完成了第Ⅱ阶段而且用户与项目组的协作是有“成效”的,那么用户将很少介入程序设计阶段,甚至完全不用介入。用户介入最多的情况将反复出现在系统设计需要澄清的时候,有时也出现为第Ⅳ阶段(转换与实现),作一些初始计划的时候。

不幸的是,有时用户管理人员也较深地卷进了程序设计阶段。这是第Ⅱ阶段进行得很糟糕,而且当开始程序设计时还没完成的一种标志。这种情况是经常发生的,特别是在时间紧迫时,项目组常常收到一些强制性的命令要求产生尚未完成的产品。由于系统开发过程的最终产品是软件,所以有时过早地开始程序设计。这种系统开发方式必然导致产生质量低劣的系统。这种系统并不能满足用户的要求,而且维护的代价很高。这种系统整个寿命期的成本可能是一个高质量的系统的两到三倍。

(24)指定程序员组长

通常项目组长是一个系统分析员或是一个用户,他并不直接参与程序设计工作。管理程序设计工作的人应该是程序设计工作实际的参加者,因此,对于要求两个人以上的程序设计工作,将由信息服务经理指定一个程序员组长。当然,项目组长仍然对整个项目负有责任。

程序员组长有时也称作为主程序员。他(或她)可能只花 10%的时间在产品的程序设计上。如果只需要管理一个下属程序员,那么主程序员可能花 80%的时间在产品的程序设计上。

(25)安排顺序和分配程序

一个信息系统的软件包,可能要求几百个程序。并不需要按照这些程序最终执行的顺序来编写它们,在建立程序开发进度表时,必须考虑到许多变化的因素。在安排程序编制顺序时,主程序员应考虑如下问题:

- ①建立和维护测试文件的需要

③程序的长度和复杂性

(26) 安排准备程序的进度

(27) 编制、测试程序和编写程序资料。

图 20.9.16 程序的甘特进度表

- ①准备一般的程序逻辑框图
- ②准备详细的程序逻辑框图
- ③编写程序(写程序语句)
- ④测试和调试程序
- ⑤编写程序的资料

表 20.9.17 程序进度表

[illegible]

(28) 完成转换计划

转换系统的处理本身就是一个系统，而且应该像最好的结果那样来处理。项目组与用户管理人员以及信息服务审计组合作，共同研究以设计出一项转换计划。该计划包括：系统验收测试，文件或数据的转换，用户培训以及并行工作(如果必要的话)的细节。转换计划详细

地细述了用户及信息服务人员的义务和责任，同时还规定了进行这些事情的时间限制。

(29) 指导系统验收测试

虽然已经测试了各个单独的程序模块，但是还没有把它们结合成一体作为一个系统来处理。一个信息系统可能有 100 个以上的程序和一打以上的文件，必须把它们作为一个整体来处理以保证使工作协调并使用户满意。整体的测试将验证全部系统软件和应用软件、输入/输出，文件和数据库以及各种过程。在测试期间用户人员是实际的参加者。在测试过程中，有可能发现错误(忽略了系统的某些方面)，某些过程的缺点将会暴露出来。可以肯定，一部分验收测试过程必须在系统设计和程序设计方面进行较小的修改。如果系统是正确开发的，那么任何这种修改将只是微小地调整系统。任何重大的修改应该推迟到系统实现之后，并且至少在进行生产性工作一年之后再行进行。这种推迟避免了通常敲打膝部那种反作用引起的改变而提交可观的资源。这是因为为了减少重大修改的要求，项目组长和受影响的用户管理人员将要停止信息系统的每一方面。这时，重大修改的要求才是一种分界清楚的标志，它表明有人忽略了他们对项目的责任。

整个系统的测试实际上是分两个部分完成的。首先利用测试数据来验证每一个子系统。一旦证实所有子系统的功能是适合的，则有“生存的”数据来测试整个系统。测试数据是为了测试特定的环境而产生的，而“生存的”数据通常是来自过去处理使用的实际的数据。

在测试联机系统时(此时响应时间是关键问题)，为了测试系统的能力，包括了用几种生存数据的测试会话。系统可能运行良好，但是由于计算机能力不够大或是程序的效率不高，也可能导致不可接受的响应时间。

(30) 设计用户手册

项目组设计一套用户手册，并且在对系统验收测试的同时指导用户的培训活动。每个信息系统都应该有一套用户手册，它们提供有关系统运行的命令和解释。用户手册和有关的培训对于系统的最后成功是至关重要的。光有一套用户手册是不够的，这些用户手册还必须是一种高质量的资料，它们能对系统的每一方面提供快速和容易的参照。用户手册至少包括：

- 系统的目标
- 系统的描述
- 工作流程和一般的操作方法
- 完成和理解输入/输出的命令
- 数据收集和更新的方法
- 控制
- 其他(例如，术语唯一的词汇表，硬件的描述和用法，性能的界限，等等)

用户手册的内容来自系统资料。然而，在编写和编译这种手册时必须考虑到能为预期的用户所理解，而且不会被错误地解释。

(31) 提供用户培训大纲

如果不能跟培训相关联，那么用户手册的价值就很小。项目组的成员指导一系列的培训课程以使得用户熟悉系统。用户培训大纲的一般内容包括：

- 系统的用途和目标
- 现有系统与新系统的差别
- 系统工作概述
- 如何使用用户手册
- 与系统有关的信息服务人员 and 用户人员的义务和责任

一个有各地分号的大型百货商店实现了一个联机销售点(POS)系统并将用户手册分发给每一个 POS 终端地点。如果没有正规的培训，销售员将丢下他们自己的工作而去揣摩用户手册(有 100 页以上)以了解系统的用途。由于销售人员不能处理基本事务，于是使得顾客不再等待，而跑到其他地方买货。在他们认识到问题不在于市场、产品质量或地点之前，百货商店的这些分号几乎要关闭。问题在于缺乏对系统用户的训练。

(32) 建立和转换文件或数据库

很难找到一个已实现的系统而不需要修改原有的文件或数据库。有些文件和数据库需要新建，而其他一些则需要从现有的转换成适合的格式。用户部门负责将手写的数据统一格式并变成机器可读的形式。用户部门也可能负责抄写和录入数据的工作。如果数据不是现成可用的或没有用人工存储起来(例如，存放在 3×5 的卡片上)，那么数据的准备工作可能耗费

相当长的时间。

在项目组的指导下，用户负责新产生的和转换的那些文件的一致性。数据的校对是将人眼现场检查与计算机自动校验结合起来进行的。随机抽样检查可以有效地用于非常大的文件或数据库。在建立和转换处理期间掌握时间是很重要的，因为一旦建立了一个文件或数据库，此后就必然要对它们进行连续地更新。因此，最好的策略是：在并行工作开始之前（或者在不要求并行操作的情况下，在系统实现时）正好完成建立和转换工作。

(33) 完成并行工作

并行工作意味着同时运行原有的系统和新的信息系统。并行工作是常用的手段，特别是当系统故障相当大地影响到公司的运营时更是如此，在并行工作期间，用户和信息服务人员被分散开了，因为两个系统都需要维护。完成并行工作是十分困难的，因为参加的人员仍然处于开始阶段。

通常安排并行工作持续一个主要的系统周期（一般是一个月）。项目组长和受影响的用户管理人员以及有关的信息服务经理监督并行工作的进程。某些单位已经接受了并行工作至少要进行一个主要周期的方针，而另一些单位则决定维持原有系统直到经理认为新系统已经全部运行时为止。

如果在并行工作期间出现了一次较大的故障，则应中断并行工作并进行有关的修复工作。由于必须维护文件和数据库，所以及时性是十分重要的。

如果公司改进他们的系统测试方法，那么信息服务和用户人员就会自信他们有能力去实现一个系统。有些公司放弃并行工作，尽管这种做法有很大的危险，但是这样将把力量集中在成功地实现一个新系统上。在某些情况下，由于时间和人力有限，不能进行并行工作，因而经理的代替办法是直接实现新系统，并且要求进行充分的系统测试。

5. 第V阶段—实现后的评价

第V阶段（实现后的评价）常常被忽略。由于其他紧急的信息系统项目需要人员，往往进行很少的，甚至不进行实现后的评价，不管好坏，系统就被接受了。实现后的评价或定期系统评价应该是系统开发过程的组成部分。任何信息系统在刚刚实现之后都将要求做某些“微小的调整”。为此，必须在系统投入生产前，对它进行评价。因为一旦系统投入使用，即便实现前的测试设计得很好，也不可能完全暴露出某些在系统投入运行时必将出现的问题。委托并进行评价活动的好处是获得更高质量的系统并且使用户更为满意。

(34) 调整成本

项目组长调整项目的成本以如实反映I、II、III、IV阶段的最终系统开发成本。此外还将成本汇总以反映出维持系统运行的成本。

直到系统实现至少一个月之后，才有可能算出精确的、符合实际的成本数据。

(35) 指导系统实现后的评价

系统实现后的评价（系统的一个关键检查步骤），由从项目组和受影响的用户部门挑选出的人员来指导进行。在系统运行的头几个月，由于存在着对改革的阻力，对系统的把握不够以及非预期的问题等，因此，不宜立即进行系统实现后的评价。通常在第IV阶段完成后的3~6个月之间进行系统实现后的评价。

项目组和用户部门挑选和人员并指导系统实现后的评价以决定：

实际的与预期的性能的比较。利用在系统设计时已建立起来的某些标准（例如，在峰值工作负荷时的响应时间），将实际的性能与预期的性能进行比较。

系统目标实现的程度。针对在可行性研究中建立的那些目标来评价系统。例如，系统能否给审计员提供非常及时的信息以进行更好的决策。

非预期的利润或耗费。几乎任何基于计算机的系统都将导致非预期的利润和耗费。这些利益或负荷提供了评价整个信息系统效率的直接输入数据。坦率地讨论错误。很少有在系统开发过程中不犯错误而实现一个系统的。应该将项目组、用户经理、用户人员、其他信息服务人员或信息服务对策委员会坦率、详细地讨论的错误编写成资料。列举这些错误并非用来追究某个人或某团体的责任，而只是着重强调为什么会产生这些错误以及可以做些什么努力以便在今后的项目中消除这些错误。

系统实现后的评价被提交给信息服务经理和用户经理以便采取适当的措施。

(36) 准备系统检查的计划

很多数据处理系统和信息系统在实现之后都保持原样，而没有作出任何共同的努力去显

著地提高它们。在这些系统中，所谓的改进工作是不超出例行维护的范围的，而且是由于用户的反映才作的。这种被动地改进系统的方法其效率远比由定期的系统检查来保证的主动的方法要差得多。有很多原因会导致忽视定期检查，因此，应该通过一个正式的书面检查计划来督促进行检查。两次检查之间的间隔时间是根据系统的复杂性和易变性来决定的。

定期系统检查是业务领域管理人员的责任。由检查所产生的各种建议将最终反映在由用户管理人员提交的一个服务请求中。