

第3章 软件项目策划与项目计划

3.1 软件项目策划

项目策划是一种具有建设性、逻辑性的思维过程,其目的是把所有可能影响决策的决定总结起来,对未来项目开展起到指导和控制作用,最终借以达到项目的目标。软件项目策划是软件项目开始前必须进行的一项活动,对软件开发和软件管理进行问题定义,为项目系统提供一个总体框架。

3.1.1 软件项目策划概述

一个软件项目提出的主要原因表现在两方面:一是软件可以解决某个问题;二是软件的使用可为某个组织带来改进或提高效益的机会。软件项目策划就是要选择可行的软件项目,并做出开发的准备工作。

1. 软件项目策划的特点

软件项目策划是软件开发的前导工作,这项工作的好坏将直接影响整个软件项目的成败。充分认识这个阶段工作所具有的特点,可以提高策划工作的科学性和有效性。

软件项目策划工作是面向长远的、未来的、全局性和关键性的问题。因此,它具有较强的不确定性,非结构化程度较高。软件项目策划不是解决软件开发中的具体业务问题,而是为整个项目确定目标、战略、总体框架和资源计划,整个工作是一个管理决策过程。

软件项目策划的工作环境是组织管理环境,高层管理人员(包括高层信息管理人员)是工作的主体。软件策划人员对管理与技术环境的理解程度,对管理与技术发展的见识,以及开创精神与务实态度是软件项目策划工作的决定因素。

2. 软件项目策划的内容

软件项目策划包括先期调查与分析、选择软件项目、定义目标和基本要求等内容,并在上述基础上,编写软件项目建议书。

1) 先期调查与分析

软件项目的先期调查是项目策划的基础,它采用科学的方法,有目的、有系统地收集和分析项目的各种信息,作为项目分析的前提。

项目先期调查主要内容包括对社会因素、经济因素、文化因素、技术因素、环境因素等项目环境的调查;对项目现状、用户业务需求、产品功能和性能需求的调查;对项目所开发的产品发展趋势的调查;对项目产品规格、商业渠道、应用环境约束条件等测控因素的调

查;对项目竞争情况的调查;等等。

根据先期调查的情况,有助于发现和寻找新的软硬件产品,为项目提供决策依据。在估计用户需求总量和获利可能性的基础上,进一步挖掘项目所产生的软硬件产品的性能和特征优势,预测新产品或新服务的需求量和增长率,了解项目市场多样化的需求和竞争者动向,提高决策的有效性。

2) 选择软件项目

软件项目由不同的人根据不同的原因提出,分析人员首先应分析项目提议的动机,在此基础上,选择软件项目应当考虑管理人员的支持,项目的执行时间,提高实现系统目标的可能性、资源和技术的可行性,与其他项目比较的效益。

3) 定义软件目标和确定软件基本要求

定义软件目标和确定软件基本要求主要通过与客户沟通来完成,其主要任务是确定软件项目中所涉及的工作、产品和功能,包括软件项目的目标、需求、约束和可交付成果等。

在确立项目目标和范围的基础上编写项目建议书,对项目背景、项目的意义和必要性、项目产品预测、项目规模和期限以及项目建设条件进行说明。

3. 软件项目团队的策划原则

软件项目策划活动包括一系列管理和技术实践,可以为软件开发团队定义一个便于他们向着战略目标和战术目标前进的路线图。不管我们做多少努力,都不可能准确预测软件项目会如何进展,也不存在简单的方法来确定可能遇到的不可预见的技术问题,在项目后期还有什么重要信息没有掌握,以及会出现什么误解,或者会有什么商务问题发生变化。为了应对上述问题,软件项目策划必须遵循如下原则。

(1) 理解项目范围。范围可以为软件开发团队提供一个准确的目的地。

(2) 吸收利益相关者参与策划。利益相关者(干系人)能够限定一些优先次序,确定项目的约束。为了适应这种情况,软件工程师必须经常商谈交付的顺序、时间表以及其他与项目相关的问题。

(3) 保持脚踏实地。人们不能每天每时每刻都工作。生活中常常会有疏忽与含糊,变化总是在发生,甚至最好的软件工程师都会犯错误。这些现实的东西都应该在项目制订计划的时候加以考虑。

(4) 基于已知的估计。估计的目的是基于项目组对将要完成工作的当前理解,提供一种关于工作量、成本和任务工期的指标。如果信息是含糊的或者不可靠的,估计也将是不可靠的。

(5) 计划的制订应按照迭代方式进行。项目计划不可能一成不变。工作开始后,有很多事情有可能会改变,那么计划就必须调整以适应这些变化。另外,迭代式增量过程模型指出了要根据用户反馈的信息(在每个软件增量交付之后)重新制订计划。

(6) 计划时考虑风险。如果团队已经明确了哪些风险最容易发生且影响最大,那就应该制订应急计划。另外,项目计划(包括进度计划)应该可以调整以适应那些可能发生的风险。

(7) 调整计划粒度。粒度是指项目计划细节的精细程度。一个“细粒度”的计划可以提供重要的工作任务细节,这些细节是在相对短的时间段内计划完成的。一个“粗粒度”的计划提供了更宽泛的长时间工作任务。通常,粒度随项目的进行而从细到粗。在几个星期或几个月的时间里可以详细地策划项目,而在几个月内都不会发生的活动则不需要细化。

(8) 制订计划确保质量。计划应该有利于软件开发团队确保开发的质量。如果要执行正式技术评审,应该将其列入进度;如果在构建过程中用到了结对编程,那么在计划中要明确描述。

(9) 描述如何适应变化。即使最好的策划也有可能被无法控制的变化破坏。软件开发团队应该确定在软件开发过程中一些变化需要怎样去适应,例如,客户会随时提出变更吗?如果提出了一个变更,团队是不是要立即去实现它?变更会带来怎样的影响和开销?等等。

(10) 经常跟踪并根据需要调整计划。需要每天都追踪项目计划的进展,找出计划与实际执行不一致的问题所在,当任务进行出现延误时,计划也要随之做出调整。

最高效的方法是软件开发项目组所有成员都参与到策划活动中,只有这样,项目组成员才能很好地认可所指定的计划。

3.1.2 现有系统分析

项目策划是在对现有系统分析的基础上开展的。现有系统一般是当前实际使用的系统,这个系统可能是计算机系统,也可能是机械系统,甚至是一个人工系统。分析现有系统的目的是通过分析现有系统或目前局部使用的软件的功能和缺陷,以进一步阐明建议中的开发新系统或修改现有系统的必要性。

1. 现有系统的分析与评价

对软件项目进行分析时,一般会面临一个选择,是淘汰现有系统并完全用新系统来代替,还是对现有系统增量地进行改造。对现有系统处理恰当与否,直接关系到新系统的成败和开发效率。

现有系统一般具有的特点:系统虽然可以完成企业中许多重要的业务管理工作,但已经不能完全满足要求;系统在性能上已经落后,采用的技术已经过时;系统没有使用现代软件工程的方法进行管理和开发,现在基本没有技术文档,很难理解,维护十分困难。

对现有系统评价的目的是获得对现有系统的更好理解,一般包括3方面,即商业价值评价、外部技术环境评价和应用软件评价。

(1) 商业价值评价。商业价值评价的目标是判断现有系统对企业的重要性。评价的基础是,充分了解系统在业务处理过程中使用的价值,系统与系统的关系,若系统不再运行需要付出的代价,系统的缺点及存在的问题等。

(2) 外部技术环境评价。外部技术环境评价包括系统的硬件、支持软件和企业基础

设施的评价。硬件包括许多需要经常进行维护的部件,评价特征包括供应商、维护费用、失效率、运行时长、功能、性能等。支持软件包括操作系统、数据库管理系统、事务处理程序等,支持软件一般依赖于某个硬件,评价时应注意这种依赖性。基础设施包括开发和维护系统的企业职责,以及运行该系统的企业职责,基础设施评价对系统进化起着关键作用。

(3) 应用软件评价。应用软件评价包括系统级和部件级两个级别:系统级关注整个系统,部件级考虑系统的每个子系统。

2. 现有系统的进化策略

对技术质量的全面评价结果与商业价值评价进行比较,可以为系统进化提供基础资料。按照商业评价价值和技术质量值的情况,可以把评价结果分为4种类型,如图3-1所示。

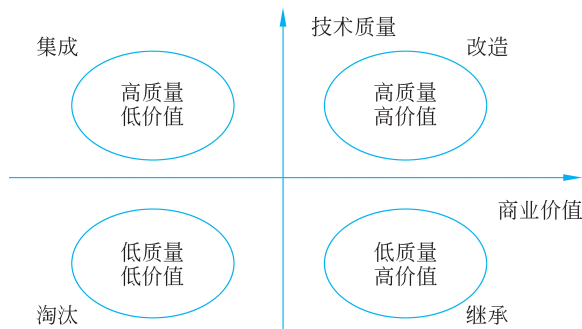


图 3-1 评价结果综合分析图

图 3-1 中,将现有系统的评价结果分列在平面坐标的4个象限内,处于不同象限的系统采取不同的进化策略。

(1) 改造策略。第一象限为商业价值高且技术含量高的系统,本身还有极大的生命力,基本能够满足企业业务运作和决策的要求。系统的进化策略分为功能的增强和数据模型的改造两方面。

(2) 集成策略。第二象限为商业价值低但技术含量高的系统,系统只能完成企业某个部门的业务管理,系统各自的局部领域工作良好,但对整个企业,存在多个系统,不同的系统基于不同的平台、不同的数据模型,其进化策略为集成扩展。

(3) 淘汰策略。第三象限为商业价值低且技术含量低的系统,对这种系统需要全面重新开发新的系统以代替现有系统。通过对淘汰系统功能的理解和借鉴,可以帮助新系统的设计,降低新系统的开发风险。

(4) 继承策略。第四象限为商业价值高但技术含量低的系统,已难满足企业运作的功能或性能要求,但目前企业业务尚紧密依赖该系统,可采用继承性淘汰的策略。在开发新系统时,需要完全兼容现有系统的功能模型和数据模型。

对现有系统进行分析与改造,需要全盘考虑、综合部署,寻求一种较好的解决方案。

3.2 可行性研究

众所周知,并不是所有问题都有简单明显的解决办法,事实上,许多问题不可能在预定的系统规模之内解决。如果问题没有可行的解,花费在这项开发工程上的时间、资源、人力和费用都是无谓的浪费。

可行性研究(Feasibility Study)是在项目建议书被批准后,对项目在技术、经济和社会等方面是否可行所进行的科学分析和论证。

3.2.1 可行性研究概述

可行性研究是确定建设项目前具有决定性意义的工作,是在投资决策之前,对拟建项目有关的自然、社会、经济、技术等进行调研、分析比较,以及预测建成后的社会经济效益。并在此基础上,综合论证项目建设的必要性、经济合理性、技术先进性和适应性以及建设条件的可能性和可行性,从而为投资决策提供科学依据。

1. 可行性研究的意义

可行性研究的目的是通过运用科学的方法对拟议中的项目进行全面的、综合的技术经济分析,回答本项目在技术上是否可行,经济上是否有生命力,财务上是否有利可图,需要多少投资,资金来源能否保证,建设周期多长,需要多少物力、人力资源等,进而判断该项目“行”还是“不行”;建设还是放弃。可行性研究的本质就是用最小的代价在尽可能短的时间内确定问题是否能够解决,其目的不是解决问题,而是确定问题是否值得去解。一项好的可行性研究,还要探讨从各种具有实际意义的可能方案中遴选出最佳方案。

可行性研究本身不是最终目的,其主要作用:建设项目投资决策和编制设计任务书的依据;项目建设单位筹集资金的重要依据;建设单位与各有关部门签订各种协议和合同的依据;建设项目进行工程设计、施工、设备购置的重要依据;向当地政府、规划部门 and 环境保护部门申请有关建设许可文件的依据;国家各级计划综合部门对固定资产投资实行调控管理、编制发展计划、固定资产投资、技术改造投资的重要依据;项目考核和后评估的重要依据;等等。总之,可行性研究是保证提高基本建设投资效果的重要手段,在基本建设中具有举足轻重的地位,是决定投资项目命运的关键文件。

可行性研究工作是由浅到深、由粗到细、前后连接、反复优化的一个研究过程。前阶段研究是为后阶段更精确的研究提出问题创造条件。可行性研究要对所有的商务风险、技术风险、经济风险和社会风险进行准确落实,如果经研究发现某方面的缺陷,就应当找出主要风险原因,从市场营销、产品及规模、工艺技术、原料路线、设备方案以及公用辅助设施方案等方面寻找更好的替代方案,以提高项目的可行性。如果所有方案都经过反复优选,项目仍是不可行的,应在研究文件中说明理由。应当注意的是,研究结果即使是不可行的,这项研究仍然是有价值的,因为避免了资金和时间的浪费。

2. 可行性研究的特殊性

软件项目可行性研究从技术、经济、工程等角度对项目进行调查研究和比较分析,并对项目建成以后可能取得的财务、经济效益及社会环境影响进行科学预测,为项目决策提供公正、可靠、科学的软件咨询意见。

软件项目可行性研究与其他工程项目可行性研究相比,存在着较大的差异。引起这些差异的因素主要有以下3方面。

(1) 人的因素。软件项目提供的实际上是一种服务,服务质量的好坏主要来自用户的体验,而这种体验具有个体差异性和主观性。在软件工程实施过程中存在变化的可能性很大,而这种变化对项目的成败以及其经济、技术和管理因素有很大的影响。

(2) 社会环境的因素。软件作为一种纯知识产品,使得软件项目的开发、交付(销售)和使用都涉及很多的道德、法律和人文因素。这些因素对软件开发过程存在着或多或少的影响,甚至会为整个项目带来意外的风险。

(3) 系统的因素。软件产品开发项目,尤其是具有特殊针对性的软件开发项目,往往是在现有系统(例如,人工系统)的基础上开发替代产品或升级产品,即是以现有系统为基础的项目。为了涵盖现有系统中需要保留的部分,需要对现有系统进行全面分析。

软件项目的可行性研究实质上是要进行一次大大压缩,简化系统分析和设计的过程,也就是在较高层次上以较抽象的方式进行系统分析和设计的过程。

3. 可行性研究的任务

在进行软件项目可行性研究中,有以下7个研究任务。

(1) 分析问题。进行概要的分析研究,初步确定项目的规模和目标,确定项目的约束和限制,并一一列举出来。

(2) 研究现有系统。研究与分析现有系统是开发新系统的基础。现有系统一定能完成某些有用的工作,新系统的目标也必须能完成这些功能;现有系统存在的问题,也是新系统必须解决的问题;现有系统的运行费用是一个重要的经济指标,新系统的运行费用不得大于或等于该指标;现有系统反映了用户的操作习惯,新系统应该继承这些习惯。

(3) 生成现有系统的物理模型。现有系统的物理模型包括新系统运行时所需的公共实体,是新系统的基础。

(4) 导出新系统的逻辑模型。从现有系统的物理模型出发,分析新系统的需求、相关实体、基本架构与原理,并据此生成新系统的逻辑模型,以此作为问题的最终描述。

(5) 提出解决方案。针对逻辑模型,探索出若干种可供选择的解决方案。

(6) 方案的可行性分析。对每种解决方案仔细研究分析其可行性,一般主要研究的是项目方案的经济可行性、技术可行性、社会可行性和操作可行性4方面。

(7) 选择最佳解决方案。对比每种解决方案的可行性论证结果,决定该项目是否值得开发。若值得开发,提出相对最佳解决方案,或者针对不同的侧重面(如效益、技术和效率),提出最佳解决方案。

4. 可行性评价准则

一般可行性评价准则主要包括经济可行性、技术可行性、社会可行性和操作可行性 4 方面。

(1) 经济可行性。经济可行性包括成本与效益两方面。分析经济可行性一方面要进行开发和运行成本的估算;另一方面要进行效益上的评估,包括经济效益和社会效益、短期效益和长远利益,都要进行科学的综合分析。结合成本、效益以及软件产品的生命周期等特点,确定软件产品是否值得开发。

(2) 技术可行性。技术可行性分析需要考虑的因素较多,包括技术现状、技术潜力、生产率和风险处理、软件质量等。需要分析当前可利用的技术人员、软硬件资源和技术环境是否支持该方案;是否有足够的技术潜力完成解决方案中新技术所需的条件;在给定的时间、功能和经费限制范围内,能否高效完成提出的所有工作和应对可能产生的风险;在满足用户需求的前提下,该方案开发的软件的质量等级。评估的指标有性能、精确性、可靠性、容错性、效率、兼容性、可理解性、简洁性和可扩充性等。

(3) 社会可行性。社会可行性研究内容主要有知识产权、市场、政策与道德等。软件产品及其开发过程中涉及的实体、技术和资源是否存在任何侵犯、妨碍等责任问题,包括合同、责任、侵权、用户组织的管理模型及规范等。进入未成熟的市场存在高风险,同时也可能带来高收益;成熟的市场的进入风险不高,相应的收益也不高;而即将消亡的市场则没有进入的必要。政策不仅影响软件企业自身的运作,关系软件的开发成本,同时影响产品的收益甚至是成败。研究软件行业以及软件为之服务的行业的相关政策,对于认识软件项目的利润空间乃至生存空间都是必要的。此外,还有必要研究软件开发环境与工作环境的道德情况,避免因为道德矛盾产生的风险。

(4) 操作可行性。操作可行性即研究项目的运行方式在用户组织内是否行得通,现有的管理制度、人员素质和操作方式是否可行。

3.2.2 可行性研究的主要问题

可行性研究实质上是进行一次简化的软件过程,其主要问题集中在投资与效益、技术支持与风险、系统影响以及方案选择等方面。



成本效益分析

1. 投资与效益分析

软件项目投资受项目的特点、规模等多种因素的制约,尤其是其中的软件要素的开发成本在可行性研究阶段很难准确估算。投资与效益分析的目的是从经济角度评价开发一个新的系统是否可行。首先要估算新系统的支出,然后与可能取得的效益进行比较和权衡。

1) 支出分析

对于所选择的方案,说明所需的费用。如果已有一个现存系统,则包括该系统继续运行期间所需的费用。支出主要包含基本建设投资、其他一次性支出以及非一次性支出。

(1) 基本建设投资。包括采购、开发和安装下列各项所需的费用,如房屋和设施;硬件设备,包括服务器、存储器、移动设备等;网络设施;环境保护设备;安全与保密设备;操作系统和应用软件;数据库管理软件。

(2) 其他一次性支出。包含研究(如需求和设计的研究);开发计划与测量基准的研究;数据库的建立;已有软件的修改;检查费用和技术管理性费用;培训费、差旅费以及开发安装人员所需要的一次性支出;人员的退休及调动费用等。

(3) 非一次性支出。列出在该系统生命周期内按月或按季或按年支出的用于运行和维护的费用,包括设备的租金和维护费用;软件的租金和维护费用;数据通信方面的租金和维护费用;人员的工资、奖金;房屋、空间的使用开支;公用设施方面的开支;保密安全方面的开支;其他经常性的支出等。

2) 收益分析

对于所选择的方案,说明能够带来的收益,这里所说的收益,表现为开支费用的减少或避免、差错的减少、灵活性的增加、动作速度的提高和管理计划方面的改进等,包括一次性收益、非一次性收益、不可定量的收益等。

(1) 一次性收益。说明能够用货币数目表示的一次性收益,可按数据处理、用户、管理和支持等项分类叙述,如开支的缩减,包括改进了的系统运行所引起的开支缩减,如资源要求的减少,运行效率的提高,数据进入、存储和恢复技术的改进,系统性能的可监控,软件的转换和优化,数据压缩技术的采用,处理的集中化/分布化等;价值的增升,包括由于一个应用系统的使用价值的增升所引起的收益,如资源利用的改进、管理和运行效率的提高以及出错率的减少等;其他收益,如从多余设备出售回收的收入等。

(2) 非一次性收益。说明在整个系统生命周期内由于运行所建议系统而导致的按月的、按年的能用货币数目表示的收益,包括开支的减少和避免。

(3) 不可定量的收益。逐项列出无法直接用货币表示的收益,如服务的改进,由操作失误引起的风险的减少,信息掌握情况的改进,组织机构给外界形象的改善等。有些不可确定的收益只能大概估计或进行极值估计(按最好和最差情况估计)。

3) 其他分析

其他分析包括收益/投资比、投资回收周期以及敏感性分析。收益/投资比,即求出整个系统生命周期的收益/投资比值;投资回收周期,即求出收益的累计数开始超过支出的累计数的时间;敏感性分析,是指一些关键性因素,如系统生命周期长度、系统的工作负荷量、工作负荷的类型与这些不同类型之间的合理搭配、处理速度要求、设备和软件的配置等变化时,对开支和收益影响的最灵敏的范围估计。在敏感性分析的基础上做出的选择会比单一选择的结果要好一些。

4) 效益的度量

度量效益的方法有以下4种。

(1) 货币的时间价值。货币的时间价值指同样数量的货币随时间的不同具有不同的价值。一般货币在不同时间的价值可用年利率来折算。假设年利率为 i ,如果现在存入 P 元,则 n 年后的价值为 F 元,则有

$$F = P(1 + i)^n$$

如果 n 年后能收入 F 元,这些钱折算成现在的价值称为折现值,折现公式为

$$P=F/(1+i)^n$$

(2) 纯收入。纯收入指在整个生命周期系统的累计收入的折现值 PT 与总成本折现值 ST 之差,以 T 表示,则有

$$T=PT-ST$$

如果纯收入小于或等于 0,则这项工程单从经济角度来看是不值得投资的。

(3) 投资回收期。投资回收期是指系统投入运行后累计的经济效益的折现值正好等于投资所需的时间。投资回收期越短,就能越快地获得利润,工程越值得投资。

(4) 投资回收率。把资金投入项目中与把资金存入银行比较,其中投入项目中可获得的年利率就称为项目的投资回收率。设 S 为现在的投资额, F_i 是第 i 年年初到年底一年的收益($i=1,2,\cdots,n$), n 是系统的寿命, j 是投资回收率,则 j 满足方程

$$S=F_1/(1+j)^1+F_2/(1+j)^2+\cdots+F_n/(1+j)^n$$

解这个方程就可以得到投资回收率 j 。

如果仅考虑经济效益,只有项目的投资回收率大于年利率时,才考虑开发问题。

【例 3-1】 已知一个基于计算机系统的软件升级的开发成本估算值为 5000 元,预计新系统投入运行后每年可以带来 2500 元的收入,假定新软件的生命周期(不包括开发时间)为 5 年,当年的年利率为 12%,试对该系统的开发进行成本-效益分析。

对本题将来的收入折现,计算结果如表 3-1 所示(表中折现值用四舍五入的数据计算)。

表 3-1 将来的收入折算成现在值

n (年)	第 n 年的收入	$(1+i)^n$	折 现 值	累计折现值
1	2500	1. 12	2232.14	2232.14
2	2500	1. 25	1992.98	4225.12
3	2500	1. 40	1779.45	6004.57
4	2500	1. 57	1588.80	7593.37
5	2500	1. 76	1418.57	9011.94

$$T=PT-ST=9011.94-5000=4011.94(\text{元})$$

投资回收期为

$$2+(5000-4225.12)/1779.45=2+0.44=2.44(\text{年})$$

单从经济效益看,投资回收率为 41.04%,投资回收率大于年利率时,可考虑开发。

2. 技术支持与风险分析

技术支持与风险分析明确给出资源分析、技术分析和风险分析的结论,以便使项目管理人员据此做出是否进行系统开发的决策。如果技术风险很大,或者资源不足,或者当前的技术、方法与工具不能实现系统预期的功能和性能,项目管理人员就应及时做出撤销项目的决定。

1) 资源与支持技术分析

资源有效性分析是论证是否具备系统开发所需各类人员的数量和质量、软硬件资源和工作环境等。支持技术分析是论证现有的科学技术水平和开发能力是否支持开发的全过程并达到系统功能和性能的目标。主要包括在当前的限制条件下,该系统的功能目标能否达到;利用现有技术,该系统的功能能否实现;对开发人员的数量和质量的要求及这些要求是否满足;在规定期限内,本系统的开发能否完成等。

2) 风险分析

开发一个软件项目总存在某些不确定性,即存在风险。有些风险如果控制得不好,可能导致灾难性的后果。风险分析就是在给定的约束条件下,论证能否实现系统所需的功能和性能。

风险按影响的范围,可分为项目风险、技术风险和商业风险3类。项目风险是指项目在预算、进度、人力资源、客户和需求等方面可能存在的问题;技术风险是指在需求、设计、实现、接口、验证和维护等方面的潜在问题;商业风险是指开发一个没人需要的优质软件产品,开发一个销售部门不知道如何销售的软件产品,或开发一个不再符合整体商业策略的产品等。

可以使用风险检测表来标识风险。在风险检测表中列出所有可能的与每个风险因素有关的问题。

【例 3-2】“人员风险检测表”如表 3-2 所示。在表中,可以根据实际情况选用 0、1、2、3、4、5 来回答某个问题,某个问题取值越大表示该项风险也越大。人员风险检测表反映了人的因素可能对软件项目的影响。

表 3-2 人员风险检测表

序 号	问 题	回答(0,1,2,3,4,5)
1	开发人员的水平如何?	2
2	开发人员在技术上是否配套?	1
3	是否有足够的人员可用?	0
4	开发人员是否能自始至终参加软件项目的工作?	2
5	开发人员是否能把全部精力投入软件开发工作中?	2
6	开发人员对自己的工作是否有正确的期望?	1
7	开发人员是否已接受了必要的培训?	0
8	开发人员的流动是否还能保证工作的连续性?	3

要对风险进行估算,首先应建立风险度量指标体系,指明风险带来的影响和损失,确定影响风险的因素,估计风险出现的可能性或概率,即进行定量的估算。

【例 3-3】 估算方法举例。

设:某个风险检测表由 m 项组成,每项可在 $0,1,2,\dots,N$ 中根据实际情况选取一个

整数值。其中,0 表示最好的情况, N 表示最差的情况。

又设:第 i 种风险检测表的第 j 项取值为 X_{ij} ,对应的加权系数为 W_{ij} ,则第 i 种风险的估算值可以定义为

$$\sigma_i = \sum W_{ij} X_{ij} / (mN)$$

式中, $\sum W_{ij} = m, W_{ij} \geq 0$ 。

又设:第 i 种风险对整个项目的风险估算的加权系数为 $\rho_i, i = 1, 2, \dots, k$,其中, k 为风险的种类数,且满足 $\rho_1 + \rho_2 + \dots + \rho_k = 1$,则整个软件项目的风险估算值 R 定义为

$$R = \sum \rho_i \sigma_i = \sum \rho_i \left[\sum W_{ij} X_{ij} / (mN) \right]$$

容易验证, $0 \leq R \leq 1$ 。估算的结果,如果 R 接近于 0,说明项目风险比较小;如果 R 接近于 1,说明项目风险比较大。如果 $\rho_i \sigma_i$ 的值比较大,说明第 i 类风险出现的可能性比较大。

常采用三元组 $[r_i, p_i, x_i]$ 来描述风险。其中, r_i 表示第 i 种风险; p_i 表示第 i 种风险发生的概率; x_i 表示该风险带来的影响, $i = 1, 2, \dots, k$,即软件开发项目共有 k 种风险, i 为风险序号。

一个风险评价技术就是定义风险参照水准。对于大多数软件项目,成本、进度、性能就是典型的风险参照水准。在软件开发过程中成本超支、进度拖延、软件性能下降、支持困难,或它们的某些组合,都有一个水准。当软件项目风险的某种组合达到或超过了一个或多个参照水准时,项目就应终止。

例如,在软件开发的过程中,项目的进度应与投入的成本相一致,如果投入的成本与进度的拖延之间超过某个参照水准,项目就应该终止。

【例 3-4】 图 3-2 给出了风险参照水准的参照曲线,当风险的一个组合所引起的成本超支和进度拖延超过参照水准而进入图中的封闭区域时,项目将被迫终止。

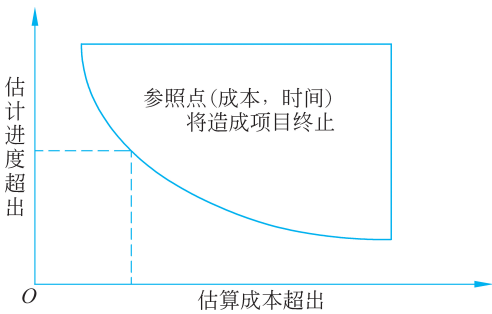


图 3-2 风险参照水准的参照曲线

一般参照点不是一条平滑的曲线,而是一个易变动的区域,在这个区域要做出基于参照值组合的管理判断往往是不准确的。

3. 系统影响分析

在可行性研究中,应当预期所开发系统将带来的影响,主要包括如下 7 方面。

(1) 对设备的影响。说明新提出的设备要求及对现存系统中尚可使用的设备需要做出的修改。

(2) 对软件的影响。说明为了使现存的应用软件和支持软件能够同所建议系统相适应,而需要对这些软件所进行的修改和补充。

(3) 对用户的影响。说明为了建立和运行所建议系统,对用户单位机构、人员的数量和技术水平等方面的全部要求。

(4) 对系统运行过程的影响。说明所建议系统对运行过程的影响,如用户的操作规程;运行中心的操作规程;运行中心与用户之间的关系;源数据的处理;数据进入系统的过程;对数据保存的要求,对数据存储、恢复的处理;输出报告的处理过程、存储媒体和调度方法;系统失效的后果及恢复的处理办法。

(5) 对开发的影响。说明对开发的影响,如为了支持所建议系统的开发,用户需要进行的工作;为了建立一个数据库所要求的数据资源;为了开发和测验所建议系统而需要的计算机资源;所涉及的保密与安全问题。

(6) 对地点和设施的影响。说明对建筑物改造的要求及对环境设施的要求。

(7) 对经费开支的影响。扼要说明为了所建议系统的开发、设计和维持运行而需要的各项经费开支。

4. 方案选择

在可行性研究阶段,系统工程师根据系统分析所确定的系统目标开始研究问题的求解方案。对于较复杂的大系统,一般都要将其分解为若干子系统,接着精确地定义各子系统的界面、功能和性能,给出各子系统之间的关系。分解技术可降低解的复杂性,有利于人员的组织与分工,提高开发生产率和开发质量。

由于系统的分解方法可以有多种,因此实现系统目标的方案也可以有多种。采用的方案不同,对成本、进度、技术及各种资源的要求就会不同,系统在功能和性能方面也可能有较大差异。从另一个角度来看,在系统开发的总成本不变的前提下,系统开发各阶段的成本分配方案的不同也会影响系统的功能和性能。

另外,系统的各功能和性能可能由多种因素组成,而某些因素之间又是相互关联、彼此制约、不可兼得的。例如,系统的计算精度和系统的执行时间就是互相矛盾的。

总之,要选择一个较好的方案,首先要对系统采用多种分解和组合方法提出多种备选的求解方案;其次依据系统的功能、性能、成本、进度、系统开发所采用的技术、风险、软硬件资源、对开发人员的要求等评价每个预选方案,并利用折中手段对预选方案进行充分论证,反复比较各种方案的成本-效益;最后选择出一种较好的方案。

3.2.3 可行性研究的过程

可行性研究的过程是一个逐步深入的过程,一般要经过机会研究、初步可行性研究和可行性研究等活动。机会研究的任务,主要是为开发项目提出建议,寻找最有利的投资机会。许多项目在机会研究之后,还不能决定取舍,需要进行比较详细的可行性研究。



可行性研究的过程

1. 可行性研究的步骤

可行性研究是一个严谨的、科学的论证过程,必须依据当前的技术水平和系统分析人员的经验,按照一定的步骤进行。

(1) 系统定义。系统定义的主要工作是分析问题、明确问题、初步确定问题的范围(系统的边界)、问题的规模(系统的规模)和问题的内容(系统的目标),描述系统的一切限制和约束。

分析人员对问题的提出者和相关人员进行调查访问,仔细阅读和分析有关材料,确保正在分析的问题是用户需要解决的问题。

(2) 对现有系统进行物理建模。物理建模,即建立物理模型。现有系统是信息的重要来源,需要在物理模型中体现其功能、性能、基本架构、相关实体、操作流程、环境、问题和运行消耗,从而评估新系统的运行环境与运行费用。

分析人员实地考察现有系统,收集、研究和分析现有系统的文档资料,考察系统的操作人员和管理人员,使用系统流程图描述现有系统的物理模型。

(3) 构建新系统的逻辑模型。在现有系统物理模型的基础上,逐步明确新系统的功能、相关实体和工作原理(处理流程),并将其反映在新系统的逻辑模型中。

分析人员以现有系统的物理模型为参考,明确新系统的需求、实体和工作原理后,从而设计出新系统的逻辑模型,并非在现有系统物理模型的基础上抽象逻辑模型。逻辑模型使用数据流图和数据字典进行描述。

(4) 设计系统的解决方案。针对系统的逻辑模型,从技术角度出发,根据用户的要求和当前可利用的资源,提出解决方案。一般应分别设计效益优先、技术优先、效率优先和兼顾平衡等多种解决方案。

分析人员应具备快速构建系统的能力,在短时间内对不同解决方案的系统架构、技术框架、解题方法等技术方面的内容进行概要性的描述,同时还必须对每种解决方案进行简单的计划,包括预计时间、生产效率、开发/维护成本、预计收益以及软件产品的质量评估。该步骤是设计开发步骤的预演,解决方案的正确性和精确性,很大程度上取决于分析人员的项目开发经验。

(5) 分析、评估提出的解决方案。对提出的解决方案分别进行技术可行性、经济可行性、社会可行性和操作可行性的分析,去掉不可行的解决方案,保留若干可行的解决方案。一般保留的解决方案中应有分别偏重于效益、技术和效率的方案,同时也必须有兼顾平衡的折中方案。

解决方案的评估不仅仅是系统分析人员的工作,而应该由相关人员参与,尤其是用户。解决方案的取舍以满足用户需求为首要原则,其次必须保证软件产品的质量,最后是效益、工作效率和技术等内部因素,在选择时可分别有所侧重。

(6) 推荐可行的解决方案。根据相关客观条件和主观判断,决定项目是否值得开发。若值得开发,则从上述步骤选择的解决方案中,选择最佳的解决方案,并说明选择该方案的理由。

主要由项目负责人推荐可行的解决方案,在确保用户需求和产品质量的前提下,推荐

解决方案既取决于项目负责人的经验和主观意识,又受到软件开发团队及所在实体的外在因素(例如,企业文化、企业经营状况、工作指导方针等)的影响。

(7) 撰写可行性研究报告。将可行性研究的任务、过程和结果按照固定格式撰写可行性研究报告。提请用户和上级部门进行审查,从而得出可行性研究的最终结果。

2. 可行性研究报告

可行性研究报告的编写目的,是说明该软件开发项目的实现在技术、经济和社会条件方面的可行性,评述为了合理地达到开发目标而可能选择的各种方案,说明并论证所选定的方案。

可行性研究报告的编写内容要求如下。

(1) 引言。包括报告编写目的,所建议开发的软件系统的背景,报告中引用的专门术语的定义和外文首字母组词的原词组,以及用得着的参考资料。

(2) 可行性研究的前提。说明对所建议的开发项目进行可行性研究的前提,如要求、目标、条件、假定和限制,进行可行性研究的方法,对系统进行评价时所使用的主要尺度等。

(3) 对现有系统的分析。需要说明现有系统的处理流程和数据流程、工作负荷、费用开支、运行和维护所需要的人员的专业技术类别和数量、所使用的各种设备、系统的主要局限性等。

(4) 所建议的系统。说明所建议系统的目标和要求将如何被满足,包括对所建议系统的说明、处理流程和数据流程、改进之处,预期将带来的影响,以及技术条件方面的可行性。

(5) 可选择的其他系统方案。扼要说明曾考虑过的每种可选择的系统方案。

(6) 投资与效益分析。对于所选择的方案,说明所需的费用和能够带来的收益。支出包括基本建设投资,其他一次性支出,在该系统生命周期内按月或按季或按年支出的用于运行和维护的费用;收益表现为开支费用的减少或避免、差错的减少、灵活性的增加、动作速度的提高和管理计划方面的改进等,包括一次性收益、非一次性收益、不可定量的收益。求出整个系统生命周期的收益/投资比值,收益的累计数开始超过支出的累计数的时间。

(7) 社会因素方面的可行性。说明对社会因素方面的可行性分析的结果,包括法律方面的可行性、使用方面的可行性。

(8) 结论。在进行可行性研究报告的编制时,必须有一个研究的结论。结论应明确指出以下内容:系统具备立即开发的可行性,可进入软件开发的下个阶段;若可行性分析结果完全不可行,则软件开发工作必须放弃;不具备某些条件,可以创造条件,增加资源或改变新系统的目标后,再重新进行可行性论证。

3.3 软件项目计划

一个软件项目经过可行性分析后,若认为值得开发,则应制订项目开发计划。软件项目计划主要进行的工作包括确定详细的项目实施范围,定义递交的工作成果,评估实施过

程中主要的风险,以及制订项目实施的时间计划、成本和预算计划、人力资源计划等。

3.3.1 软件项目计划概述

项目计划(Project Plan)是根据对未来的项目决策,项目执行机构选择制定包括项目目标、工程标准、项目预算、实施程序及实施方案等的活动。在一个具体的项目环境中,项目计划是预先确定的行动纲领。制订项目计划旨在消除或减少不确定性,改善工作效率,对项目目标有更好的理解以及为项目监控提供依据。

1. 项目计划的作用

古人云“凡事预则立,不预则废”,事先有准备,就能得到成功,否则就会失败。足以说明计划的重要性。在软件项目管理中也当然是计划先行,做好计划是软件项目成功实施的基础。

项目计划是项目组为实现项目目标而科学地预测并确定项目生命周期的行动方案。项目计划围绕项目目标的完成,系统地确定项目的任务、安排任务进度、编制完成任务所需的资源预算等,从而保证项目能够在合理的工期内,用尽可能低的成本达到尽可能高的项目质量要求。

项目计划所起到的作用:确定完成项目目标所需的各项任务范围,落实责任,制订各项任务的时间表,明确各项任务所需的人力、物力、财力;确定项目的工作规范,遵循的标准,成为项目实施的依据和指南;明确项目组各成员及其工作责任范围以及相应的职权;使项目组成员明确自己的工作目标、工作方法、工作途径、工作期限要求;保证项目进行过程中项目组成员和客户之间的交流、沟通与协作,使得项目各项工作协调一致,增加客户满意度;为项目的跟踪控制提供基础。

项目计划在项目中起到承上启下的作用,计划批准后应当作为项目的工作指南。

【例 3-5】 软件开发项目失败的背景和原因很多,但共通性的原因有:计划方案不好;没有按照计划执行;主要管理人员未参加;项目管理人员、项目领导的运营管理水平低。美国联邦调查局进行了 150 例调查,开发项目失败的原因是由于计划不完备的占 50%,不按计划进行管理的占 33%,其他原因占 17%。由此可知,重视计划的编制,加强工程管理,有利于确保软件项目的开发成功。

2. 项目计划的制订原则

要使项目计划得以顺利实现,必须明确项目目标,综合分析考虑各因素,权衡利弊,扬长避短。在项目计划制订过程中一般应遵循以下原则。

(1) 目的性。任何项目都有一个或几个确定的目标,以实现特定的功能、作用和任务,而任何项目计划的制订正是围绕项目目标的实现而展开的。

(2) 系统性。项目计划本身是一个系统,由一系列子计划组成,各个子计划不是孤立存在的,而是彼此相对独立,又紧密相关。这使得制订出的项目计划也具有系统的目的性、相关性、层次性、适应性、整体性等基本特征,使项目计划形成有机协调的整体。

(3) 动态性。这是由项目的生命周期所决定的。一个项目的生命周期短则数月,长则数年,在这期间项目环境常处于变化之中,因此项目计划要随着环境和条件的变化而不断调整和修改,以保证完成项目目标。

(4) 相关性。项目计划是一个系统的整体,构成项目计划的任何子计划的变化都会影响其他子计划的制订和执行,进而最终影响项目计划的正常实施。制订项目计划要充分考虑各子计划间的相关性。

(5) 职能性。项目计划的制订和实施不是以某个组织或部门内的机构设置为依据,也不是以自身的利益及要求为出发点,而是以项目和项目管理的总体及职能为出发点,涉及项目管理的各个部门和机构。

(6) 可操作性。可操作性包括范围的适中及可理解程度。项目计划如果只有很少的细节,就不可能取得比较精确的估计;如果项目计划包含太多的细节,就会超出项目经理所控制的范围,使其无所适从。如果任务在执行前就有了较好的理解,许多工作就能提前进行准备;如果任务是不可理解的,在实际执行中就比较难于操作。

3. 项目计划的制订策略

制订软件项目计划的目的在于建立并维护软件项目各项活动的计划,软件项目计划其实就是一个用来协调软件项目中其他所有计划,指导项目组对项目进行执行和监控的文件。一个好的软件项目计划可为项目的成功实施打下坚实的基础。

软件项目有其特殊性,不确定因素多,工作量估计困难,项目初期难于制订一个科学、合理的项目计划。制订软件项目计划时以下策略可供参考。

(1) 项目计划的层次性。项目计划可分层次展开,如分为总体计划和详细计划。总体计划应当是粗粒度的,主要是进行项目的阶段划分,确定重大的里程碑,所需相关的资源包括人力资源、设备资源、资金资源,即人、财、物三个要素;详细计划要确定各项任务的负责人、开始时间、结束时间、任务之间的依赖关系、设备资源、事件点等。

如果项目规模相对较大,可以有多级的计划,例如,高级计划、二级计划、三级计划、低级计划等。一个项目组可能分为几个开发组,二级计划是各开发组制订的适合自己小组的计划。如果开发组还分了小组,可以有小组的三级计划。开发人员的个人计划是低级计划,由开发人员根据自己的任务自行制订。

(2) 重视与客户的沟通。与客户的沟通是非常重要的。用户会提出一些对项目时间、进度、效果的要求,要站在科学的分析和解决问题的立场与客户沟通,争取用户的理解与支持。开发者也有义务要让用户知道项目的计划,这样才能让用户主动、积极参与项目,达到项目的最终目标。既让用户感觉心里踏实,也让项目组具有责任感,有利于促进项目的成功。

(3) 繁简适度。目标的描述应当是简单而直接的,使得每个参与人员都能明确而无二义性。项目的工作安排一定要责任到人,成本、时间安排尽可能详尽,这些是必须详细的。在项目计划中可采用图表的形式制订人员计划、培训计划、风险计划、成本估计、文档大小估计、进度计划,一目了然,责任到人,其效果和效益是很明显的。

(4) 切合现实。确定的每个目标都是可以实现的,而不是追求理想化的结果。项目

计划最终要能够被项目组成员所实现。

(5) 利用成熟的项目管理工具。项目管理工具是为了使工作项目能够按照预定的成本、进度、质量顺利完成,而对项目要素进行分析和管理的—类软件。例如,Microsoft Project 是一款公认的功能强大、操作方便的项目管理工具软件。它自带一个“软件开发”模板,可以用它来生成大体的框架,再进行细节方面的改动,也可以自己制作一个符合软件项目运作流程的模板。

3.3.2 软件项目计划基础工作

软件项目计划的目标是为项目组提供一个框架,使之能合理地估算软件项目开发所需的资源、经费和开发进度,并控制软件项目开发过程按此计划进行。在做项目计划时,必须首先对软件规模做出估算。

1. 工作分解结构

工作分解结构(Work Breakdown Structure,WBS)是以可交付成果为导向,对项目要素进行的分组,它归纳和定义了项目的整个工作范围,每下降一层代表对项目工作的更详细定义。工作分解结构将项目分解为可管理的任务及活动,可帮助项目相关成员直观了解软件项目中的各项任务及活动,是项目计划与跟踪的基础。图 3-3 是一个典型的工作分解结构。

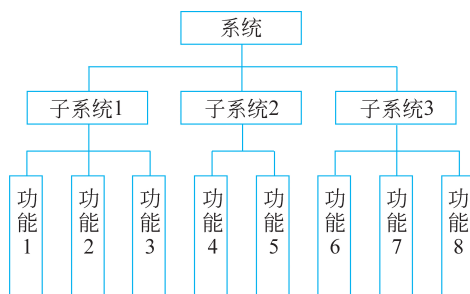


图 3-3 典型的工作分解结构

工作分解结构的构建应该注意的原则如下:一个任务只应该在工作分解结构中的一个地方出现;工作分解结构中某项任务的内容是其下所有工作分解结构项的总和;一个工作分解结构项只能由一个人负责任,其他人只能是参与者;工作分解结构必须与实际工作中的执行方式一致;应让项目团队成员积极参与创建工作分解结构,以确保工作分解结构的一致性;每个工作分解结构项都必须文档化,以确保准确理解已包括和未包括的工作范围;工作分解结构可以根据需求进行必要变更维护。

构建工作分解结构的过程:在确定项目范围的基础上,召集有关人员讨论所有主要项目工作,确定项目工作分解的方式;分解项目工作,画出工作分解结构的层次结构图;将主要项目可交付成果细分为更小的、易于管理的组分或工作包;验证上述分解的正确性。随着其他计划活动的进行,不断地对工作分解结构进行更新或修正,直到覆盖所有工作。

【例 3-6】 图 3-4 是开发项目各个工程阶段的任务分解逐步细化的例子。

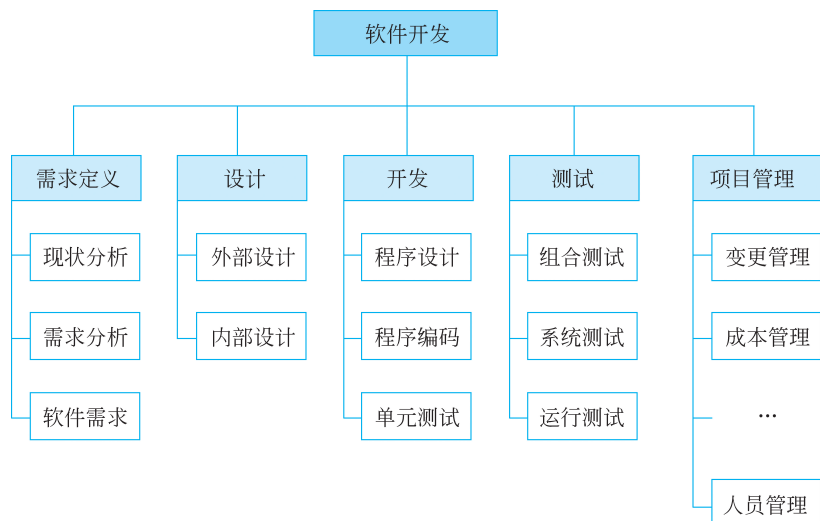


图 3-4 软件开发的工作分解结构示例

工作分解结构描述了项目的工作范围,它使人们能够清楚地知道整个项目要做些什么工作,以及项目的可交付成果是通过开展哪些工作生成的。工作分解结构不但是项目工作的客观描述,而且是后续项目估算、进度计划和跟踪考核的基本单位。

2. 软件规模估算

软件项目估算是软件计划和管理中非常重要的问题。在项目计划中,项目时间和成本的估算值是必不可少的,但是如果能够知道软件规模和工作量的估算值,时间表、成本预算就会变得容易。所以,软件规模估算是任何一个软件项目最困难和最重要的活动之一。

软件规模是确定到底需要付出多少工作量才能创建出软件的主要因素。当在构想一个软件项目时,实际软件的规模是未知的,而软件本身也是不存在的。如果需要采用规模来作为工作量估算模型的输入信息,首先必须对规模进行估算。

最初的软件规模评估主要依靠专家的经验来完成,评估过程较慢,而且不同的专家给出的评估结果可能会有较大的偏差。之后,业界通过代码行数来衡量软件规模。但是,用代码行数估算软件规模的方法只能在软件开发完成以后才能进行计算,由于不同的程序设计语言、不同的开发团队实现同样功能的代码行数有很大差异,导致评估结果也有很大偏差。因此,研究人员开始寻求更精确的、更易于使用的估算方法。

在众多估算方法中,构造性成本模型(Constructive Cost Model, COCOMO)是其中最有影响力的模型,代码行数仅作为它的一个模型参数。为了克服代码行数方法依赖于程序设计语言的缺点,业界应用了功能点估算法。功能点估算法通过量化系统功能来度量软件规模、估算软件成本,它基于系统的逻辑设计,从用户角度分析,在整个生命周期都能进行估算。随着统一建模语言(UML)在软件系统中的广泛应用,基于 UML 的规模度量方法应运而生,其中最为典型的是用例点估算法。用例点估算方法是一种根据软件开



软件项目
估算

发前期的 UML 用例图,从开发人员角度分析的评估方法。近年来,基于人工智能和机器学习的方法也被逐渐应用于软件规模评估。

不同的软件规模评估方法对软件项目评估的结果也会有偏差,目前还没有一种通用的评估方法能够很好地适用于各类软件项目在各个软件工程阶段的估算。下面简要介绍应用广泛的 6 种方法。

1) 专家决策法

在传统的软件规模评估中,通常会依靠专家经验,即专家决策法。它是一种由相关技术专家根据历史数据和资料以及个人经验给出评估结果,并且通过组织讨论后达成共识,最终得到软件规模、工作量、工期和成本的评估方法。

专家决策可以调和各类专家的意见,在历史数据集缺失或不足时也能进行评估。但是专家模型非常依赖于专家的经验,软件项目评估的精确性取决于专家的能力。事实上,并不能保证每个软件项目团队都有足够多的专家,并且专家的经验也很难度量,所以专家决策法很容易受专家主观因素的影响,并不是一种非常有效的评估方法。

专家决策法依据预定的程序,采用匿名发表意见的方式,即专家之间不得互相讨论,不发生横向联系,通过多轮次调查专家对问卷所提问题的看法,经过反复征询、归纳、修改,最后汇总成专家基本一致的看法,作为预测的结果。专家决策法实施步骤如图 3-5 所示。

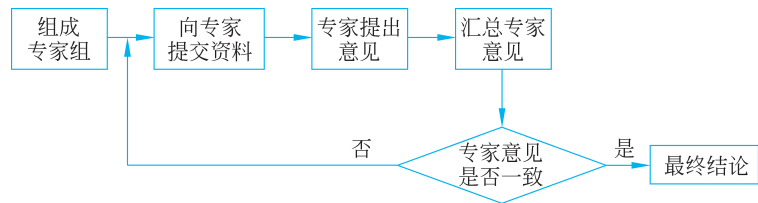


图 3-5 专家决策法实施步骤

2) 类比估算法

类比估算法又称自顶向下估算法(Top Down Estimates),适合评估一些与历史项目在应用领域、环境和复杂度方面相似的项目,通过新项目与历史项目的比较得到规模估计。类比估算法估计结果的精确度,取决于历史项目数据的完整性和准确度,因此,用好类比估算法的前提条件之一是组织建立起较好的项目后评价与分析机制,对历史项目的数据分析是可信赖的。

软件项目中用类比估算法,往往还要解决可重用代码的估算问题。估算可重用代码量的最好办法就是由程序员或系统分析员详细地考察已存在的代码,估算出新项目可重用的代码中需重新设计的代码百分比、需重新编码或修改的代码百分比以及需重新测试的代码百分比。根据上述 3 个百分比,可用下面的计算公式计算等价代码行,即

等价代码行 = $[(\text{重新设计}\% + \text{重新编码}\% + \text{重新测试}\%) / 3] \times \text{已有代码}$

【例 3-7】 有 10 000 行代码,假定 30%需要重新设计,50%需要重新编码,70%需要重新测试,那么其等价代码行可以计算为

等价代码行 = $[(30\% + 50\% + 70\%) / 3] \times 10\ 000 = 5000$

即重用这 10 000 行代码相当于编写 5000 代码行的规模。

3) COCOMO 法

著名软件工程专家、经济学家巴利·W.玻姆(Barry W.Boehm)在其著作《软件工程经济学》中提出了软件估算模型层次结构,称为 COCOMO,该模型已经成为软件界通用的估算模型。

COCOMO 是一种将软件特征作为参数的估算模型,分为基本、中间和详细三级。基本 COCOMO 将代码行数作为自变量计算软件开发的工作量,可用于估算整个系统的工作量;中间 COCOMO 在基本 COCOMO 的基础上,增加了工作量调节因子(Effort Adjustment Factor,EAF),对涉及产品、硬件、人员、项目等因素的调整工作量进行估算,可用于估算各个子系统的工作量;详细 COCOMO 是对中间 COCOMO 的细化,按开发周期的不同阶段给出工作量因素分级表,可以估算系统、子系统和模块 3 个层次中每个阶段的工作量。

1995 年,玻姆等进一步提出了 COCOMO II 模型。COCOMO II 模型由应用组合模型、早期设计模型及后体系结构模型 3 个子模型组成。应用组合模型基于对象点(Object Point)对采用集成计算机辅助软件工程工具快速应用开发的软件项目工作量和进度进行估算,用于项目规划阶段;早期设计模型基于功能点(Function Point,FP)或可用代码行,以及 5 个规模指数因子、7 个工作量乘数因子,选择软件体系结构和操作,用于信息还不足以支持详细的细粒度估算阶段;后体系结构模型发生在软件体系结构完好定义和建立之后,基于源代码行或功能点,以及 5 个规模指数因子、17 个工作量乘数因子,用于完成顶层设计和获取详细项目信息阶段。

COCOMO 法的优点是因子比较细化,将影响工作量的因素都进行了分类和等级划分,包括产品因素、平台因素、人员因素和项目因素等,因而便于计算。但是 COCOMO 是基于国外数据库 161 个数据的经验得到的,特定条件下各个因子之间的内在关系还没有完全明确,所以不能适应软件行业精细化程度越来越高的要求。

4) 功能点估算法

相对于类比估算法侧重于从开发者角度分析软件的规模,功能点估算法更侧重于从用户的角度分析软件的规模。

功能点估算法的主要步骤:确定项目类型,识别项目范围和边界,根据事务类型功能点以及数据类型功能点估算法进行未调整功能点(Unadjusted Function Points,UFP)分析,即确定加权因子并计算未调整的功能点数,同时确定功能点调整因子,最后计算得出交付功能点数(软件调整规模),如图 3-6 所示。

在功能点分析中,任何一个软件都被看作是由外部输入(External Input,EI)、外部输出(External Output,EO)、外部查询(External Query,EQ)、内部逻辑文件(Internal Logical File,ILF)和外部接口文件(External Interface File,EIF)5 种要素组成。

用功能点分析确定软件的未调整功能点数,需要分 3 个步骤进行。

(1) 确定功能点的计算范围。识别应用程序边界的规则是不能从开发者即技术角度去看,而必须从用户的角度来定义;如果项目牵扯到多个系统,那么必须将这多个系统的边界全部描述清楚。由于海洋要素的特殊性,海洋测绘软件数据处理量庞大。以海图一