

进度管理是 IT 项目管理中非常重要的环节之一。进度计划是为完成项目目标的各类活动设定日期,并进行逻辑排序。进度计划要明确项目的开始时间和结束时间,以及其中每个活动的起止时间,并合理配置项目活动的所需资源。良好的项目进度管理对于成功地完成项目开发工作具有重要意义。

5.1 IT 项目进度管理概述

5.1.1 IT 项目进度管理定义

IT 项目进度管理是指项目在分析、设计、编码、测试以及交付用户过程中,对各阶段活动的进程实施管理,是在规定的项目起止时间内,制订合理且经济的进度计划,确保项目按时完成所需的过程。进度问题是项目冲突的主要原因。时间是一种特殊的资源,具有单向性、不可重复性、不可替代性,时间是项目规划中灵活性最小的因素,能否按进度交付是衡量项目是否成功的重要标志之一。项目进度计划是跟踪和沟通项目进展状态的依据,软件项目进度管理的主要目标是:在给定的条件下,通过分析项目需求,合理安排项目资源,以最短时间、最低成本、最优质量完成用户满意的软件产品。

做好进度管理要求 IT 项目经理在项目开发前,充分了解软件项目的需求,将其明确划分成各项子任务,识别出关键任务;掌握项目团队成员的技术能力水平,协调项目资源分配;有效跟踪、监控项目的实际进展,若发现实施过程偏离了计划,及时找出原因,并实施风险规避措施。

5.1.2 进度管理的主要活动

PMBOK 6 及软件分册定义的进度管理包括 6 个过程,见表 5-1。

(1) 规划进度管理:为规划、编制、管理、执行和控制项目进度而制定政策、流程和文档的过程;这个过程的主要输出是项目进度管理计划。

(2) 定义活动:识别项目成员和干系人为完成项目必须执行的特定活动;活动(或任务)是构成项目开发阶段工作的基本要素,包括活动的历时(工期)、成本和资源要求;这个过程的主要输出是活动清单、活动属性和里程碑清单。

(3) 排列活动顺序:识别并记录项目活动之间的时序关系;这个过程的输出包括项目进度网络图、项目文件更新、软件项目特性集、软件项目发布计划(尤其是敏捷开发)、软件构架和非功能性需求的依赖关系。

(4) 估算活动持续时间:估计完成项目各类计划活动所需的时间;这个过程主要输出包括活动持续时间估算和项目文件更新。

续表

活动	启动 (Initiating)	规划(Planning)					执行(Executing)	监控(Controlling)		收尾 (Closing)
		规划进度管理	定义活动	排列活动顺序	估算活动持续时间	制订进度计划		控制进度		
工具和技术		1. 专家判断 2. 分析技术 3. 会议	1. 专家判断 2. 分解 3. 滚动式规则 4. 会议 5. 故事分解结构(软件分册) 6. 故事板(软件分册) 7. 用例(软件分册)	1. 紧前关系绘图法 2. 确定和整合依赖关系 3. 提前量和滞后量 4. 项目管理信息系统 5. 时间盒(软件分册) 6. 进行中限制与服务等级(软件分册) 7. 特性集评估(软件分册) 8. 服务等级协议(软件分册)	1. 专家判断 2. 类比估算 3. 参数估算 4. 三点估算 5. 自上而下估算 6. 数据分析 7. 决策 8. 会议	1. 进度网络分析 2. 关键路径法 3. 资源优化 4. 数据分析 5. 提前量与滞后量 6. 进度压缩 7. 项目管理信息系统 8. 敏捷发布规划 9. 增量式产品规划(软件分册)		1. 数据分析 2. 关键路径法 3. 项目管理信息系统 4. 资源优化 5. 提前量与滞后量 6. 进度压缩 7. 循证审查(软件分册) 8. 回顾(软件分册) 9. 累计流量图(软件分册) 10. 带有日常演练的工作流公告板(软件分册) 11. 重新确定优先级审查(软件分册) 12. 燃耗图与燃尽图(软件分册) 13. 方差分析(软件分册)		

(5) 制订进度计划：通过活动的顺序分析、活动资源估算和活动工期估算来创建项目进度；这个过程主要输出包括进度基准、项目进度计划、进度数据、项目日历、项目管理计划更新、项目文件更新、发布和迭代计划更新。

(6) 控制进度：控制和管理项目进度的变更；这个过程主要输出包括工作绩效信息、进度预测、变更请求、项目管理计划更新、项目文件更新、组织过程资产更新、其他输出(软件分册)。

5.2 规划进度管理

进度计划是时间管理的基础。规划进度管理是根据项目管理计划和项目章程(规定了项目的起止时间)、组织的环境因素、组织过程资产情况,充分考虑软件项目的安全性,项目经理通过专家判断、分析技术和会议来开发进度管理计划。

项目范围说明书和工作分解结构(WBS),可用于定义活动、持续时间估算和进度管理。因此,范围管理计划是项目管理计划中制订进度管理计划的主要信息来源之一;另外,项目管理计划中的其他信息也是制订进度计划的依据。例如,与规划进度相关的成本计划、人力资源计划、风险计划和沟通计划决策等。

项目章程中规定了项目里程碑进度计划和项目的审批要求。事业环境因素是能影响进度管理的软件组织文化和结构、资源可用性、人力资源技能水平、组织中的工作授权系统、组织用于进度规划工具的项目管理软件是否有利于设计管理进度的多种方案、软件项目组合等。组织过程资产指影响规划进度管理过程的进度控制工具、组织与进度控制有关的政策和流程及指南、模板、变更控制程序、风险控制程序、项目收尾指南、项目的生命周期等。

公共安全和网络安全问题是指为了满足软件项目开发期间的安全标准、规定、政策和特殊要求,其可能会影响项目活动工序的安排,从而影响规划进度管理。

专家判断是基于某应用领域、知识领域、学科、行业等的专业知识,根据以往类似项目历史信息 and 当前项目环境信息,提出制订进度管理计划建议。分析技术指在规划进度管理过程中,需要选择项目进度估算和规划的战略方法,例如,进度规划方法论、进度规划工具与技术、估算方法、格式和项目管理软件。进度管理计划中还需详细描述对项目进度进行快速跟进或赶工的方法,如活动的并行开发工作。

进度管理是通过对项目活动的跟踪和监控实现的。项目进度管理是随着项目的进行而持续进行的,是一个动态过程。

根据项目管理的需求,进度管理计划可以是正式或非正式的,非常详细或高度概括的。通常,进度管理计划涵盖的信息如下。

(1) 进度管理计划的方法,如关键路径法、关键链法等。

(2) 识别项目所使用的进度工具,如 IT 进度管理软件、挣值软件等。

① 精准度: 估算数据的准确程度,如各项活动的持续时间的偏差程度。另外,准确程度可以随着时间演变,如果把滚动规划初期活动(或开发阶段)的进度精度到月,但随着项目进展项目后期的活动估算可以精确到周。

② 计量单位: 持续时间的估算单位,如天、周和月等。

③ 偏差临界值: 指出活动估算时间的偏差值,以进度估算的绝对值或基准计划的百分比表示。

(3) 进度报告信息和格式。

记录活动状态和进展报告所需要的进度信息,如果要使用特殊的报告格式,需要约定模板。

(4) 进度计划的过程管理。

① 活动识别。描述活动识别方法,如分解、头脑风暴和访谈等。

② 活动排序。描述活动排序、创建网络图的准则和依赖关系的活动。

③ 估算资源。指出资源进入或退出活动的时间。

④ 估算人力投入和持续时间。指出用于估算人力投入和持续时间的技术。例如,类比估算、三点估算和参数估算等。

⑤ 更新、管理和控制。记录更新进度的过程,包括更新频率、权限、版本等,支持维护基准完整和必要时重设基准的准则。

5.3 活动定义

5.3.1 项目活动的有关概念

项目是由多个相互关联的活动组成,范围计划中定义了工作分解结构(WBS),给出项目底层工作包,即最小成果交付单元,而活动定义是识别和记录为完成项目可交付成果而需采取的具体行动的过程。

活动定义是一个过程,项目经理可以利用活动定义的主要输入信息以及企业环境因素和组织过程资产,开始制定详细的活动清单,鉴别活动属性,列出里程碑清单,确保项目团队成员能够彻底理解所从事的项目范围内的工作,以便于制定工作进度。

(1) 活动清单是项目进度中的活动列表,包括活动标识(编码)、活动名称以及活动简述。

(2) 活动属性主要描述活动与进度的相关信息,例如,前置任务、后续任务、活动间的逻辑关系、提前和滞后、约束条件、资源需求、强制日期以及与活动相关的假设等。活动清单和活动属性应该与 WBS 及 WBS 字典保持一致。

(3) 项目里程碑是项目中的重大事件,是项目过程中的一个时间节点,不占资源,通常指一个可交付成果的完成。项目里程碑定义的核心是围绕事件、活动、检查点或决策点,以及可交付成果这些概念来展开的。里程碑与常规的进度活动类似,有相同的结构和属性,但不是每个可交付成果或事件都为项目创建一个真正的里程碑,里程碑代表的是一个时间点,持续时间为零,里程碑是重要的可见事件。里程碑清单列出了所有项目里程碑,并标明里程碑是强制性的(如合同要求)还是选择性的(如根据历史信息确定)。

5.3.2 活动定义的方法

活动定义是将工作包分解为活动,作为对项目工作进行估算、进度规划、执行、监督和控制的基础,以便实现项目目标。WBS 中最底层的可交付成果,即工作包;工作包应进一步细分为更小的组成部分,即“活动”,通过这些“活动”来完成相应的可交付成果。活动清单和活动属性均与 WBS 保持一致。项目团队中常使用自动化系统追踪活动相关信息。

1. 基于活动的方法(生命周期)

根据软件项目的生命周期阶段,通过头脑风暴法和对组织开发的历史类似项目分析,得到项目所有活动的列表。WBS 和 WBS 词典是制定最终活动清单的基础。WBS 中的每个工作包都需分解成活动,最底层的活动构成活动列表,而高层节点(或中间层)构成项目的活动集,描述了活动的隶属关系。

根据图 5-1 中的系统设计向下分解为数据库设计,数据库设计分为数据库概念结构设计、

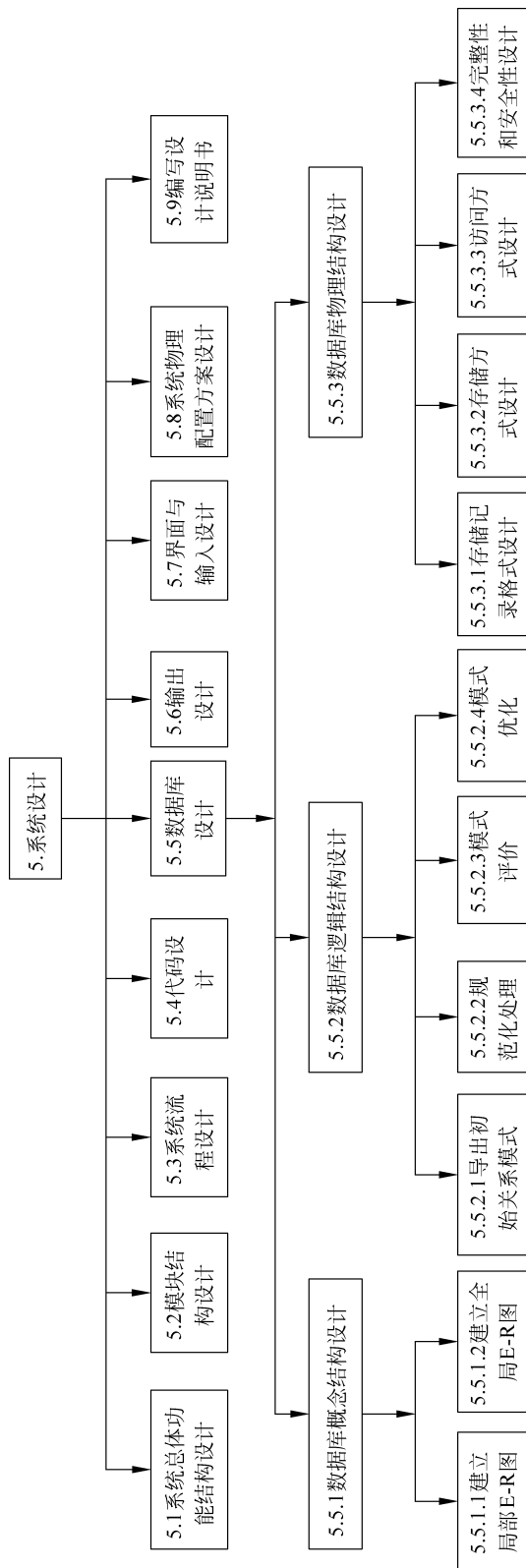


图 5-1 基于生命周期的活动定义图

数据库逻辑结构设计和数据库物理结构设计三个工作包,每个工作包向下分解为“活动”。其中,概念结构设计工作包分为建立局部 E-R 图、建立全局 E-R 图两个活动;逻辑结构设计工作包分为导出初始关系模式、规范化处理、模式评价、模式优化 4 个活动;物理结构设计工作包分为存储记录格式设计、存储方式设计、访问方式设计、完整性和安全性设计 4 个活动。

2. 基于产品的方法(可交付成果)

根据 WBS 底层工作包确定的可交付成果,按照软件工程方法确定完成可交付成果所需的开发活动。图 5-2 中的“2.2 出库管理”工作包可分解的开发活动为:输出信息分析、出库功能项设计、输入界面设计、数据库表物理实现、编写程序代码和单元测试。

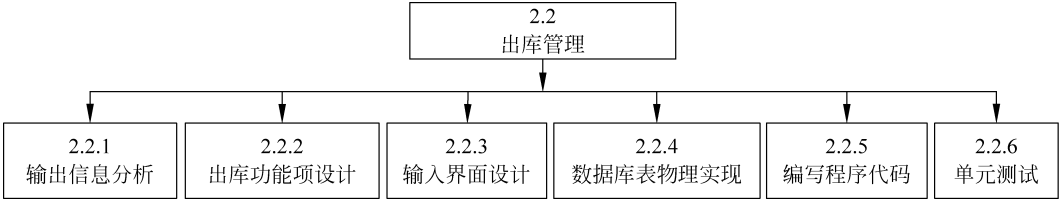


图 5-2 基于产品的活动定义图

滚动式计划是一种迭代式计划技术,即详细规划近期要完成的开发工作,同时在较高层级上粗略规划远期工作,滚动式规划是一种渐进明晰的规划方式。

IT 项目开发是渐进明晰的动态过程,在生命周期的不同阶段,工作的详细程度会有所不同。在早期的系统规划阶段,对于软件需求信息尚不够明确,工作包和活动只能分解到已知的详细水平;随着项目的进展,项目的特征信息逐渐清晰,WBS 越来越准确,项目的工作包越来越具体,则完成工作包的活动就越来越完善。因此,在制订项目范围计划、工作分解结构和项目进度计划时,要广泛地吸收具有经验和技能的项目团队成员或其他专家,为定义活动提供专业知识。

5.4 活动排序

活动定义之后,软件项目进度管理需要确定项目活动的时序关系。通过对项目范围说明、组织过程资产、里程碑事件、软件产品构成分析,找出项目活动之间的依赖关系、特殊领域的依赖关系和工作顺序。常见的活动排序工具包括甘特图、网络图、里程碑图和资源图等。

5.4.1 活动关联关系

1. 活动间的关系

活动排序需要先确定活动间的相关性与活动的实施顺序。紧前活动是在进度计划的逻辑路径中,排在非开始活动前面的活动;紧后活动是在进度计划的逻辑路径中,排在某个活动后面的活动。活动间关系主要有以下 4 种情况,如图 5-3 所示。

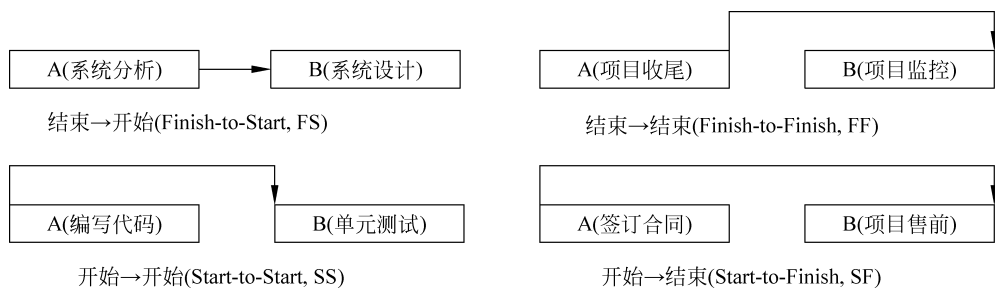


图 5-3 项目各活动间关系

其中,FS 表示 A 活动结束之后 B 活动才能开始,其中 A 是 B 的前置活动,也称紧前活动,B 是 A 的后续活动,也称紧后活动;SS 表示 A 活动开始之后 B 活动才能开始;FF 表示 A 活动结束以后,B 活动才能结束;SF 表示 A 活动开始之后,B 活动才能完成(一般 A 任务是交接 B 任务)。在 4 种活动间关系中,FS 是最常见的关系,SF 是最少使用的关系。

2. 活动间关系的依据

活动间依赖关系可能来源于软件项目的内部或外部;从活动逻辑关系来看,可以是强制或选择。这四种依赖关系可以组合成强制性外部依赖关系、强制性内部依赖关系、选择性外部依赖关系和选择性内部依赖关系。

1) 强制性依赖关系

强制性依赖关系是活动间自身固有的、无法改变的逻辑关系,往往是由客观规律或者活动内在性质决定的依赖,如法律和合同要求的活动关系。强制性依赖关系一般是不可调整、不可违背的。例如,在 IT 项目开发过程中,必须在编写出代码之后才能进行单元测试。

2) 选择性依赖关系

选择性依赖关系是可以由项目开发人员主观意志去调整和确定的项目活动关系,又称为首选逻辑关系、优先逻辑关系或软逻辑关系,是两项活动可先可后的顺序关系。开发人员根据以往项目的经验和知识,识别选择性依赖关系,并进行全面记录,因为它们会影响总浮动时间,并限制后续的进度安排。如果打算进行快速跟进,则应当审查相应的选择性依赖关系,项目经理确定哪个模块先开发,哪个模块后开发,哪个任务先做好些,哪个同时开发好些。

3) 外部依赖关系

外部依赖关系是项目活动与非项目活动之间的依赖关系,如软件安装试运行,取决于用户方是否具备软硬件环境。在排列活动顺序过程中,项目管理团队应明确哪些依赖关系属于外部依赖关系。

4) 内部依赖关系

内部依赖关系是项目活动之间的紧前关系,通常在项目团队的控制之中,如只有先编写程序代码,才能进行程序测试。

5.4.2 进度管理工具

为便于管理项目活动,直观展示软件项目活动间的关系,一般通过进度管理模型,反

映活动的时序和资源的关联。常用的进度管理工具包括甘特图、里程碑图、网络图、资源图等。

1. 甘特(Gantt)图

甘特图,又称横道图。其中,纵轴表示项目活动序列,横轴表示起止时间、先后次序以及各项活动的资源信息,如图 5-4 所示。甘特图简明直观,为团队提供易于理解的沟通工具。甘特图适合于简单的项目,由于不能明显体现各项活动之间的依赖关系、各项活动的主次关系及重要程度等,因此不适合活动逻辑关系复杂的项目。常常借鉴项目管理软件,绘制甘特图,辅助项目进度管理实施。

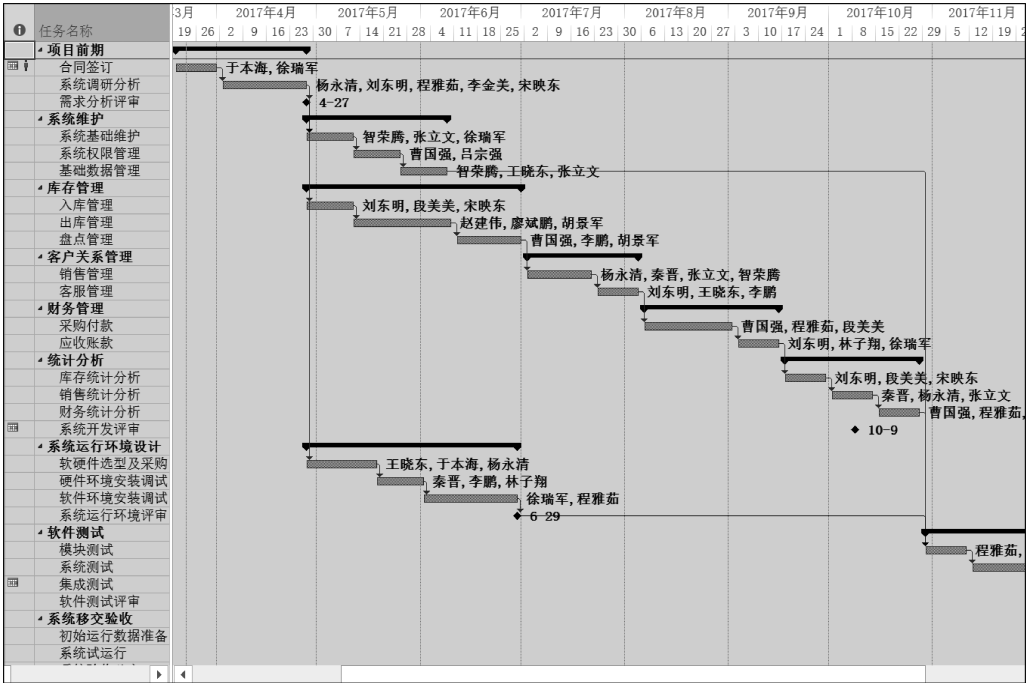


图 5-4 Project 2013 生成的进销存管理信息系统项目甘特图

2. 里程碑图

里程碑是项目中较为重大的事件或时间节点,不占用项目资源和时间。通常指项目特定事件、活动、检查点(或决策点)以及一个可交付成果的完成,即工期为零的活动。

软件项目里程碑通常以生命周期开发阶段为节点,以大的开发阶段估算活动所需时间。里程碑计划编制通常是从完成项目的最后一项活动,即项目的最后一个里程碑反向设置的;由里程碑事件组成的图示即为里程碑图,用黑菱形符号代表里程碑,如图 5-4 所示。需求分析评审、系统开发评审、系统运行环境评审活动的完成,是该开发阶段工作结束标志,在某一时刻发生,工期为 0 工作日,均为里程碑事件。

3. 网络图

在活动延误的情况下,甘特图调整需要较大的工作量,而网络图将任务计划和进度安排分开,能够清晰地展示现在和将来完成的开发活动及活动间的关系,也可以预先确定各任务的时差。根据工作分解结构(WBS),通过判断项目活动在执行过程中的逻辑关系和先后顺

序,绘制网络图。网络图的主要作用有:①直观清晰地展示项目活动的关键活动、关键路径和工期,找到最有效地完成项目任务的途径和制订出符合实际的、切实可行的进度计划;②项目开发人员对于所负责的开发任务在整个项目成功实现中的关键作用;③有助于确保软件组织资源得到有效的利用;④能够在项目开发前发现进度计划本身存在的缺陷,便于在项目开发过程中根据实际进度对项目的进度计划进行调整和优化。

常用的网络图有前导网络图(Precedence Diagramming Method,PDM)和箭线网络图(Arrow Diagramming Method,ADM)两种。

1) PDM 网络图

PDM 又称单代号网络图,网络节点的矩形框或圆圈表示一个项目活动或任务,箭线表示各项活动(任务)之间的依赖关系,如图 5-5 所示。

2) ADM 网络图

ADM 网络图也称双代号项目网络图(Activity-on-Arrow, AOA),用箭线表示某项活动(任务),用节点将活动连接起来,节点表示前一项活动的结束和下一项活动的开始。节点通常用圆圈表示,两个节点之间唯一确定一个任务,如图 5-6 所示。

在 ADM 网络图中,两个节点确定的活动具有唯一性,如果两项活动是同时展开的,两个代号之间的活动怎样区分?为了更好地表示活动间的逻辑关系,需要设置一个虚活动,虚活动是工期为 0,没有资源消耗的活动,通常用虚箭线表示,如图 5-7 所示。通过增加节点 C 解决了活动“硬件选型”和活动“数据库设计”同时开始和同时结束的问题,活动 CB 即为虚活动,工期为 0。

在一个软件项目中首先完成“硬件选型”活动,才能开始“硬件采购”活动和“软件编码”活动,“软件编码”活动开始前,需要完成“数据库设计”活动,如图 5-8(a)所示;但“硬件采购”活动不需要“数据库设计”活动结束,因此,引入节点 F,形成虚活动 CF 有效地解决 ADM 的逻辑错误问题,如图 5-8(b)所示。

网络图绘制要求包括:①按照活动的逻辑关系绘制;②不准出现循环回路;③不准出现双向箭头和无箭头的连线,从开始一致性地指向结束;④不要出现没有箭尾节点和没有箭头节点的箭线;⑤不要在箭线上引入或引出箭线;⑥尽量避免箭线交叉(采用过桥法或指向处理法);⑦只有一个起点节点和终点节点。

4. 资源图

资源图主要用来表示 IT 项目开发过程中的资源(人力资源、设备、资金和材料等)的分配及利用情况。图 5-9 表示某项目的人力资源分布情况。

资源分解结构(Resource Breakdown Structure,RBS)是根据资源平衡以及编制资源约束型进度计划所需的资源类型,对资源进行分类的层级结构。资源类别包括人力、设备和材料;资源类型包括技能水平、等级水平或适用于项目的其他类型。资源分解结构有助于结合资源使用情况,组织与报告项目的进度数据,详见第 8 章。

常用项目进度管理工具对比见表 5-2。

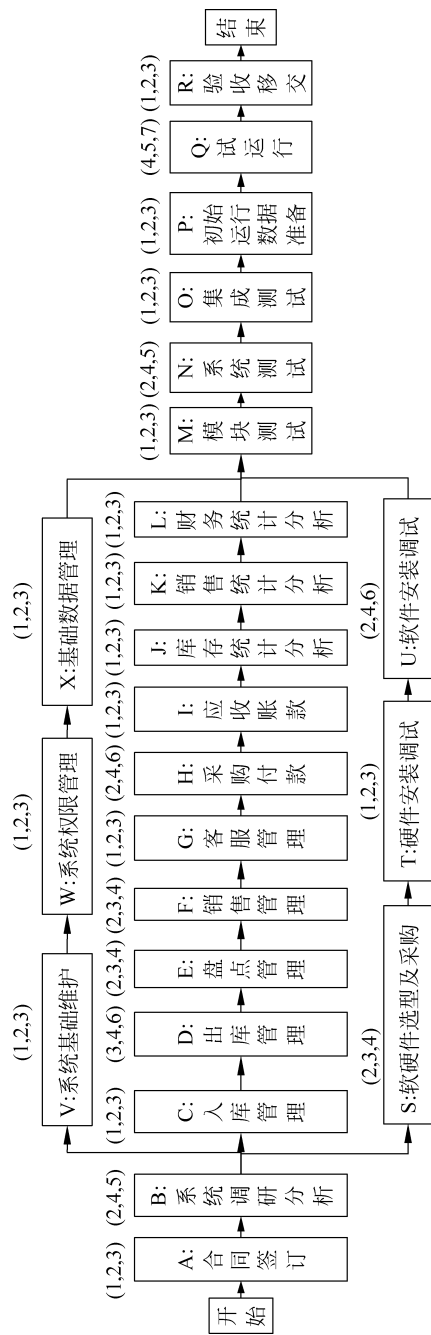


图 5-5 进销存系统开发项目 PDM 网络图

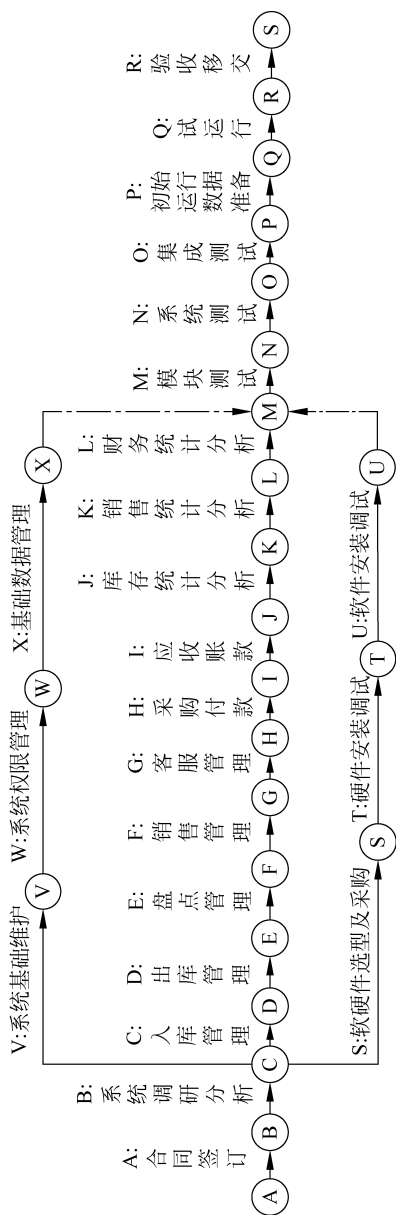


图 5-6 进销存系统开发项目 ADM 网络图

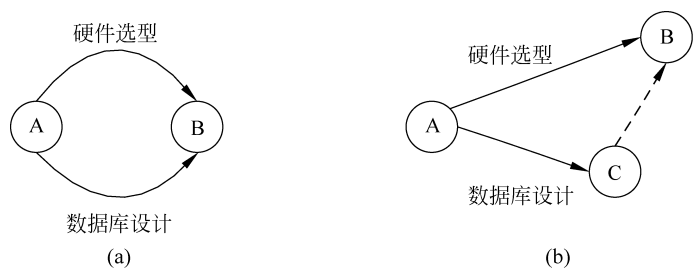


图 5-7 引入虚活动的 ADM 网络图

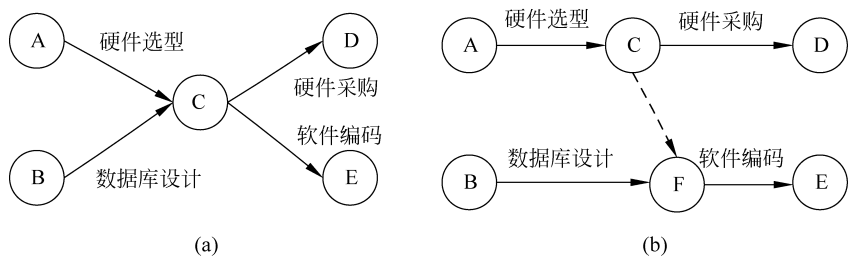


图 5-8 引入虚活动 ADM 网络图

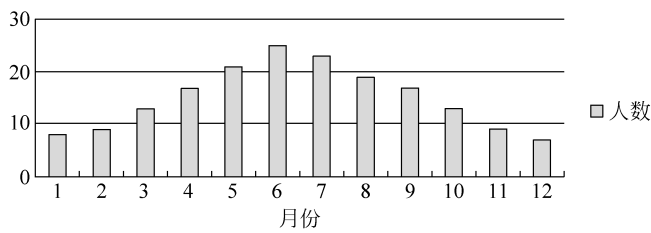


图 5-9 人力资源图

表 5-2 常用项目进度管理工具对比

进度管理工具	作用	优势	备注
甘特图	便于管理层掌握项目进度,用于追踪项目活动的进展	追踪概况性活动进度	概况性活动也称为汇总活动
里程碑图	项目经理、管理层及用户控制阶段性活动进展	概述项目进展情况	标志一系列活动完成,无工期
网络图	项目团队使用,反映了详细的进度计划	定义了项目活动间的逻辑关系	分为单代号图、双代号图
资源图	人力资源部门使用,反映项目生命周期内的人力资源需求	便于组织有效地调整项目间的人力资源分布	反映了资源的时间进度特征
资源分解结构图	可用于识别和分析项目人力资源配备	资源分解结构有助于结合资源使用情况,组织和报告项目的进度数据	资源分解结构是项目成本预算的基础

5.5 活动资源估算

5.5.1 项目活动资源

在 IT 项目开发过程中,项目活动所需的资源类型、数量是影响项目进度的主要因素。因此,在计算项目活动的历时之前,需要对活动所需资源进行估计。IT 项目活动资源主要包括:

(1) 人力资源主要指软件项目的开发人员(分析、设计、实施等)、管理人员(项目经理、QA 等),以及与项目相关的用户和外聘人员等。

(2) 设备包括服务器、计算机和打印设备等。

(3) 资金主要用于购买设备、差旅费以及人员的工资等。

(4) 服务主要指用于完成项目的必要支持服务,如通讯、会议等。

(5) 场地指完成项目的研发和办公场所。

(6) 材料是软件项目开发过程中消耗的资源,如优盘、打印纸等。

(7) 软件项目的其他资源需求可能包括一些支持活动(如配置管理系统、质量保证系统、文档、用户培训等)的资源。软件测试设备和多目标环境或平台部署是可能需要的其他资源。

为了准确估算活动所需资源,项目负责人要明确活动清单、活动属性、资源的可用情况等。项目负责人根据类似项目历史数据以及专家经验进行资源类型和数量估算。

5.5.2 项目活动资源估算方法

活动资源估算就是确定完成项目活动所需要的资源类型、数量和资源使用时间。软件项目经理在资源估算过程中需要考虑因素包括:①已经完成类似项目活动资源消耗情况;②当前项目活动的复杂度情况;③组织资源是否满足项目活动要求的程度;④项目资源可替代性。

软件组织的主要资源是人力资源,软件项目资源估算主要是对人力资源的估算。在估算所需的软件开发人员的数量时,开发人员的技术水平和学习能力是重要参考因素。

软件项目资源估算的主要方法:

(1) 头脑风暴法

通过相关专家对软件项目活动以及组织已完成类似项目分析,专家估算项目活动所需资源的数量和类型。

(2) 自下而上估算

根据 WBS 分解结果,首先估算处于 WBS 底层的活动所需资源,然后自下而上逐层汇总项目活动资源,得到项目整体资源的需求情况。在估算中,还要对活动之间影响资源利用的依赖关系进行说明。

(3) 备选方案分析

部分软件项目活动对资源的需求可以有不同的方案,如一项软件编程活动可以由高水平的程序员在较短工期内完成,由中等水平的程序员完成需要的时间要长一些,因此,在项目资源估算时,要提供多个可选方案,供决策者选择使用。

5.5.3 项目活动资源平衡

在项目开发过程中,当项目活动需要的资源超过组织现有可利用资源时;或者需要的资源在数量、质量(能力水平)方面发生变化时;或者同一资源同一时间被安排不同活动,引发资源冲突时;可能会影响项目进度、成本、质量或范围等目标,导致项目目标无法实现。实际开发工作中通常用资源平衡的方法来解决这个问题。

1. 资源平衡的作用

(1) 根据进度计划合理安排项目所需的人力和物力等资源,避免资源不够、资源冗余或资源不均衡等现象的发生,提高资源的使用效率。

(2) 充分利用非关键路径上的浮动时间来灵活安排项目资源,优化和合理调整项目进度计划,确保进度计划的有效实现,提高项目管理的效率和效益。

(3) 在进度延期的情况下,调整组织对于各个项目资源的投入水平,控制项目进度。

(4) 通过资源平衡,使得项目的人力、物力等资源得到合理、充分、高效的使用。

2. 资源平衡的方法

资源平衡的方法有两种:①灵活使用非关键路径上活动的浮动时间,平衡使用组织现有资源;②调整资源的不同配置(指软件开发人员的能力水平),来避免资源分配不合理或资源的过度分配。

5.6 活动工期估算

活动工期估算是根据活动资源估算的结果、活动间关系,估算完成项目活动所需的时间。根据项目中完成各项活动进行时间估算,确定项目总工期估算。项目活动工期的估算应力求准确,工期估算偏长,则影响项目总的历时,浪费组织资源;工期估算偏短,影响项目完成质量。因此,对项目活动工期估算时,不应受到活动重要性及预定项目完成期限的影响,要在充分考虑项目资源的情况下,把项目活动置于独立的正常状态下进行估算,保证估算过程的客观公正。

5.6.1 活动工期估算的基础数据

软件项目活动属性信息的准确性、有效性、及时性是有效进度管理的前提,进行项目活动工期估算,需要的基础信息如下。

(1) 软件项目开发人员和技能水平。活动的工作量是确定的,资源能力决定了可分配项目活动资源数量的多少,如由全职人员实施某项活动,熟练人员通常能比不熟练人员在更短时间内完成该活动。同时,资源的数量在某种程度上决定了工期的长短,项目人员技术水平不同,完成项目的生产率不同。例如,某项目开发人员和每天完成代码行数,用代码行数/天表示,但是参与项目活动人员数量并不是越多越好。不同技能水平的开发人员消耗的项目资源以及导致项目的成本也有所不同。

(2) 资源日历。项目开发人员和实际可用的工作时间,除去节假日、休假、出差等时间。

(3) 项目人员工作时间。包括连续工作时间、有效的工作时间等。连续工作时间指项目持续、不间断的工作时间。有效的工作时间指正常的工作时间,除去休息、开会等时间,一

般采用每周工作 5 天,每天 8 小时工作制,但是并不是 8 个小时全部花费在项目活动中。

(4) 已完成项目情况。组织已经完成的类似项目数据,可作为当前项目工期估算的参考,主要包括已完成项目工期估计数据库、软件项目特征的信息。

(5) 活动工期表示。可以用具体的数值表示,也可以用某一数值范围。如图 5-5 中的“出库管理模块”编写程序代码活动工期估算为 2 周,或者 10 ± 2 天,表示该活动至少 8 天,但不超过 12 天。

(6) 项目活动实现约束和假设条件。约束条件是指项目活动工期估算所面临的限制因素;假设条件是指项目工期估算所假定的各种存在风险以及可能发生的情况。

5.6.2 活动工期估算方法

活动工期估算主要包括:专家判断法、类比估算法、计划评审技术和参数估算法等。

1. 专家判断法

根据以往类似项目的经验和专家的背景,估算项目活动工期,或根据以往类似项目的信息,给出活动工期的上限。专家也可决定是否使用多种估算方法,以及如何调整各种估算方法之间的误差,最终得到项目活动工期的准确估算。

2. 类比估算法

在软件项目早期,对项目特征数据知之甚少,通常采用类比估算法,来估算当前活动或项目的工期。类比估算是应用已完成相似活动或项目的工期数据,估算当前活动或项目工期的技术,是一种粗略的估算方法,有时需要根据项目复杂性方面的差异进行调整。类比估算通常成本较低、耗时较少、准确性低,可以针对整个项目或项目中的某些活动,进行类比估算。类比估算可以与其他估算方法联合使用。

3. 计划评审技术

考虑到项目活动工期估算的不确定性和风险,以及提高活动持续时间估算的准确性,20 世纪 50 年代美国海军部提出了计划评审技术(Program Evaluation and Review Technique, PERT),认为项目活动的持续时间和整个项目工期是随机的,并服从某种概率分布。通常 PERT 使用三点估算值来判断活动工期的近似区间。

(1) 最乐观时间(Optimistic Time):表示项目开发过程顺利的情况下,完成某项工作的最少时间。

(2) 最可能时间(Most Likely Time):表示完成项目活动所需的最可能估算值,是基于项目的最佳期望估算结果。

(3) 最悲观时间(Pessimistic Time):表示在不利的情况下,完成项目活动所需的最大估算值。

如果三点值估计服从 β 分布(PERT 方法),则项目活动工期如下。

期望值: $E = (O + 4M + P) / 6$

方差: $\delta^2 = [(P - O) / 6]^2$

标准差: $\delta = (P - O) / 6$

其中, E 表示活动历时估算,活动历时估算表示完成该活动所需时间的多少; O 表示活动完成的最乐观时间; M 表示活动完成的最可能时间; P 表示活动完成的最悲观时间。

标准差 δ 表示该任务在期望的时间内完成的可能性大小,标准差越小则表明在期望时

间内完成的可能性就越大,标准差越大则表明在期望的时间内完成的可能性就越小。

在网络图中的一条路径有 n 项活动,每项活动的历时估算为 $E_1, E_2, \dots, E_n$,标准差分别为 $\delta_1, \delta_2, \dots, \delta_n$,则该条路径的历时估算 $E = E_1 + E_2 + \dots + E_n$; 方差 $\delta^2 = (\delta_1)^2 + (\delta_2)^2 + \dots + (\delta_n)^2$; 标准差 $\delta = ((\delta_1)^2 + (\delta_2)^2 + \dots + (\delta_n)^2)^{1/2}$ 。

假设 PERT 历时估算 E 、标准差 δ 符合正态概率分布,则该路径在 $E + 1\delta$ 内完成的概率分布为 68.3%,在 $E + 2\delta$ 内完成的概率分布为 95.5%,在 $E + 3\delta$ 内完成的概率分布为 99.7%。

某进销存系统开发项目各活动最乐观时间、最可能时间和最悲观时间,见图 5-5。假设活动历时时间估计服从 β 分布(PERT 方法),则项目活动工期、标准差和方差,见表 5-3。

表 5-3 项目活动历时估计(PERT)

活动	最乐观时间/周	最可能时间/周	最悲观时间/周	PERT 估计值/周	标准差 δ	方差
A	1	2	3	2	0.3	0.09
B	2	4	5	3.8	0.5	0.25
C	1	2	3	2	0.3	0.09
D	3	4	6	4.2	0.5	0.25
E	2	3	4	3	0.3	0.09
F	2	3	4	3	0.3	0.09
G	1	2	3	2	0.3	0.09
H	2	4	6	4	0.7	0.49
I	1	2	3	2	0.3	0.09
J	1	2	3	2	0.3	0.09
K	1	2	3	2	0.3	0.09
L	1	2	3	2	0.3	0.09
M	1	2	3	2	0.3	0.09
N	2	4	5	3.8	0.5	0.25
O	1	2	3	2	0.3	0.09
P	1	2	3	2	0.3	0.09
Q	4	5	7	5.2	0.5	0.25
R	1	2	3	2	0.3	0.09
S	2	3	4	3	0.3	0.09
T	1	2	3	2	0.3	0.09
U	2	4	6	4	0.7	0.49
V	1	2	3	2	0.3	0.09
W	1	2	3	2	0.3	0.09
X	1	2	3	2	0.3	0.09

该项目的 3 条活动路径总的历时估计期望值、标准差和方差见表 5-4。

表 5-4 项目活动路径历时估计

项目活动路径	PERT 估计值 (周)	方差	标准 差 δ	$\pm\delta$ 内完 成概率		$\pm 2\delta$ 内完 成概率		$\pm 3\delta$ 内完 成概率	
				从	到	从	到	从	到
A-B-V-W-X-M-N-O-P-Q-R	28.8	1.47	1.21	27.59	30.01	26.38	31.22	25.17	32.43
A-B-C-D-F-E-F-G-H-I-J-K-L-M-N-O-P-Q-R	49	2.66	1.63	47.37	50.63	45.74	52.26	44.11	53.89
A-B-S-T-U-M-N-O-P-Q-R	31.8	1.87	1.37	30.43	33.17	29.06	34.54	27.69	35.91

(I-J-K-F-G)路径活动历时估计,见图 5-10。

该路径历时期望值为 23.5 周,在 24.89 周完成的概率 $P=50\%+(68.3)/2=84.2\%$ 。

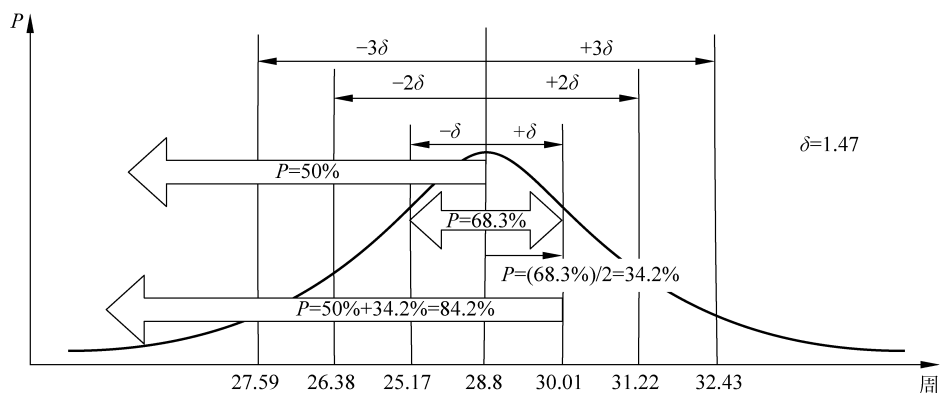


图 5-10 (A-B-V-W-X-M-N-O-P-Q-R)路径活动历时估计

4. 参数估算法

参数估算法是根据历史数据和项目参数,应用某种算法来计算项目活动历时的估算方法。例如,把项目活动开发的工作量(如软件开发中的代码行数、功能点和对象点等)乘以完成单位工作量所需的工时,计算出活动历时时间;图 5-5 中的客服管理模块 H 开发工作量 $Q=15\text{KLOC}$,项目小组的开发效率为 $R=0.8$ 周/千代码行,活动历时时间 $T=Q \times R=15 \times 0.8=12$ 周。

5.7 进度计划编制

进度计划编制是为实施软件开发和管理项目进展,制订合理的计划,在分析活动顺序、历时、最早和最晚时间、工作日历、资源需求和制约因素基础上,对完成的工作进行估计,明确活动的开始和结束时间、项目开始和结束日期、约束条件和目标。

5.7.1 软件项目进度计划方法

编制软件项目进度计划的主要方法有:关键路径法(Critical Path Method, CPM)、关键链项目管理(Critical Chain Project Management, CCPM)、计划评审技术(Program Evaluation and Review Technique, PERT)、时间压缩法(Time Crash Method, TCM)、资源

平衡法(Resource Leveling Method,RLM)等。

1. 关键路径法

关键路径法是一种网络图分析技术,根据项目活动的特定依赖关系,确定各项活动的最早、最迟开始时间以及最早、最迟完成时间,通过计算网络图中历时最长的路径,进而用来预测项目的总工期。

1) CPM 主要时间参数

(1) 最早开始时间(Early Start,ES)是活动开始执行的最早时间,由该活动所有紧前活动中最后一个最早结束时间确定。

(2) 最早完成时间(Early Finish,EF)是活动完成的最早时间,由该活动的最早开始时间加上其工期确定。

(3) 最迟开始时间(Late Start,LS)表示在不影响整个项目按期完成和有关时限约束的条件下,活动最迟必须开始执行的时间,等于活动的最迟结束时间减去活动的工期。

(4) 最迟完成时间(Late Finish,LF)表示在不影响整个项目按期完成和有关时限约束的条件下,活动最迟必须完成的时间。

(5) 活动历时(Duration,D)表示活动从开始到完成的时间。

(6) 总浮动时间(Total Float,TF)是在不影响整个项目按期完成和有关时限约束的条件下,活动可以推迟的总时间量(最早开始时间可以推迟的量)。

(7) 自由浮动时间(Free Float,FF)是在不延误紧后活动最早开始日期的条件下,当前活动可以推迟的时间量(最早结束时间可以延后的量)。

(8) 提前量(Lead)是在不影响整个项目按期完成和有关时限约束的条件下,相对于紧前活动,紧后活动可以提前的时间量。

(9) 滞后量(Lag)是在不影响整个项目按期完成和有关时限约束的条件下,相对于紧前活动,紧后活动可能推迟的时间量,Lag 表示网络图中紧后活动的等待时间。

在网络图中,活动一般用比较直观的“七格图”表示,活动与工期相关的属性填写在七个矩形块中,如图 5-11(a)所示。

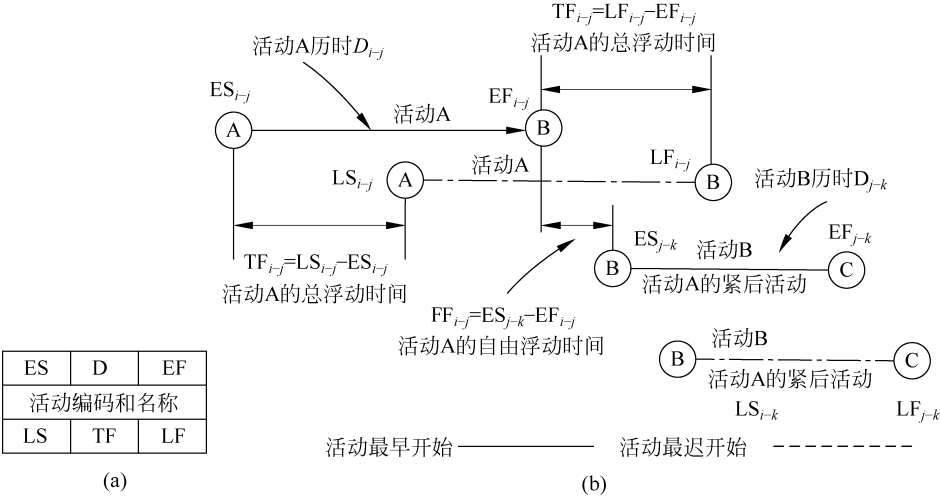


图 5-11 CPM 活动描述及主要时间参数之间的关系

图 5-11(b)表达了 CPM 主要时间参数之间的关系,B 活动是 A 活动的紧后活动,活动 A 开始时间为 i ,完成时间为 j ,活动 B 开始时间为 j ,完成时间为 k 。 ES_{i-j} 表示活动 A 的最早开始时间, EF_{i-j} 表示工作 A 的最早结束时间。 $D_{i-j} = EF_{i-j} - ES_{i-j}$ 表示活动 A 历时时间, $TF_{i-j} = LF_{i-j} - EF_{i-j} = LS_{i-j} - ES_{i-j}$ 表示活动 A 最晚开始时间与最早开始时间之差或者活动 A 最晚结束时间与最早结束时间之差,即活动 A 的总浮动时间。 $FF_{i-j} = ES_{j-k} - EF_{i-j}$ 表示活动 B 的最早开始时间与活动 A 的最早结束时间之差,即活动 A 的自由浮动时间。

总浮动时间是在整个路径上有效,不会影响到整个项目的完成时间,而自由浮动是不影响到紧后活动的最早开始时间。总浮动的值是大于等于自由浮动的值。

活动浮动时间可以为正值、0 或负值。活动浮动时间为 0 的活动是项目的关键活动,表示该活动一旦延期则影响整个项目的完成时间。由关键活动构成的网络路径就是项目的关键路径。关键路径上的活动可能出现负的浮动时间,一般是顺推法所得的完工时间晚于用户要求的完工时间。

关键路径法首先应用正推法(Forward Pass)正向计算活动的最早开始时间(ES)和最早完成时间(EF),然后应用逆推法(Backward Pass)反向计算活动的最迟完成时间(LF)和最迟开始时间(LS)。

2) 正推法

正推法是指按照网络图中活动发生的先后顺序即网络图中的箭头顺序,自左向右计算各项活动最早开始和最早完成时间的方法。具体步骤如下。

(1) 根据项目活动的逻辑关系,绘制项目的网络图。

(2) 从网络图的“开始”活动开始,按照从左到右、从上到下计算每项活动的最早开始和最早完成时间。“开始”活动为虚拟活动,开始时间为 0,工期为 0,总浮动时间和自由浮动时间均为 0。设置第一个活动最早开始时间为 1。

(3) 活动最早完成时间等于最早开始时间加上活动历时时间;紧后活动的最早开始时间取决于紧前活动的最早完成时间。当活动有多个紧前活动时,该活动的最早开始时间取决于前置活动中最早完成时间最晚的那项活动。因为当前活动有多项紧前活动意味着所有活动结束后该项活动才能开始,所以紧后活动的最早开始时间取决于紧前活动最早完成时间最大的活动,即“正向算流入,正向取最大”。

根据进销存系统开发项目网络图 5-5 以及表 5-2 项目活动历时估计(PERT)值(采取四舍五入),用正推法得到活动的最早开始时间和最早完成时间,见图 5-12。

3) 逆推法

逆推法是指按照网络图中活动发生的逆顺序即网络图中的箭头反向顺序,自右向左、自上而下计算各项活动最晚完成和最晚开始时间的方法。具体步骤如下。

(1) 从网络图右边开始,计算每个活动最晚开始和最晚结束时间。

(2) 在项目中所有活动结束后,设置一个虚拟活动“完成”活动,该活动与“开始”活动性质相同,工期为 0,总时差和自由时差为 0。或者设置最后活动的最迟时间,令其等于正推法计算出的最早完成时间。

(3) “完成”活动的完成时间为项目完成时间。活动最晚开始时间为最晚完成时间减去活动工期;紧前活动最晚完成时间根据其紧后活动的最晚开始时间确定。

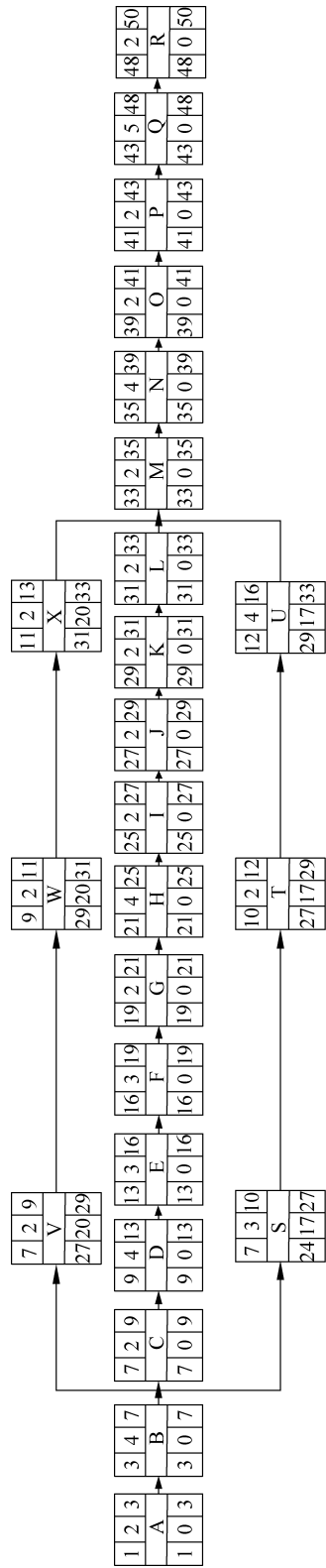


图 5-12 进销存系统开发项目关键路径图

(4) 活动有多个紧后活动时,该活动的最晚完成时间根据紧后活动中最晚开始时间的最小值确定,以紧后活动最晚开始时间中数值最小的时间作为该项活动的最晚完成时间,即“反向算流出,反向取最小”。

进销存系统开发项目用逆推法得到活动的最晚完成时间和最晚开始时间,见图 5-12。

4) 确定关键路径的方法

(1) 最长路径法。分别计算网络图中每条路径的时间长度,以时间长度最大的为关键路径。

(2) 总浮动时间最小法。网络图中总浮动时间最小的路径为关键路径。总时差为 0 的活动,即为关键活动,由总浮动时间等于零的活动组成的路径即为关键路径。

图 5-12 进销存系统开发项目关键路径为 A-B-C-D-E-F-G-H-I-J-K-L-M-N-O-P-Q-R (下简称关键路径 A→R),项目工期为 49 周。

关键路径上活动的总历时即为项目的总工期,项目经理必须对关键路径上的各项活动特别关注,如果关键路径上有些活动延期,项目经理必须及时采取纠正性措施,否则整个项目就会延期。另外,项目的关键路径并不唯一,针对有多条关键路径的项目,此时项目经理要注意各条关键路径的活动执行情况。对于软件项目,由于变更较多,导致关键路径具有动态性特征,在开发过程中,要时刻关注关键路径变动,以保证项目按期完成。

CPM 从理论上计算项目活动的最早最晚开始和结束时间及项目工期,却忽略了项目开发过程中的资源限制、条件约束等相关因素对进度计划的影响,因此 CPM 计算出的活动历时应充分考虑资源限制以及项目的特殊要求,对活动的历时和项目工期进行调整,才能真实地反映项目的实际运行状况。时间压缩法、资源平衡等方法调整活动历时和项目工期的主要方法。

2. 关键链项目管理法

CPM 是严格按照活动紧前紧后的依赖关系确定项目进度计划。由于软件开发活动的不确定性以及资源的有限性,活动延期经常发生,CPM 测算活动的历时和起止时间也随之变动,因此,需要不断地调整项目活动的起止时间。关键链项目管理法在 CPM 的路径上设置缓冲时间,以应对项目资源限制和不确定性。

为了保证开发活动能够有较高的概率在计划时间内完成,一般活动计划时间都大于完成任务所需的平均时间,即在活动所需的时间中增加一部分“安全时间”(Safety Time,ST)。ST 的正面效果是能够更好地应对管理不确定因素,负面效果是延长了完成项目工期。关键链方法则把分配给各个活动的安全时间进行整合,在各个活动路径设置缓冲时间,能够有效地缩短项目工期。

关键链法包括三种类型的缓冲:项目缓冲(Project Buffer,PB)、接驳缓冲(Feeding Buffer,FB)和资源缓冲(Resource Buffer,RB)。项目缓冲是放置在关键链末端的缓冲,用来保证项目不因关键链的延误而延误;接驳缓冲是设置在非关键链与关键链的接合点,用来保证关键链不受非关键链延误的影响;资源缓冲是当需要投入某种资源来启动关键链上的工序,而其前续关键工序又使用其他资源时,需要在该工序之前设置资源缓冲,是为了保证关键链上资源的预警机制,一般没有规模大小的规定,其作用在于确保资源供应。在关键链上由于工序 F 和 G 使用的资源不同,应在工序 G 之前设 RB,如图 5-13 和图 5-14 所示。

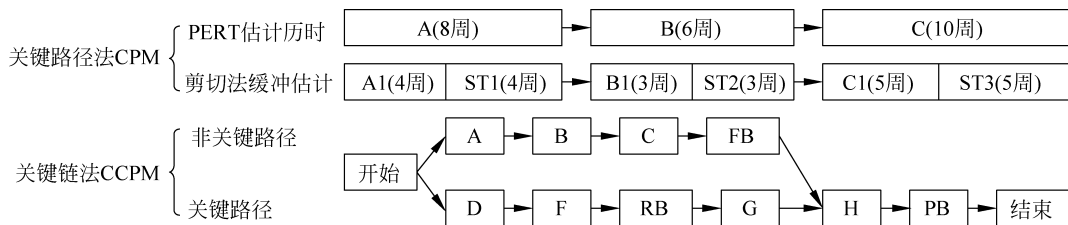


图 5-13 CPM 和 CCPM 缓冲时间设置对比

1) 关键路径法的活动历时估计存在的问题

(1) 活动历时估计过高。

由于考虑软件开发活动的不确定性,活动的完成概率可以认为是 β 分布或三角分布,开发人员往往做出最悲观的时间估计,人为地增加了更多的额外时间。

(2) 项目经理的安全边界。

由于项目经理往往在开发人员估计的活动完成时间基础上,也考虑活动的不确定性,因此,重复增加了活动的安全完成时间。

(3) 预计高层管理者减少项目开发工期。

预计高层管理者为了压缩成本,而减少项目开发工期,项目管理团队常常事前采取多增加一些工期的方式予以应对。因此,增加了项目开发的整个工期。

2) 项目延期的原因

(1) 帕金森定律。开发人员尽可能用尽活动的历时,即使提早完工也不报告,导致各个活动节省的时间,无法使其他活动和整个项目受益。程序员可以一周完成一项软件功能,也可能2~3周完成该功能,如果活动计划时间充裕,就会放慢开发节奏,直至活动历时消耗殆尽。

(2) 资源冲突。当有限的资源用到多项开发活动时,某些任务无法获得相应资源,导致项目延期。

(3) 缺乏有效约束与激励机制。

3) 项目缓冲时间大小的设定

墨菲定律(Murphy's Law)认为任何可能变坏的事都会变坏,即事情变糟的可能性总是比变好的可能性要大,以及开发人员防患于未然的心理行为,项目开发活动的历时呈现 β 分布,如图5-15所示。其中,横轴表示活动历时,纵轴表示对应历时的完成概率。在确定的工期下,概率曲线和横轴之间围成的面积表示活动在该历时下可完成的保证率。

高德拉特(Goldrat)认为开发人员往往因考虑到个人声誉、切身利益等原因,趋向于选取具有高保证率(如95%)的活动历时估计,而这要比低保证率(如50%)下的活动历时估计大得多,如图5-15所示。

常用的计算缓冲大小的方法有两种:剪切法和根方差法。

(1) 剪切法(Cut and Paste Method, CP)。

剪切法是1997年高德拉特(Eliyahu M. Goldratt)提出的,是通过剔除活动历时估计中隐含的安全时间来计算项目缓冲时间大小的方法。Goldratt认为活动历时包括活动计划历

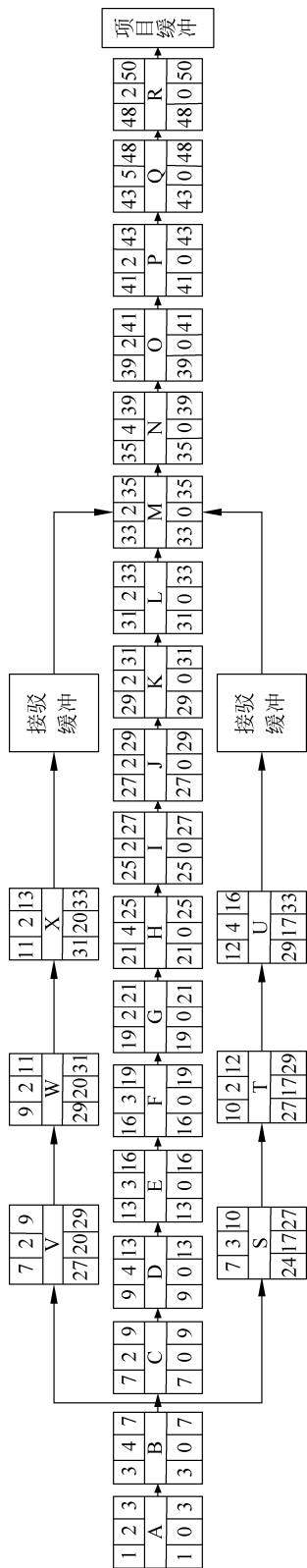


图 5-14 进销存系统开发项目缓冲设置示意图

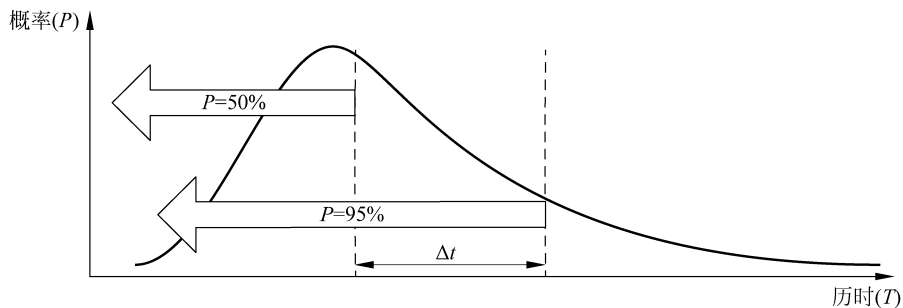


图 5-15 活动历时概率分布

时和安全缓冲(Δt_i)时间两部分;活动计划历时应为活动估计历时估计值(如 PERT)的 50%,安全缓冲(Δt_i)时间也设置为活动历时估计值(如 PERT)的 50%,因此,剪切法也称为 50%法;链路上活动的安全缓冲 Δt_i 总和的 50%作为活动缓冲时间的大小。将关键链长度的 1/2 作为项目缓冲的大小,将非关键链到关键链入口处的长度的 1/2 作为输入缓冲的大小,如图 5-13 所示。

$$\Delta B = \frac{1}{2} \sum_{i=1}^n \Delta t_i$$

式中, ΔB 为缓冲, Δt_i 为活动的安全缓冲, n 为路径上的活动数。

CP 法先用 PERT 或由项目经理和团队成员估计各开发活动的历时,然后用 50%的概率在估算中除去安全时间,剩余就是活动的限定历时;最后取关键链上所有活动被除去的安全时间总和的 1/2 作为 PB。将非关键链上活动被除去的安全时间总和的 1/2 作为非关键链的 FB。

应用剪切法对图 5-5 中项目工期分析如下。

① 关键路径 A→R 所有活动的时间减半后,路径 A 的安全工期为 24.5 周,路径缓冲时间为 12.25 周,则路径的总工期为 36.75 周。

② 非关键路径 V→X 所有活动的时间减半后,路径的安全工期为 3 周,路径缓冲时间为 1.5 周,则路径的总工期为 4.5 周。

③ 非关键路径 S→U 所有活动的时间减半后,路径的安全工期为 4.5 周,路径缓冲时间为 2.25 周,则路径的总工期为 6.75 周。

按照剪切法计算的项目工期为 36.75 周,比关键路径法估计的项目工期 49 周,减少 12.25 周。

CP 简单易行,但安全时间 Δt_i 主要是根据项目管理人员经验确定,准确度低,误差比较大,缺乏理论依据,缓冲时间大小随着相关链路工序数的变化而变化。链路上活动增多时,容易使缓冲时间过大,活动较少时,容易使缓冲时间过小。

(2) 根方差法(Root Variance Method, RSM)。

1998 年,Newbold 提出了根方差法,认为活动的安全缓冲 Δt 代表了活动历时的不确定性,并建议用 $\Delta t_i/2$ 作为活动历时的标准差,并以 2 倍的链路标准差作为链路的缓冲估计。假定链路上活动历时相互独立,则根据中心极限定理可得:

$$\Delta B = 2 \left[\sum_{i=1}^n \left(\frac{\Delta t_i}{2} \right)^2 \right]^{1/2} = D \left[\sum_{i=1}^n (\Delta t_i)^2 \right]^{1/2}$$

式中, ΔB 为缓冲, Δt_i 为活动的安全缓冲, n 为路径上的活动数。

PERT、剪切法和根方差法活动历时及工期估计见表 5-5。

表 5-5 PERT、剪切法和根方差法活动历时及工期估计

路径或活动	PERT 估计历时/周	安全缓冲时间/周		路径周期/周	
		剪切法	根方差法 (标准差)	剪切法	根方差法
A	2	1	0.5	0.5	0.25
B	3.8	1.9	0.95	0.95	0.9
C	2	1	0.5	0.5	0.25
D	4.2	2.1	1.05	1.05	1.1
E	3	1.5	0.75	0.75	0.56
F	3	1.5	0.75	0.75	0.56
G	2	1	0.5	0.5	0.25
H	4	2	1	1	1
I	2	1	0.5	0.5	0.25
J	2	1	0.5	0.5	0.25
K	2	1	0.5	0.5	0.25
L	2	1	0.5	0.5	0.25
M	2	1	0.5	0.5	0.25
N	3.8	1.9	0.95	0.95	0.9
O	2	1	0.5	0.5	0.25
P	2	1	0.5	0.5	0.25
Q	5.2	2.6	1.3	1.3	1.69
R	2	1	0.5	0.5	0.25
S	3	1.5	0.75	0.75	0.56
T	2	1	0.5	0.5	0.25
U	4	2	1	1	1
V	2	1	0.5	0.5	0.25
W	2	1	0.5	0.5	0.25
X	2	1	0.5	0.5	0.25
关键路径 A→R	49	24.5	3.1	36.75	30.7
非关键路径 S→U	9	4.5	1.3	6.75	7.1
非关键路径 V→X	6	3	0.9	4.5	4.8

路径上活动数量较多时, RSM 法比 CP 法确定缓冲更为适合。采用 RSM 法的前提是假定路径上活动的时间参数估计是相互独立的。但是实际上路径上活动可能会受各种因素影响而具有相关性, 且变坏的可能性要比变好的可能性大, 造成确定的缓冲偏小。

关键链法的使用过程中的主要注意事项如下。

① 运用关键链法首先要确定项目的约束。关键链法在关键路径的基础上, 更加注重实际调动资源的影响, 充分考虑项目活动间的逻辑关系和资源限制得出的项目最短时间路径。

② 使用关键链法要充分挖掘约束因素的潜质。关键链中的约束是限制项目进度的关键因素之一, 关键链的完成时间即为项目总时间, 通过挖掘约束因素的潜质可以缩短关键链时间, 更好地制订进度计划。

③ 保证关键链上的关键任务不受非关键任务的影响。在项目实施过程中,虽然增加了安全时间,但是仍存在项目延期完成的情况。如前置任务的延迟会导致后续任务的延迟;当项目中某项活动的前置活动较多时,项目进度取决于延迟最久的活动,即使前置活动中存在提前完工的活动,后置活动也不能提前开工。

④ 合理设置 RB 用来避免资源分配不合理造成的进度延误状况。RB 提示项目经理及时监督资源是否到位、分配是否合理等。如果关键任务所需要的资源被前置的非关键任务占用,关键链法要求提前在项目进度计划中标识 RB,同时项目负责人及时调配资源,避免因资源欠缺导致关键任务的延误。

3. 时间压缩法

时间压缩法是在不改变项目范围的情况下(如开发内容),寻找缩短项目活动历时的方法。常用的时间压缩法主要包括应急法和平行作业法。

1) 应急法

应急法又称赶工法或时间-成本平衡方法,是通过分析成本和进度平衡关系,采取一定措施,加快项目进度,达到缩短项目总工期的目的。赶工通常采用适当增加项目开发人员或加班的方式,用最小增量成本以达到最大量的缩短时间。项目进度压缩会引起开发费用的增加,应急法的实质是以成本换时间。如果项目的进度目标优先于成本目标,项目经理就要平衡成本增加和进度缩短之间的关系,确定合理的进度压缩幅度和具体的开发活动环节。

应用应急法须具备的条件如下。

- (1) 项目活动(尤其是关键路径上的活动)存在正常进度以及可压缩进度。
- (2) 项目具备进度压缩的资源。
- (3) 项目活动进度压缩具有一定限度,因此进度必须在可压缩进度范围内。

在可压缩范围内,成本增长与进度压缩成正比的公式为:

单位进度压缩成本 = (压缩后成本 - 正常成本) / (正常进度 - 压缩后进度)

某小型项目包括 A、B、C 和 D 四项活动,网络图如图 5-16 所示。各活动的正常进度、正常成本、压缩后进度和压缩后成本以及单位进度压缩成本,见表 5-6。项目的 A-B-C 路径活动的总工期为 18 周,成本为 78 万元; D-E 路径活动的总工期为 16 周,成本为 57 万元。根据成本增加最小的原则,选择单位进度压缩成本最小的活动进行压缩。

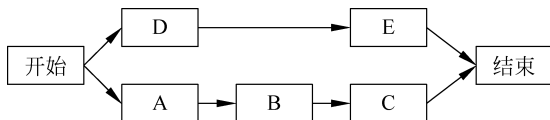


图 5-16 某项目 PDM 网络图

(1) 项目的目标总工期为 17 周, A-B-C 路径需要压缩, A 活动单位压缩成本最低, 则 A 活动历时由 6 周压缩到 5 周, 压缩 1 周, 增加成本 2 万元。

(2) 项目的目标总工期为 16 周, A-B-C 路径需要再压缩, A 活动已经不能再压缩, B 单位压缩成本最低, 则 B 活动历时由 8 周压缩到 7 周, 压缩 1 周, 增加成本 3 万元。

(3) 项目的目标总工期为 15 周, A-B-C 路径需要再压缩, A 活动已经不能再压缩, B 单位压缩成本最低, 则 B 活动历时由 7 周压缩到 6 周, 再压缩 1 周, 增加成本 3 万元; D-E 也需要压缩, D 单位压缩成本最低, 则 D 活动历时由 7 周压缩到 6 周, 压缩 1 周, 增加成本 2 万

元；两个路径工期均达到 15 周。

表 5-6 项目活动进度和成本

任 务	A	B	C	D	E
正常进度/周	6	8	4	7	9
压缩后进度/周	5	5	2	5	8
正常成本/万元	18	24	36	21	36
压缩后成本/万元	20	33	44	25	40
单位进度压缩成本/(万/周)	2	3	4	2	4

该项目活动压缩后的详情,如表 5-7 所示。项目正常成本为 135 万元,正常进度为 18 周,压缩到 15 周后,项目的成本为 145 万元。

表 5-7 压缩后的项目成本

完成周期/周	需要压缩的路径	选择压缩任务	成本计算/万	项目总成本/万
17	A-B-C	A	18+2	135+2
16	A-B-C	B(A 不能再压缩)	18+2+3	135+2+3
15	A-B-C	B(A 不能再压缩)	18+2+3+3	135+2+3+3
	D-E	D	21+2	135+2+3+3+2

2) 平行作业法

项目开发过程中,在资源允许的条件下,根据活动间的逻辑顺序关系,尽可能将顺序发生的活动调整成并行开发,更合理利用资源的空闲时间,缩短项目的总工期。

选取图 5-5 关键路径上的部分活动 C-D-E-F,正常的逻辑关系下,系统功能开发需要 11 周;在资源满足要求和各个活动逻辑关系不冲突情况下,C 活动开始 1 周后,D 活动开始,E 活动在 D 活动开始 2 周后开始,F 活动在 E 活动结束后进行(保持逻辑顺序不变),则路径活动周期为 8 周,比正常逻辑顺序减少 3 周时间,如图 5-17 所示。

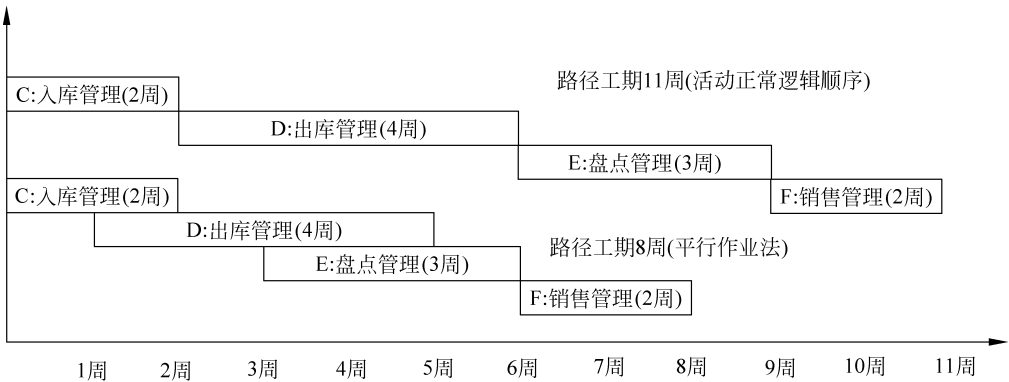


图 5-17 平行作业法

平行作业法通过活动有序衔接,压缩项目进度,如果紧前紧后活动逻辑关系衔接有误,可能造成风险增加,项目后续活动返工的概率也会增大,因此,项目经理以及团队成员必须注意合理安排平行作业法。

4. 资源平衡法

通过合理地调整项目活动的开发顺序和并行作业方式,有效地分配与调度资源,有助于提高资源的使用效率和缩短项目开发工期。

资源平衡法主要通过有效利用项目资源,调整不同阶段项目活动对资源的需求数量,在松弛量允许的范围内移动活动,最小化地降低资源闲置时间,实现资源的合理高效使用。资源平衡重点在于将稀缺资源优先应用到关键路径的关键活动中,形成资源使用的平稳分布,尽可能形成连续的资源链。

资源平衡法通常应用在以下情况。

- (1) 时间安排需要满足规定交工日期的计划活动。
- (2) 只有在某些时间动用或只能动用有限数量的必要的共用或关键资源数量。
- (3) 在项目工作具体时间段内按照某种水平均匀地使用选定资源。

某项目有 R、S 和 T 三项活动,项目活动的网络图及活动的历时,如图 5-18(a)所示。三项活动同时开始需要 12 人,2 周后需要 9 人,有 3 人闲置;4 周后需要 6 人,有 6 人闲置,如图 5-18(b)所示。如果利用 T 活动的浮动时间,采取最晚开始时间,即 T 活动推迟到第 4 周开始,则最多需要 9 人即可完成任务。因此,同样在 6 周的工期内完成项目,仅需要 9 个开发人员,如图 5-18(c)所示,降低了人员闲置,提高了资源利用率。当然资源平衡的前提是 9 个开发人员都具备完成活动 R、S 和 T 的能力。

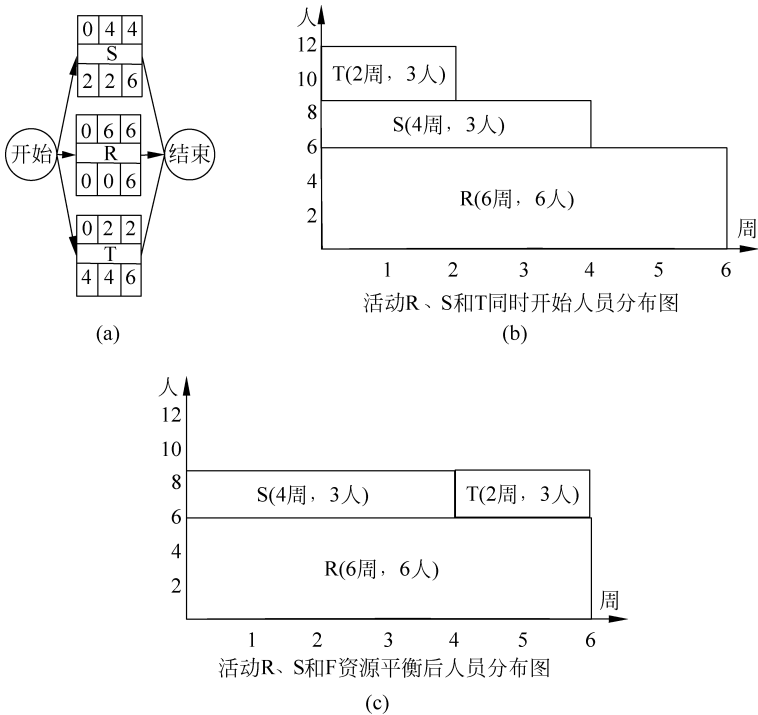


图 5-18 资源平衡示例图

5.7.2 进度计划编制的步骤

软件项目进度计划制订的主要依据是项目范围和 WBS,包括活动界定与描述、活动工

作量估计、活动排序、绘制网络图、资源分配、活动历时估计和进度安排等活动,如图 5-19 所示。

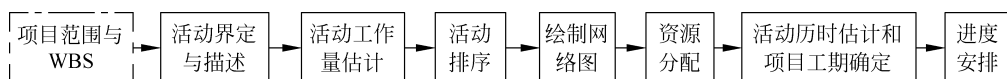


图 5-19 进度计划制订的步骤

1. 活动界定与描述

根据项目范围和 WBS 结果,界定完成工作包所需的各项活动,活动细化到能够由一个小组来完成,明确各项活动的具体工作内容和要求。

2. 活动工作量估计

根据项目具体开发活动的规模和复杂程度,计算各活动的工作量。

3. 活动排序

根据开发活动的内在联系,确定活动间逻辑顺序关系和工作开展的前提条件。对于没有明显逻辑关系的活动,应充分考虑组织的资源情况协调统筹安排,以达到资源的有效利用。

4. 绘制网络图

根据开发活动排序结果,初步建立项目活动网络图,确定项目中各项活动之间的依赖关系,通过网络图进行直观显示,尽可能地避免活动的遗漏,并确定关键路径和关键活动。

5. 资源分配

根据活动的工作内容、复杂度和活动规模,考虑资源的能力、水平和成本等因素,分配相应的人力、设备等资源,平衡分配活动所需资源。为了明确各部门或团队责任,便于项目经理在项目开发过程中的管理协调,应根据工作分解结构和项目组织结构对项目的每一项开发活动分配责任者,即工作责任分配矩阵,确保项目人员在最佳的工期内承担合适的工作量,并根据实际情况及时调整资源的供需情况。

6. 活动历时估计和项目工期确定

根据活动的工作量和分配资源的工作效率,估计各项活动历时。历时估计要求客观正确,如果工作时间估计太长,则影响项目的开发效益;如果工作时间估计太短,则会导致被动紧张局面;反而会延长项目工期。估计活动历时应充分考虑活动的重要性及活动完成期限的限制,以及各种资源供应、工作量、工作效率等因素,综合确定活动的历时。确定项目的浮动时间、关键路径和项目的总工期。

7. 进度安排

进度安排即对软件项目进度的计划编制,在明确项目范围、资源分配以及项目活动历时估计之后,确定活动的开始和结束时间以及项目整体的起止时间,制订项目进度计划。软件项目的活动工作量估计、历时估计、资源分配与进度安排通常是交替进行,相互协调,以实现更为精确的项目进度管理目标。

8. 冲突解决

影响软件项目进度计划的各种因素很可能出现冲突,如活动的提前或滞后、个人活动进度偏差、资源占用、突发事件和任务分配等都有可能产生冲突,通过平衡、调整以及增加资源,解决进度计划中的冲突问题。

5.8 进度控制

进度控制是通过跟踪记录项目开发状况,更新项目进展信息。分析项目活动计划与实际进展情况偏差,通过管理进度基准变更,及时采取纠正和预防措施,以实现进度计划正常进行的过程。

控制进度作为实施整体变更控制过程的一部分,主要完成判断项目进度的当前状态、对引起进度变更因素施加影响、判断进度是否已经发生变更、在变更发生时对其进行管理工作。进度控制的主要方法有绩效审查、项目管理软件、资源优化技术和建模技术。

5.8.1 进度控制原则

1. 系统原则

为实现项目进度目标,进度计划由大到小,活动内容从粗到细,逐步分解。为了保证项目进度,项目组织应设有专门的职能部门或人员系统地负责进度的记录、跟踪、统计、分析及调整等工作。对于大型软件项目,需要多名项目进度管理人员,不同的人员负有不同的进度控制责任,分工协作,形成一个较为复杂的项目进度控制系统。所以无论是控制对象,还是控制主体,无论是进度计划,还是控制活动,都应是一个完整的系统。

2. 动态控制原则

由于活动历时估计的不确定性,导致项目进度计划也存在着不确定性,随着项目开发活动推进,早期的项目活动有的可能提前完成,有的可能延期。因此,按早期制订的进度计划可能不符合后续软件开发阶段的要求,所以需要对当前生命周期阶段的后续生命周期阶段的软件进度计划重新制订。当前生命周期阶段或里程碑事件结束后,要对前一个节点的进度绩效进行分析,根据绩效分析结果,重新制订后续软件活动进度计划,实施继续控制。因此,进度控制应采用这种动态循环的控制方法,如图 5-20 所示。

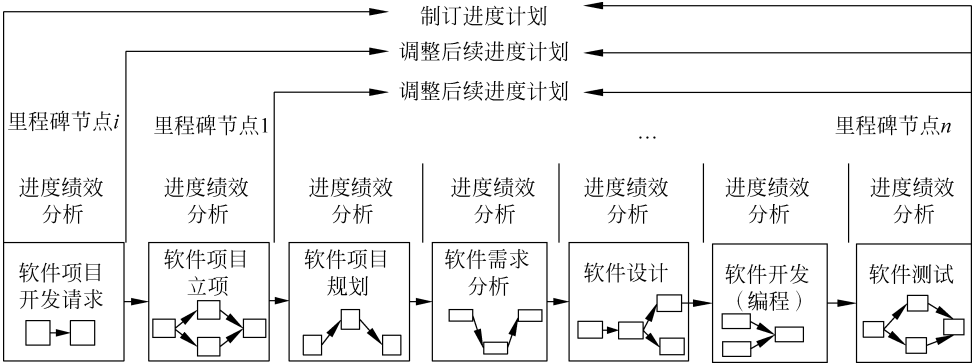


图 5-20 进度计划动态控制示意图

3. 封闭循环原则

项目进度控制的全过程是一种循环性的例行活动,其中包括编制计划、实施计划、检查、比较与分析、确定调整措施和修改计划,从而形成了一个封闭的循环系统。进度控制过程就是这种封闭循环中不断运行的过程,如图 5-21 所示。

4. 弹性原则

软件项目工期长,影响进度因素多,因此,计划编制应考虑各种因素的影响程度和出现的可能性,并在确定进度目标时,分析目标的风险,从而使进度计划留有余地。在控制项目进度时,可以利用活动弹性缩短工作的持续时间,或调整活动之间的链接关系,以使项目最终能实现工期目标。

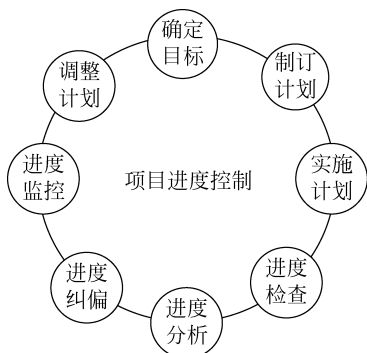


图 5-21 进度计划循环控制

5. 网络计划技术原则

网络计划技术不仅可以用于编制进度计划,而且可以用于计划的优化、管理和控制。网络计划技术是一种科学且有效的进度管理方法,是项目进度控制,特别是复杂项目进度控制的完整计划管理和分析计算的有效工具。

5.8.2 进度控制方法

绩效审查是指度量、对比和分析进度绩效,如活动的实际开始和完成时间、已完成百分比及当前活动的剩余持续时间。绩效审查可以使用各种技术,其中包括以下几项。

1. 趋势分析

趋势分析检查项目绩效随活动推进的变化情况,以确定绩效是否满足需要。图形分析技术有助于理解当前绩效水平,并与未来的目标绩效(如完工日期)进行对比分析。

2. 关键路径法

通过比较关键路径的进展情况来确定整体进度状态。关键路径上的差异将对项目的完成日期产生直接影响。评估关键路径上活动的进展情况,有助于识别进度风险。

对于非关键路径上的某个活动发生较长时间的延误,可能不会对整体项目进度产生影响;而某个关键或次关键活动的稍许延误,却可能需要立即采取行动。

3. 关键链项目管理法

通过对项目缓冲时间和剩余项目缓冲时间、接驳缓冲与剩余接驳缓冲时间分析,确定进度状态。是否需要采取纠正措施,取决于所需缓冲与剩余缓冲之间的差值大小。

4. 挣值管理

采用进度绩效测量指标:进度偏差(SV)和进度绩效指数(SPI),评价偏离初始进度基准的程度。总浮动时间和最早结束时间偏差也是评价项目进度绩效的基本指标。

对于不使用挣值管理的项目,需要开展类似的偏差分析,比较活动的计划开始和结束时间与实际开始和结束时间,确定进度基准和实际项目绩效之间的偏差,根据偏离进度基准的原因和程度,决定是否需要采取纠正或预防措施。

5. 资源优化技术

资源优化技术是在同时考虑资源可用性和项目进度的情况下,对活动和活动所需资源进行调整和优化。

6. 提前量和滞后量

在网络分析中调整活动进度的提前量与滞后量,设法使进度滞后的活动赶上计划。

①通过增加活动之间的提前量,如在学生成绩管理软件开发过程中,把成绩报表统计分析模

块开发,调整到成绩查询模块开发完工之前开始;②通过消除或减少滞后量,如把软件环境调试活动调整到硬件运行环境调试结束完成之后立即开始。

案例分析

张明是一家软件公司的项目经理,最近他正在负责向 A 公司提供一个新的 ERP 系统的项目,该项目要求在 10 月 1 日前完工。张明和他的团队制订了一套详细的进度计划,以便帮助项目尽早完成。

制订进度计划容易,使项目沿着既定的轨道前进则要难得多。管理人的问题及解决进度计划的冲突是两个最大的挑战。不巧的是,该客户的一些员工临时休假,因此错过了一些项目评审会议。由于张明的团队在系统开发生命周期各阶段都需要有来自客户的确认指令,所以上述变更使得张明的团队难以按原来计划的进度进行。此外,张明团队中的一位高级程序员又临时退出了,于是张明只得重找了一个程序员来接替工作。这又打乱了原来的进度计划。

时间转瞬即逝,现在是 9 月 1 日了,离项目规定完工日期只有 1 个月了,项目目前是一片混乱。张明很自负,未向上司说明项目进展得并不顺利,他花了大量的时间来修改项目的进度计划。他认为他的管理方法可以使项目如期完工。此外,张明又将确认指令的责任授权交给了他的一个文员,然而该文员对此过程并不了解。离 10 月 1 日越来越近,系统总算开发完成了,可在最后测试新的 ERP 系统时,张明才发现他们使用的数据并不是最新的,于是又出现了一些额外的问题。最终,项目未能如期完成,张明通过惨痛的教训知道了保持项目按进度进行是多么困难。

问题 1: 系统开发生命周期各阶段,都需要经过客户的确认指令,这属于哪种依赖关系?

问题 2: 从软件项目的角度来分析项目失败的原因,张明应该怎么做才能解决问题?

习 题

一、单选题

- 里程碑事件的持续时间是()。
 - 比最常活动的持续时间短
 - 没有持续时间
 - 等于它表示的活动时间
 - 负数
- 前导网络图(PDM)不同于箭线网络图(ADM),是因为 PDM()。
 - 可以使用 PERT
 - 活动间有四种依赖关系
 - 只有一种依赖关系:开始到开始
 - 可以使用虚拟路径
- 设计必须在建筑开始之前完成,这体现了()。
 - 选择性依赖关系
 - 外部依赖关系
 - 范围依赖关系
 - 强制性依赖关系
- 下列选项中,()不属于控制进度考虑范围。
 - 确定进度计划已经发生

- B. 管理实际发生的变更
- C. 根据客户要求变更进度计划
- D. 对造成进度计划变更的因素施加影响,保证变更是有利的

5. 有一种缩短项目进度的办法是在项目设计阶段的每个活动上安排 3 个开发员而不是 2 个,虽然你考虑这种方法,但设计团队将会扩大一倍。这种做法往往()。

- A. 降低生产率
- B. 提高生产率
- C. 降低了质量审计的必要,因此减少了整个的资源成本
- D. 随着资源增加,生产也按比例增加了

二、判断题

- 1. 规划进度管理过程的主要输出是项目文件更新。()
- 2. 进度计划是时间管理的基础。()
- 3. 活动定义是将任务分解为工作包。()
- 4. 最乐观时间指完成项目活动所需的最小可能估算值。()
- 5. 自由浮动时间是在不延误紧前活动的最早开始日期的条件下,当前活动可以推迟的时间量。()

三、简答题

- 1. 基于生命周期的活动定义方法与基于产品的活动定义方法各有什么优点?
- 2. 什么是选择性依赖关系?
- 3. 前导网络图和箭线网络图有什么异同?
- 4. 进度控制要遵守哪些原则?

四、计算题

某项目完成的最乐观时间是 5 天,最可能时间是 12 天,最悲观时间是 21 天,求这个项目的期望完成时间和方差。