

|            |         |              |
|------------|---------|--------------|
| 深圳龙视传媒有限公司 | 文 档 编 号 | LVM_PM_P_005 |
|            | 版本号     | V1.0         |
| 文档名称       | 估算指南    |              |

# 估算指南

|    |     |    |           |
|----|-----|----|-----------|
| 拟制 | 黄冬明 | 日期 | 2012-7-10 |
| 审核 | 何萍  | 日期 | 2012-8-30 |
| 批准 | 易睿  | 日期 | 2012-8-30 |

## 修订记录

| 日期         | 修订版本 | 修改描述 | 作者  | 评审成员 |
|------------|------|------|-----|------|
| 2012-07-10 | V1.0 | 新建   | 黄冬明 |      |
|            |      |      |     |      |
|            |      |      |     |      |
|            |      |      |     |      |
|            |      |      |     |      |

## 目录

|       |                      |           |
|-------|----------------------|-----------|
| 1     | 简介 .....             | 错误！未定义书签。 |
| 1.1   | 目的 .....             | 错误！未定义书签。 |
| 1.2   | 适用范围 .....           | 错误！未定义书签。 |
| 1.3   | 术语表 .....            | 错误！未定义书签。 |
| 2     | 角色与职责 .....          | 错误！未定义书签。 |
|       | 项目核心组成员 .....        | 错误！未定义书签。 |
| 3     | 过程总体描述 .....         | 错误！未定义书签。 |
| 3.1   | 概述 .....             | 错误！未定义书签。 |
| 3.2   | 进入和退出准则 .....        | 错误！未定义书签。 |
| 3.3   | 流程图 .....            | 错误！未定义书签。 |
| 4     | 主要活动描述 .....         | 错误！未定义书签。 |
| 4.1   | 任务分解分配 .....         | 错误！未定义书签。 |
| 4.1.1 | 活动说明 .....           | 错误！未定义书签。 |
| 4.1.2 | 输入 .....             | 错误！未定义书签。 |
| 4.1.3 | 输出 .....             | 错误！未定义书签。 |
| 4.2   | 执行任务 .....           | 错误！未定义书签。 |
| 4.2.1 | 活动说明 .....           | 错误！未定义书签。 |
| 4.2.2 | 输入 .....             | 错误！未定义书签。 |
| 4.2.3 | 输出 .....             | 错误！未定义书签。 |
| 4.3   | 每周更新项目风险状态 .....     | 错误！未定义书签。 |
| 4.3.1 | 活动说明 .....           | 错误！未定义书签。 |
| 4.3.2 | 输入 .....             | 错误！未定义书签。 |
| 4.3.3 | 输出 .....             | 错误！未定义书签。 |
| 4.4   | 定期更新项目进展 .....       | 错误！未定义书签。 |
| 4.4.1 | 活动说明 .....           | 错误！未定义书签。 |
| 4.4.2 | 输入 .....             | 错误！未定义书签。 |
| 4.4.3 | 输出 .....             | 错误！未定义书签。 |
| 4.5   | 定期跟进项目问题列表 .....     | 错误！未定义书签。 |
| 4.5.1 | 活动说明 .....           | 错误！未定义书签。 |
| 4.5.2 | 输入 .....             | 错误！未定义书签。 |
| 4.5.3 | 输出 .....             | 错误！未定义书签。 |
| 4.6   | 分析偏差是否影响里程碑 .....    | 错误！未定义书签。 |
| 4.6.1 | 活动说明 .....           | 错误！未定义书签。 |
| 4.6.2 | 输入 .....             | 错误！未定义书签。 |
| 4.6.3 | 输出 .....             | 错误！未定义书签。 |
| 4.7   | 持续检查项目阶段任务是否完成 ..... | 错误！未定义书签。 |
| 4.7.1 | 活动说明 .....           | 错误！未定义书签。 |
| 4.7.2 | 输入 .....             | 错误！未定义书签。 |
| 4.7.3 | 输出 .....             | 错误！未定义书签。 |
| 5     | 详细裁剪指南 .....         | 错误！未定义书签。 |

## 1 简介

### 1.1 目的

本文的目的是描述如何对项目产品规模、工作量等的估算。

### 1.2 适用范围

本指南属于参考性文档，公司所有产品开发项目都可以根据需要参考本指南。

### 1.3 术语表

**Wideband Delphi：**是一种鼓励参加估算的人员之间就相关问题进行讨论的方法。

**Pert：**这种方法共估算三个值：一般值、最大值和最小值。通过这三个值的计算可得到一个统计学上的期望值和一个标准偏差。

**类比法：**通过在历史数据库中寻找与新项目在应用、环境和复杂度方面类似的项目，利用其历史数据估算新项目的产品规模、工作量。

### 1.4 参考资料

《PMC 过程定义》

《PP&IPM 过程定义》

## 2 概述

### 2.1 角色和职责

#### 2.1.1. 项目经理

项目经理负责组织项目核心组成员进行项目工作任务的估算工作，并对估算结果进行汇总和整理。

#### 2.1.2. 项目核心组成员

负责本领域的估算，并将估算结果提交给项目经理。

#### 2.1.3. PQA

PPQA 负责对质量保证活动进行估算，并对整个估算过程进行引导与监督。

#### 2.1.4. 配置管理工程师

配置管理工程师负责对配置管理活动进行估算。

#### 2.1.5. 项目评审组

项目评审组负责评审基于估算结果形成的计划类文件。

### 2.2 入口准则

需求开发完成。

### 2.3 输入

系统需求规格说明书。

### 2.4 输出

评审通过的包含产品规模和工作量估算结果的文件。

### 2.5 出口准则

项目的开发工作完成。

## 3 项目估算

### 3.1 项目估算概述

每一个项目都需要进行估算，并将估算的结果作为制订计划的基础。估算是项目计划的核心，目的是为项目建立合理的预算，确定合适水平的人力资源，并为项目承诺提供基础。如下图所示：

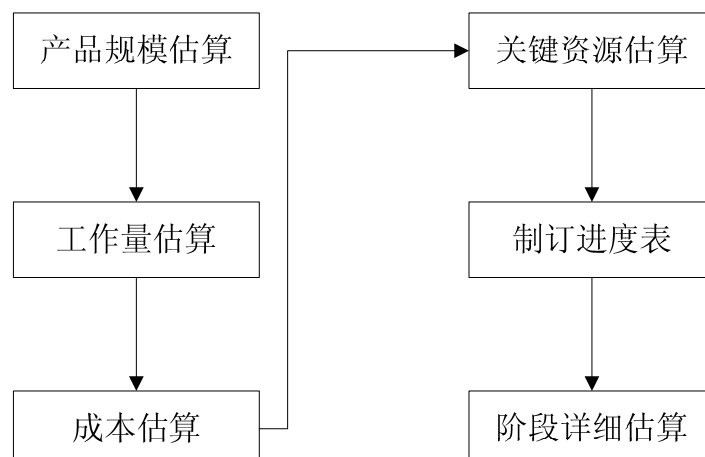


图 1 项目估算示意图

估算过程贯穿了整个项目。在项目的早期阶段，使用自上而下的方式进行粗略的估算。项目进展过程中，从下到上的方法被用于详细阶段计划来为子阶段进行更详细的估算，并更新总的项目估算。

项目经理负责估算任务的分配，项目核心组成员对各领域分别进行估算；项目核心组成员在项目经理组织下，共同完成项目的各项估算，最后由项目经理对估算结果进行整合。

估算的结果作为项目计划的一部分一起进行评审。

## 3.2 产品规模估算

产品规模是指产品的大小，估算步骤如下：

- ✓ 以需求为依据，进而对项目产品的工作分解结构进行分析，从而实现对产品规模的估算。
- ✓ 选择合适的估算单位。软件产品的规模估算可选用包括注释的千代码行（KLOC），或不包括注释的千代码行（KSLOC），还可以是功能点，硬件产品的规模估算选用模块数，结构件产品的规模估算选用零件数。有时需要估算个别工作产品的规模，可以考虑其他单位，如：页数，用于估算技术文档规模。
- ✓ 选择方法进行产品规模的估算，规模估算可采用 Wideband Delphi、Pert 方法、类比法、代码重用法等。
- ✓ 如果有类似项目的历史数据，要使用这些数据对估算进行调整和验证。参考项目组以前或者从项目组外获得的可以利用的工作产品的规模进行估算。

用于项目规模估算的假设条件要写入文档。

### 3.2.1 软件产品估算

#### 3.2.1.1 功能点

功能点估算是一种对软件功能、大小进行间接的、量化的估算方法，基于外部应用接口和内部应用复杂性的主观判断以及全局性能特点综合考虑。

功能点从用户的角度体现了一个应用的大小。它通过对主要的外部数据输入，输出，和文件类型等相关的信息处理功能的数量来估算软件应用。

面向功能的软件估算是软件对软件开发过程的间接估算。它集中于程序的功能性和实用性。

1. 指定一个负责人管理功能点估算。负责人员要估算基于外部应用接口的未被调整的功能点，和评估基于应用复杂性和性能的调整因子。负责人按照需要从技术人员那里获得相关数据和支持。
2. 以已经通过评审的需求文档为基础，参照历史数据和任务分解结果。
3. 按照以下过程确定功能点：

首先要确定五个数据域的特征，并在表（一）中相应位置给出计数。数据域的值以如下方式定义：

- ✓ 用户输入数：每个用户输入应是面向不同应用的输入数据，对它们都要进行计数。输入数据应区别于查询数据，它们应分别计数。
- ✓ 用户输出数：各个用户输出是为用户提供的面向应用的输出信息，它们均应计数。在这里的“输出”是指报告、屏蔽信息、错误信息等，在报告中的各个数据项不应再分别计数。
- ✓ 用户查询数：查询是一种联机输入，它引发软件以联机方式产生某种即时响应。每一个不同的查询都要计数。
- ✓ 文件数：每一个逻辑主文件都应计数。这里的逻辑主文件，是指逻辑上的一组数据，它们可以是一个大的数据库的一部分，也可以是一个单独的文件。内部文件（ILF）指每个逻辑主文件（即数据的一个逻辑组合，它可能是某大型数据库的一部分或者是一个独立的文件），例如数据库表。注意不是一个数据库表就是一个 ILF，例如合同数据可以包括合同信息、合同条款、合同付款计划。
- ✓ 外部接口数：对所有用来将信息传送到另一个系统中的接口（即磁带、磁盘、和可读写光盘上的数据文件）均应计数。

表（一） 功能点估算的计算

| 信息域参数 | 计数 | 运算符 | 加 权 因 数 |    |    | 运算符 | 加权计数 |
|-------|----|-----|---------|----|----|-----|------|
|       |    |     | 简单      | 中间 | 复杂 |     |      |
| 用户输入数 |    | ×   | 3       | 4  | 6  | =   |      |
| 用户输出数 |    | ×   | 4       | 5  | 7  | =   |      |
| 用户查询数 |    | ×   | 3       | 4  | 6  | =   |      |
| 文件数   |    | ×   | 7       | 10 | 15 | =   |      |
| 外部接口数 |    | ×   | 5       | 7  | 10 | =   |      |
| 总计数   |    |     |         |    |    |     |      |

对以上五个数据域，每一个数据域都有不同的复杂性，因此应计算出与每一个计数相关的加权复杂性值。使用功能点时要根据项目特点和开发单位的情况确定准则。计算功能点一般使用如下关系式：

$$FP = \text{总计数} \times (0.65 + 0.01 \times \text{sum}(Fi))$$

其中总计数是由表（一）所得到的所有加权复杂性值的和； $Fi$ （ $i = 1$  到  $14$ ）是复杂性校正值，它们应通过逐一回答表（二）所提问题来确定。 $\text{Sum}(Fi)$  是求和函数。上述等式中的常数和应用于数据域计数的加权因数需根据经验确定。

表（二） 计算功能点的校正值

| $Fi$  | 因素评定列表                      |
|---|-----------------------------|
| 1   | 系统需要可靠的备份和复原吗？              |
| 2   | 需要数据通信吗？                    |
| 3   | 有分布处理功能吗？                   |
| 4   | 性能很关键吗？                     |
| 5   | 系统是否在一个已有的、很实用的操作环境中运行？     |
| 6   | 系统需要联机数据项吗？                 |
| 7   | 联机数据项是否需要在多屏幕或多操作之间切换以完成输入？ |
| 8   | 需要联机更新主文件吗？                 |
| 9   | 输入、输出、文件或查询很复杂吗？            |
| 10  | 内部处理复杂吗？                    |
| 11  | 代码需要被设计成是可复用的吗？             |
| 12  | 设计中需要包括转换及安装吗？              |
| 13  | 系统的设计支持不同组织的多次安装吗？          |
| 14  | 应用的设计方便用户修改和使用吗？            |
| 注： 评定每个因素的尺度是 0 – 5：<br>0： 没有影响<br>1： 偶然的<br>2： 适中的 |                             |

4. 计算出功能点之后，如果需要，可以用来规范生产率、质量及其他属性的测量：

每个功能点的缺陷数

每个功能点的成本



每个功能点的文档页数

每人月完成的功能点数。

#### 1.1.1.1 代码行

对源代码行数的统计是对软件产品的一种直接的估算。估算代码行数通常需要对待开发的软件的类型有一定经验，有可用的历史数据，和一个代码行数的标准定义。

代码行估算通常用不包括注释的代码行（SLOC）或千代码行（KSLOC）来计算。这些代码可能是新开发的，也可能是修改已有的代码，两者的规模估算同等重要。

如果选用了代码行估算，可以使用 Wideband-Delphi 方法。

代码行方法的估算是对软件和软件开发过程的直接估算。

1. 指定一个负责人管理代码行估算。
2. 以通过评审的需求文档为基础，参照 WBS 和历史数据。可以按照以下过程进行估算：

对每一个功能或问题域计算三点值和期望值。估算变量（规模）的期望值 **EV**(expected value), 可以通过乐观值( $S_{opt}$ )、可能值( $S_m$ )、及悲观值( $S_{pess}$ )估算的加权平均值来计算：

$$EV = (S_{opt} + 4S_m + S_{pess}) / 6$$

其中给予可能值估算以最大的权重，并遵循 B 概率分布。

经过功能分解，确定问题域之后，将每个功能描述的期望值相加就得到总代码行数的估算值。比如项目 XYZ 经过 WBS 之后有 5 个功能，参照下表：

XYZ 项目 LOC 方法的估算表

| 功能             | 乐观值 $S_{opt}$ | 可能值 $S_m$ | 悲观值 $S_{pess}$ | 期望值 <b>EV</b> (LOC) |
|----------------|---------------|-----------|----------------|---------------------|
| 功能描述 1         | 1240          | 3100      | 4500           | <b>3023</b>         |
| 功能描述 2         |               |           |                |                     |
| 功能描述 3         |               |           |                |                     |
| 功能描述 4         |               |           |                |                     |
| 功能描述 5         |               |           |                |                     |
| <b>总代码行数估算</b> |               |           |                |                     |

参照以前做过的类似项目的历史数据如下表：

ABC 项目历史数据

| 阶段    | LOC（代码行） | 工作量 | 成本(千元) | 文档页数 | 缺陷 | 人数 |
|-------|----------|-----|--------|------|----|----|
| Alpha | 12100    | 24  | 168    | 365  | 29 | 3  |
| Beta  | 27200    | 62  | 440    | 1223 | 88 | 5  |
| gamma | 20200    | 43  | 314    | 1050 | 64 | 6  |

参照以前的历史数据，经过简单的转换，可以确定目前项目 XYZ 的生产率：

- ✓ 每千行代码（KLOC）的缺陷数
- ✓ 每千行代码（KLOC）的成本
- ✓ 每千行代码（KLOC）文档页数
- ✓ 每人月缺陷数
- ✓ 每人月代码行（LOC）
- ✓ 每页文档的成本

例如，计算每千行代码（KLOC）的缺陷数：

从 XYZ 项目 LOC 方法估算表中取出总的期望值 EV，再从 ABC 项目历史数据表中取出 gamma 阶段的代码行 X 和缺陷数 Y。这样可以计算出每千行代码（KLOC）的缺陷数=EV \* (Y / X) \* 1000。

其他各项目生产率数据可以依次类推。

3. 可以充分利用以前项目的经验，结合 Wideband-Delphi 方法来进行估算。

### 3.3 工作量估算

依据规模估算结果，对工作量进行估算，估算单位采用“人天”。工作量估算的内容并不仅仅限于项目开发部分，还包括其它的管理和支持活动的工作量，工作量主要包括：

软件开发的工作量，包括设计、编码、单元测试、集成测试等。

系统测试工作量，包括测试用例的编写、测试环境的搭建、系统功能测试、指标测试、认证测试等。

项目管理活动的工作量。

项目支持活动的工作量，包括：配置管理的工作量，质量保证活动的工作量以及培训的工作量。

注：外包任务的工作量不包括在内。

估算步骤如下：

- ✓ 对工作量的组成进行分解。
- ✓ 选择方法进行估算，可以采用 Wideband-Delphi 方法、Pert 方法等。
- ✓ 如果有类似项目的历史数据，要使用这些数据对估算进行调整和验证。

### 3.3.1 参考做法

可以使用 Wideband-Delphi 方法、Pert 方法等先得到开发工作量，根据一般的经验：

项目管理活动工作量 = 开发工作量的 15%~20%。

配置管理工作量 = 开发工作量 × 5%

质量管理工作量 = 开发工作量 × 5%

培训项工作量 = 开发工作量的 3%

工作量的分解可参考以下结构：

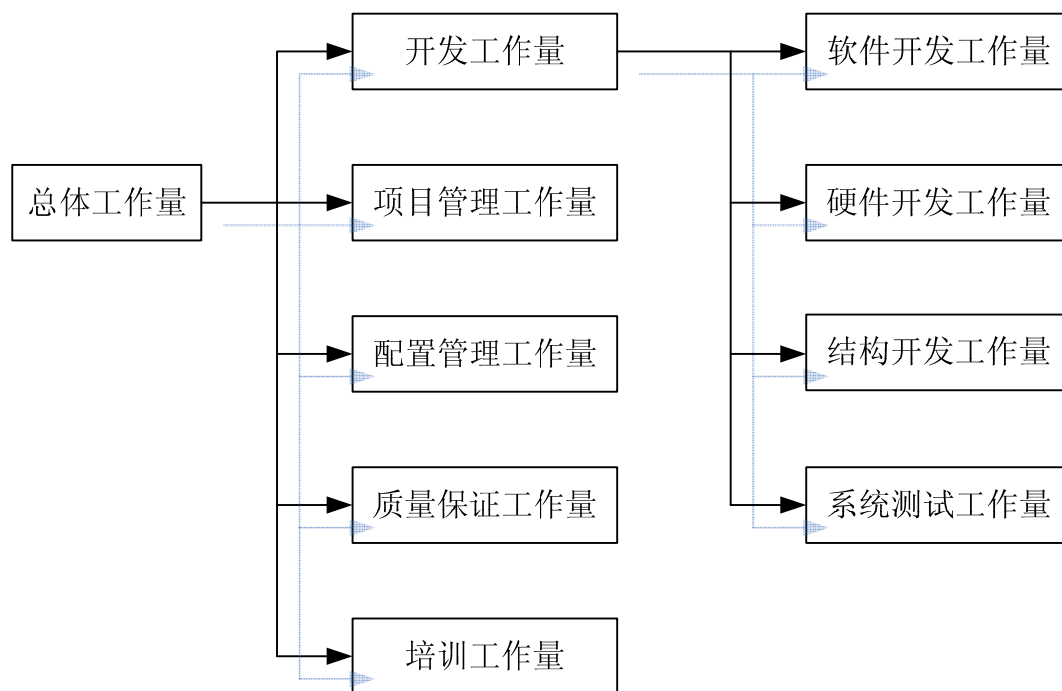


图2 工作量分解示意图

### 3.4 成本估算

估算出工作量后，进行进一步相对准确的成本估算。估算的内容有：

- ✓ 人力成本，人力成本（按日）\*（工作量）。
- ✓ 外包费用、认证费用、专利费用等。
- ✓ 差旅费、项目组活动费用等。
- ✓ 其他费用。

### 3.5 关键资源估算

项目经理应该明确完成项目所需的关键资源，并进行关键资源估算。关键资源可以是服务器环境、测试环境、目标环境等。

估算关键资源的步骤如下：

- ✓ 关键资源是在项目开发和操作中受到约束的硬件资源。项目经理根据《系统需求规格说明书》中

对关键资源的描述，确定关键资源的约束。

- ✓ 如果有关键资源，则要估算项目开发或运行中的资源状况。
- ✓ 与估算的项目产品规模、计划的处理负载、预计的数据量、或其他因素相联系，来估算所需的资源的总量，作初始估算。
- ✓ 当初始估算做好后，应对每一项资源建立一定的储备。
- ✓ 设计或者选用资源的限度作为跟踪的目标，确定跟踪的方法。
- ✓ 与有类似项目经验的人员讨论估算，并同其达成一致意见。

## 4 估算过程常用方法介绍

### 4.1 Wideband Delphi 方法

在历史数据不完全的情况下（如开发新项目，项目核心组对开发所涉及领域不够熟悉等），可采用本方法。

Delphi 方法是一种鼓励参加估算的人员之间就相关问题进行讨论的方法，用这种方法能充分发挥集体的力量，使估算的结果更切合实际。下面的表格是使用此方法估算的步骤：

| 步骤 | 活动                                |
|----|-----------------------------------|
| 1. | 召集人召集所有参加估算的人员，并将项目的需求和估算用表格分发给大家 |
| 2. | 召集人召集所有参加估算的人员进行一个会议，讨论有关估算的问题    |
| 3. | 参加估算的每个人匿名的填写估算表格                 |
| 4. | 召集人收集所有的估算表格，然后形成反馈表返回给参加估算的人员    |
| 5. | 召集人召集所有参加估算的人员进行一个会议，主要是讨论估算上差异   |
| 6. | 参加估算的人员根据讨论的结果，在反馈表上提交另一个匿名的估算    |
| 7. | 重复 4~6 直到达成关于项目估算项最大程度的一致         |

下面是利用 Delphi 方法进行软件规模估算的例子，也可以用来估算工作量。使用时请注意相应的估算量的单位。举例说明如下：

假如一个由 a、b 两个模块组成的项目：甲是召集人，乙和丙是参加估算的人。

首先，甲将项目相关的资料和估算表格发给乙和丙，并召开一个甲乙丙参加的会议，分析项目的问题。

乙估算 a 模块用 2000 行代码，b 模块用 6000 行代码，丙估算 a 模块用 4000 行代码，b 模块用 8000 行代码，然后乙和丙分别匿名的填写估算表格，甲收集表格后，进行汇总，然后将下面表格反馈给乙和丙。

| 项目             |          | 表编号      |  |       |  |         |
|----------------|----------|----------|--|-------|--|---------|
| 第 1 轮估计        |          |          |  |       |  |         |
| 单位             | 代码行      |          |  |       |  |         |
|                |          |          |  |       |  |         |
|                | 参与者1的估计值 | 参与者2的估计值 |  | 平均值   |  | 下一轮的估计值 |
|                |          |          |  |       |  |         |
| 模块a            | 2000     | 4000     |  | 3000  |  |         |
| 模块b            | 6000     | 8000     |  | 7000  |  |         |
|                |          |          |  |       |  |         |
| 总工作量           | 8000     | 12000    |  | 10000 |  |         |
|                |          |          |  |       |  |         |
| 备注：            |          |          |  |       |  |         |
| 在这里写下需要特殊说明的问题 |          |          |  |       |  |         |
|                |          |          |  |       |  |         |

注：

表中的“参与者”为匿名，不写具体的姓名。

此表的总份数应为： 总轮数 × 参与估算的人数。

以上的估算反馈表可直接作为估算表使用。在每轮估算中，每个参与者只需填写各模块与总工作量的“下一轮的估算值”即可。召集人回收此表后，负责在新的表中填写每个参与者的估算值并计算平均值，再将新表发给每个参与者。

作为简化的情况，上表中各个“参与者估算值”可不列出。

甲召集乙和丙举行一个会议讨论估算的差异，然后分别重新估算，填写估算表，甲收集后重新汇总，形成新的反馈表交给大家，重复这个过程，直到乙和丙在规模上达成一致，完成对软件规模的估算。

## 4.2 Pert 方法

在有一定历史数据，估算人员对项目涉及领域有较多的了解的情况下可使用本方法。

Pert 方法首先是个人根据经验，给出三个估算值：估算项一般值、最大值和最小值。然后通过这三个值的计算可得到一个统计学上的期望值和一个标准偏差。

Pert 公式估算的预期开发工作量是 E，标准偏差是 SD：

$$E = (a + 4b + c) / 6$$

$$SD = (c - a) / 6$$

举例说明，假如一个项目 a 工作量估算：

a=最小可能的工作量 100 人天

b=正常工作量 120 人天

c=最大可能开发工作量 150 人天

那么根据 Pert 公式估算的预期开发工作量是 E，标准偏差是 SD：

$$E = (100 + 4(120) + 150) / 6 = 121 \text{ 人天}$$

$$SD = (150 - 100) / 6 = 8 \text{ 人天}$$

这就是说，有 68% 的可能的 workload 在 113 人日（121-8）和 129 人日（121+8）之间。

本方法的前提是工作量的估算没有偏见，经验表明估算偏低的倾向大于偏高的倾向，使用时应加以考虑。

## 4.3 类比法

在历史数据较完备时可使用，且准确程度较高。估算人员有相关领域开发的经验，项目涉及的应用、环境较复杂，当待估算项目与已完成项目在应用、环境和复杂度方面相类似时，可以使用本方法。

基本步骤如下：

- 从组织过程资产库和文档库中找到类似项目的相关估算数据和文档。
- 如果是进行规模估算，则列出已完成项目中可类比的功能点数或代码行数；如果是进行工作量估算，则列出已完成项目中可类比的模块的工作量；如果是进行文档数估算，则列出已完成项目中

可类比的代码页数。

- 标识待估算项目和已完成项目可类比项的差异。
- 依据 2、3 步的结果进行估算，形成对产品规模、工作量或文档的估算。

这种估算的准确性依赖于已完成项目的完成程度和数据的准确程度，因此使用这种方法要求有一个内容丰富、准确、可靠的组织过程资产库。

## 4.4 专家判断法

需要使用者有类似系统的经验。一般只适用于小项目，由熟知此项目的专家来进行估算。

步骤如下：

- 结合工作分解，根据经验估算工作中每项功能的产品规模、工作量。
- 根据每项任务实现的难度、风险等，适当调整产品规模、工作量的估算。

本方法的特点是完全依赖个人水平，有较大的风险。

## 4.5 代码重用法

本方法仅适用于对软件可重用模块的规模估算，如软件的升级版本等情况。

本方法涉及三个参数，分别是新设计的百分比（%Redesign）、重新编码的百分比（%Recode）和重新测试的百分比（%Retest），使用这三个参数可以计算使用重用模块的代码规模计算公式为：

$$[(\%Redesign + \%Recode + \%Retest)/3] \times \text{Existing Code} = \text{Equiv SLOCs}.$$

举例说明，如果想重用有一个有 10000 行代码的模块，其中 40% 需要重新设计，50% 需要重新编码，60% 需要重新测试，那么等价的新开发的代码量是：

$$([40\% + 50\% + 60\%]/3) \times 10000 = 5000 \text{ SLOC}$$

## 4.6 估算时方法的综合运用

各种方法并不是互相排斥的，几种方法可以相互嵌套，并行使用。例如，在使用 Wideband Delphi 方法时，每个参与估算的人员可以用 Pert 法做出估算。



其次，对于选用何种方法，可从两个方面考察项目，即项目规模与历史数据的完备性。

项目规模可分为大、中、小项目。

历史数据的完备性取决于许多因素有关，比如项目是新项目还是做改制类项目，项目核心组对开发所涉及领域的熟悉程度，人员的技能，以前对类似项目的记录是否完备与系统等等。