

# 软件项目进度跟踪方法的综合研究

梁成才

(华东计算技术研究所, 上海 200233)

**摘 要:** 进度是软件项目的第一大风险, 受到顾客和承制方的高度关注。该文概述了软件项目的进度管理机制, 对历史上主要的 6 种软件项目进度跟踪方法: 里程碑进度, 人为设定活动进度, 工作单元进展, 挣值法, 甘特图和网络图进行了比较说明, 从计算公式、使用时机、使用性能和发展进化等方面指出了它们各自的特点, 推荐了最佳软件进度跟踪方案。

**关键词:** 软件项目; 进度跟踪方法; 工作分解结构; 计划值(估计值); 实际值

## Integrated Research of Schedule Tracking Methods of Software Project

LIANG Chengcai

(East China Institute of Computer Technology, Shanghai 200233)

**【Abstract】** Schedule is the top risk of software project, hence, it is a common focus on attention of customer and supplier. The schedule management mechanism of software project is described. The main six schedule tracking methods of software project used in the history such as milestone schedule, man-made activity schedule, working unit progress, earned value, gantt chart and network chart are compared and explained. Every method and its characteristics are described by explaining its calculation formulas, application time, application performance, evolution and other aspects. The best solution of schedule tracking of software project is recommended.

**【Key words】** Software project; Schedule tracking method; Work breakdown structure(WBS); Planned value(estimated value); Actual value

### 1 概述

质量、进度和成本是软件项目关注的 3 大要素, 它们相互依赖、相互制约, 软件项目管理就是要在 3 个目标的建立、跟踪和实现方面达成最佳均衡。由于软件的时效性很强, 顾客和承制方都非常关注软件进度, 在软件项目合同中都对进度进行明确规定, 至少规定了交付日期(Deadline)。事实上, 进度是软件项目的第一大风险, 软件项目能否按期保质保量地完成是衡量软件项目成败的标志。在当今业界实施的 CMM (Capability Maturity Model for Software) 或 CMMI (Capability Maturity Model Integration) 中, 软件进度管理是个核心内容。

软件项目的进度管理机制包括 3 个步骤: 计划, 跟踪和控制。计划主要是制定工作分解结构 (Work Breakdown Structure, WBS), 对软件阶段、活动和任务进行规模、工作量等参数的一系列估计, 安排软件阶段、活动和任务的进度, 确定进度跟踪基线。跟踪主要是根据进度的计划值对进度进行动态的监控, 观测进度的状态是否正常, 即实际的进度是否在计划值的容许的偏差阈值范围内。控制主要是针对跟踪发现的进度异常状态, 分析导致进度异常的原因, 采取纠正措施挽回或弥补进度的损失, 在进度调整到正常状态后, 重新回到进度状态跟踪。软件项目的进度管理机制是一个闭环控制系统, 如图 1 所示。

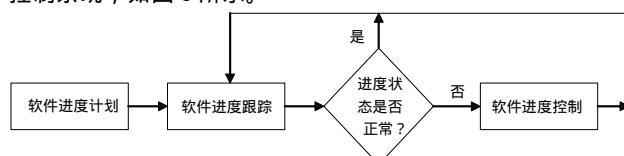


图 1 软件进度管理机制示意图

### 2 主要软件进度跟踪方法说明

#### 2.1 主要软件进度跟踪方法比较

表 1 主要软件进度跟踪方法的比较

方法名称	适用于 WBS 的节点级别	结果表示	使用时机	方法意义
里程碑进度	(1)阶段级 (2)活动级 (3)任务级	(1)偏差百分比 (2)偏差量折线图	(1)项目周期内的阶段节点 (2)阶段内的里程碑、月、周节点	反映对项目交期、阶段节点的进度偏差程度
人为设定活动进度	活动级	(1)进度百分比 (2)计划完成与实际完成比较折线图	阶段内的里程碑、月、周节点	主观假设的进度关系
工作单元进展	任务级	(1)进度百分比 (2)计划完成与实际完成比较折线图	阶段内的里程碑、月、周节点	任务完成的状态
挣值法	任务级	(1)进度和成本偏差百分比 (2)BCWS、BCWP、ACWP 比较折线图	阶段内的里程碑、月、周节点	进度与成本完成的状态
甘特图	(1)阶段级 (2)活动级 (3)任务级	计划完成与实际完成比较横道图	(1)项目周期内的阶段节点 (2)阶段内的里程碑、月、周节点	WBS 及其完成状态的可视化快照
网络图	(1)阶段级 (2)活动级 (3)任务级	WBS 节点的进度依赖逻辑关系、关键路径的网络图	(1)项目周期内的阶段节点 (2)阶段内的里程碑、月、周节点	WBS 节点的进度依赖逻辑关系分解剖面

历史上, 人们使用过多种软件进度跟踪方法, 它们主要

**作者简介:** 梁成才(1965 -), 男, 高工, 主研方向: 软件质量管理和软件测试

**收稿日期:** 2006-06-05

**E-mail:** lcc@ecict.com.cn

包括里程碑进度、人为设定活动进度、工作单元进展、挣值法、甘特图、网络图等，对这些方法有关特征的比较，如表 1 所示。

## 2.2 里程碑进度

里程碑进度方法主要用于对项目总体进度的跟踪，尤其是对项目交付日期的持续跟踪。

该方法度量里程碑的进度延迟（或提前）量，计算公式如下：

$$\text{里程碑进度差异} = \frac{\sum i}{\text{项目周期}}$$

公式中项的解释如下：

当对软件项目整体进度进行考察时，里程碑一般指软件项目生存周期内的阶段节点，项目周期指整个项目的工期；当对软件项目阶段的内部进度进行考察时，里程碑一般指阶段内的活动（或任务）节点，项目周期指该阶段的工期。

$\Delta_i$  是第  $i$  个里程碑的进度延迟量，单位是天，计算如下：

(1) 对每个已经完成的里程碑， $\Delta_i = \text{第 } i \text{ 个里程碑实际完成日期} - \text{第 } i \text{ 个里程碑计划完成日期}$ ；

(2) 对每个已经开始但尚未完成的里程碑， $\Delta_i = \text{第 } i \text{ 个里程碑实际开始日期} - \text{第 } i \text{ 个里程碑计划开始日期}$ 。

项目周期是项目(或阶段)的持续时间(Duration)，单位是天，计算如下：

项目周期 = 项目(或阶段)的计划完成日期 - 项目(或阶段)的计划开始日期

以上计算时， $\Delta_i$  取值可能为正数（对应进度滞后）或负数（对应进度超前），其累计值中正负数发生抵消。实际上，当 WBS 的阶段、活动或任务节点不在一个执行路径上时（如平行执行时），一个节点超前或滞后对另一个节点在进度上没有影响，因此它们的进度差异不能抵消。为此，可以对里程碑进度差异公式进行改进，以  $\Delta_i$  的绝对值  $|\Delta_i|$  替代  $\Delta_i$  代入公式。原里程碑进度差异公式的计算结果掩盖了不在一个执行路径上的节点的无关性，易导致人们盲目乐观。改进后的公式反映了实际进度对项目交期、阶段节点的计划进度的绝对偏差程度，昭示了项目的计划水平或实施能力，具有很高的实用价值。

## 2.3 人为设定活动进度

人为设定活动进度方法主要用于对软件项目阶段的内部进度的跟踪。

该方法对软件项目阶段内的里程碑点赋予进度百分比预算值，以测量和跟踪阶段内部的进度。某项目在概要设计阶段的进度设置如表 2 所示，表中的缩写词是软件设计文档 SDD（Software Design Document）、软件需求规格说明 SRS（Software Requirements Specification）、数据流图 DFD（Data Flow Diagram）。

表 2 体系结构设计和文档编制的进度算法

事件（活动）	完成进度的%
任务开始	10
完成 DFD 和相关算法定义	40
完成软件单元定义和结构关系定义	55
完成 SDD 到 SRS 的追溯	65
内部评审开始	75
内部评审结束	90
完成文档更新并提交基线	100

在早期的软件项目管理中，以上类似的进度设定方法使用比较普遍。该方法的问题是阶段内的活动（或任务）的进度设置完全凭主观假设，缺乏物理意义，并且带有乐观性。

当然，这也是当时缺乏有效的手段对软件进度进行度量的无奈之举。在挣值法出现以后，人们普遍使用挣值法，该方法被渐渐地抛弃。

## 2.4 工作单元进展

工作单元进展方法主要用于对软件项目阶段的内部工程任务（即工作单元）完成状态的跟踪。

该方法依据软件项目阶段内的详细 WBS 的底层任务节点及其进度的计划值，观测任务的完成状态，计算公式如下：

$$\text{任务完成率} = \frac{\text{迄今已完成的工程任务数}}{\text{迄今计划应完成的工作数}}$$

使用工作单元进展方法跟踪软件项目阶段的进展情况，只考虑了任务是否完成的信息，不涉及任务的内部信息，是一种粗粒度的跟踪方法。使用该方法时，应该注意的是阶段内的工程任务的划分需要均衡，任务之间进度、工作量等估计值的差距应不超过一个数量级。只有任务划分遵循“粒度适当、大小可比”原则，使用该方法跟踪进度才比较有效，否则易导致盲目乐观（如当许多小任务都顺利完成而一个大任务没有按期完成时）或盲目悲观，对进度的评判带来不利。

## 2.5 挣值法

挣值法主要用于对软件项目阶段的内部工程任务进度与成本完成状态的跟踪。该方法依据软件项目阶段内的详细 WBS 的底层任务节点的估计值，观测任务进度与成本的完成状态，计算公式如下：

$$\text{SPI} = \frac{\text{BCWP} - \text{BCWS}}{\text{BCWS}} \quad \text{CPI} = \frac{\text{BCWP} - \text{ACWP}}{\text{ACWP}}$$

采用挣值法跟踪项目的进度和成本时，使用下列 3 个变量：

(1) BCWS (Budgeted Cost of Work Scheduled) 已计划工作的预算成本，统计本阶段迄今为止所有计划应完成的任务的工作量估计值；

(2) BCWP (Budgeted Cost of Work Performed) 已完成工作的预算成本，即所谓的“挣值”，统计本阶段迄今为止所有已完成的任务的工作量估计值；

(3) ACWP (Actual Cost of Work Performed) 已完成工作的实际成本，统计本阶段迄今为止所有已完成的任务的工作量实际值。

挣值法利用以上 3 个变量生成一对孪生指标：进度性能指标 (Schedule Performance Index, SPI) 和成本性能指标 (Cost Performance Index, CPI)。

使用 SPI 和 CPI 可以对软件项目在本阶段的进度和成本进行跟踪和分析。2 个指标必须联合判别，判别准则如下：

(1) 2 个都为正，软件项目在进度和成本方面的进展较好，进度超，成本省；

(2) 2 个都为负，软件项目在进度和成本方面的进展较差，进度拖，成本超；

(3) 一正一负，不能说明问题，需进一步分析；

(4) 任何一个为零，说明它代表的属性恰好达到预期目标，属于最佳状态；

(5) 任何一个不为零，需将其与事先设定的跟踪阈值进行比较，超出阈值范围即表示它代表的属性处于异常状态，需进一步分析并采取纠正措施以期重新将其纳入正常轨道。

采用挣值法的前提是将任务假定为原子任务，对任务的进度度量使用二元度量，即任务要么未完成，要么完成。

挣值法依据任务的工作量和进度估计值跟踪软件项目阶段的进度与成本完成状态。在 CMM 等软件项目管理的多年

理论探索和实践确证后，挣值法已经证实是对软件项目的进度、成本进行联合跟踪的有效方法。

### 2.6 甘特图

甘特图主要用于对软件项目的阶段、活动和任务的进度完成状态的跟踪。

该方法依据软件项目 WBS 的各层节点的进度估计值，使用直观的横道图（Bar Chart，也称条形图）显示工作进度计划和工作实际进度状态。甘特图是 WBS 的图示，也是工作完成状态的可视化快照（Snapshot），能够动态地、实时地、直观地比较工作的进展状态。甘特图中横坐标是时间维，纵坐标是 WBS 维，横道图的起点和终点分别表示阶段、活动和任务的开工时间和完工时间，横道图的长度表示工期。用横道图表示工作进度时，一项工作需要两行表示，上面一行是空心横道图，表示工作计划进度；下面一行是实心横道图，表示工作实际进度。甘特图的示例如图 2 所示。

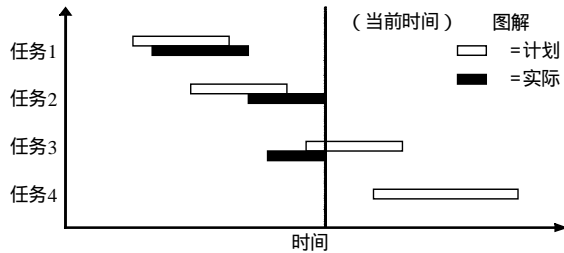


图 2 甘特图示例

甘特图简单易用，一目了然，被广泛地应用于软件项目管理中。微软（Microsoft）公司开发的 Microsoft Project 2003 项目管理软件包括了甘特图功能，是为广大软件项目经理所认同的实用软件项目管理工具。

甘特图的缺点是无法表达任务之间的衔接关系，它把各项任务视为独立的工作，只见树木，不见森林，没有考虑任务之间的逻辑关系，不能反映某一项任务的进度变化对整个项目的影响。

### 2.7 网络图

网络图主要用于对软件项目的阶段、活动和任务的进度实施状态的跟踪和调整。

该方法也称工程评价技术（Program Evaluation and Review Technique, PERT）方法或关键路径法（Critical Path Method, CPM）（实际上，PERT 与 CPM 稍有区别，PERT 计算工期采用的是最乐观值、最大可能值、最悲观值的以 1、4、1 为权重的加权平均值，CPM 计算工期采用的是最大可能值）。它是应用网络图分析技术的项目管理方法，应用于项目进度安排与工期控制。

#### 2.7.1 网络节点的表示形式

网络图方法使用活动节点网络图，网络节点的常用表示形式如图 3 所示。

活动标识		周期	
最早开始	活动名称	最早完成	
最迟开始		最迟完成	
活动跨度		缓冲期	

图 3 网络节点的表示形式

图 3 中，最早开始（Early Start, ES）：一个活动的最早可以开始执行的时间；最迟开始（Late Start, LS）：一个活动的

最迟可以开始执行的时间；最早完成（Early Finish, EF）：一个活动的最早可以完成的时间；最迟完成（Late Finish, LE）：一个活动的最迟可以完成的时间；周期（Duration）：项目工期；活动跨度：最早开始日期与最迟完成日期的差，是活动允许的最大时间；缓冲期：最早开始日期与最迟开始日期的差（或等价地，最早完成日期与最迟完成日期的差），是活动允许的机动时间。

网络图中关键路径是指由网络图上都具有零缓冲期的活动节点连接而成的路径。关键路径具有下列特征：

- (1)网络图上至少存在一条关键路径；
- (2)关键路径是网络图中的最长路径；
- (3)关键路径的工期是完成项目的最短工期；
- (4)关键路径上的活动是关键活动，任何关键活动的延迟都会导致整个项目完成的延迟；
- (5)关键路径是动态变化的，随着项目的进展，非关键路径可能会变成关键路径。

在项目初始策划和进度跟踪过程中，可确定和调整网络图，识别和监控关键路径。实现网络图功能需要采用正向遍历和反向遍历两个技术。

#### 2.7.2 正向遍历

执行正向遍历用于计算每项活动开始和完成的最早日期，步骤如下：

- (1)首先确定项目（或阶段）的开始日期；
- (2)项目（或阶段）的开始日期是网络图中第 1 个活动的最早开始日期；
- (3)从左到右、从上到下进行活动编排；
- (4)当一个活动只有一个前置活动时，将前置活动的最早完成日期作为该活动的最早开始日期；
- (5)当一个活动有多个前置活动时，选择其中最大（即最后）的最早完成日期作为该活动的最早开始日期。

公式： $ES + Duration = EF$

约定：项目（或阶段）的开始日期是 0 或 1。

#### 2.7.3 反向遍历

执行反向遍历用于计算每项活动开始和完成的最迟日期，步骤如下：

- (1)首先确定项目（或阶段）的完成日期；
- (2)项目（或阶段）的完成日期是网络图中最后一个活动的最迟完成日期；
- (3)从右到左、从上到下进行计算；
- (4)当一个活动只有一个后置活动时，将后置活动的最迟开始日期作为该活动的最迟完成日期；
- (5)当一个活动有多个后置活动时，选择其中最小（即最早）的最迟开始日期作为该活动的最迟完成日期。

公式： $LF - Duration = LS$

约定：项目（或阶段）的最迟完成日期与项目（或阶段）的最早完成日期一样，并将其规定为项目（或阶段）的完成日期。

#### 2.7.4 网络图示例

使用以上的网络图技术生成的一个网络图示例（图 4）。图 4 显示了某项目在编码阶段实施策划时确定的任务网络图，T03、T06、T09 3 个任务节点连接成网络图中的关键路径；在进度跟踪过程中，还需要定期对该网络图进行分析，根据实际情况适时调整网络图，识别可能的新的关键路径。

（下转第 75 页）