软件估计指南

Q/HX-G5B-B03-C03

成都中科合迅科技有限公司

Chengdu Sinux Tech Co.，ltd

2023-01-31 发布 **2023-02-01 实施**

软件估计指南

文件编号：Q/HX-G5B-B03-C03

版 本 号：V1.0

编　　制：冉　婷　　日期　2023.01.31

审　　核：林正权　　日期　2023.01.31

批　　准：李　牧　　日期　2023.01.31

成都中科合迅科技有限公司

Chengdu Sinux Tech Co.，ltd

2023-01-31 发布　　**2023-02-01** 实施

文档修订记录

版本号	修改内容描述	修改人	日期	备注
D1.0.1	编制初稿	冉　婷	2023 年 01 月 13 日	
V1.0	评审后发布	冉　婷	2023 年 01 月 31 日	

目录

1 目的及范围

本文的目的是为软件估算提供指南。

本文档的适用范围为组织中所有的软件项目。

2 引用文件

QHX-G5B-B03-2023 《项目策划（PP）实践域》

3 规模估算的方法

下面说明一些常用的估算方法，它们是 Wideband Delphi Technique、Pert Sizing Technique、Sizing By Analogy、Reused Code。

3.1 基本模块估算法

基本模块估算方法，是先详细估算一个基本软件模块的难度、复杂度等因素，确定其软件规模，并以此作为单位基准。

进行其他模块的估算时，以基本软件模块作为标准，衡量其他软件模块的规模。

3.2 Wideband Delphi 技术

Wideband Delphi Technique 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，使用这样的方法估算的步骤详见表 1：

表 1 Wideband Delphi 估算步骤

步骤	活动
1.	召集人召集所有参加估算的人员，并将估算内容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，讨论有关软件规模的问题。
3.	参加估算的每个人提交估算结果。

4.	参加估算的人员讨论估算结果差异（项目组自行定义差异阈值，一般推荐 20%），给出最大值和最小值的估算人员要做出解释。
5.	如果差异超过阈值，参加估算的人员根据讨论的结果，提交另一次估算。
6.	重复 4~6 直到达成关于软件规模最大程度的一致。

估算反馈表的格式见

图 **1**：

范围: *下次:*

模块A:

20 40 60 80 _____SLOC

模块B:

20 40 60 80 _____SLOC

模块C:

20 40 60 80 _____SLOC

模块D:

20 40 60 80 _____SLOC

模块E:

20 40 60 80 _____SLOC

模块F:

20 40 60 80 _____SLOC

模块G:

20 40 60 80 _____SLOC

X - 估计
X* - 你的估计
X! - 估计的中间值

请给出你下一轮的估计(合计)：___ SLOC

备注:
在这里写下需要特殊说明的问题

图 1估算表反馈格式

举例说明如下：

假如由一个由 a、b 两个模块组成的项目：例子项目，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 4000 行代码，丙估算 a 模块用 6000 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，反馈给乙和丙的表格见图 2。

项目：例子项目　　日期：2001/6/1
这是第 1 轮的估计范围：

	范围:				*下次:*
模块A:					
	2k(x)	4k(x!)	6k(x)		1(
	2k	4k	6k	8k	1 ___SLOC
模块B:					
		4k(x)	6k(x!)	8k(x)	1(
	2k	4k	6k	8k	1 ___SLOC
模块C:					
	20	40	60	80	1 ___SLOC
模块D:					
	20	40	60	80	1 ___SLOC
模块E:					
	20	40	60	80	1 ___SLOC
模块F:					
	20	40	60	80	1 ___SLOC
模块G:					
	20	40	60	80	1 ___SLOC

X - 估计
X* - 你的估计
X! - 估计的中间值

请给出你下一轮的估计(合计）：__ SLOC

备注：
在这里写下需要特殊说明的问题

图 2 反馈表格

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，直到得出的

偏差在可接受范围内（例如第一阶段偏差为 40%，第二阶段偏差为 20%，第三阶段偏差为 10%即为可接受的规模估算），完成对软件规模的估算。

3.3 Pert Sizing 技术

这种方法共估算三个值： 软件产品预期规模的一般值、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。

公式：PERT 公式估算的预期规模是 E，标准偏差是 SD:

$$E = \frac{a + 4b + c}{6}, \qquad SD = \frac{c - a}{6}.$$

举例说明，假如一个新的通信程序：

a=最小可能的规模，例如：10KSLOC；

b=软件产品的正常规模，例如 12KSLOC；

c=软件产品的最大可能规模，例如 15KSLOC；

那么根据 PERT 公式估算的预期规模是 E，标准偏差是 SD:

$$E = \frac{a + 4b + c}{6}, \qquad SD = \frac{c - a}{6}.$$

既：

$$E = \frac{10 + 4(12) + 15}{6} = 12.167 \text{ KSLOC}$$

$$SD = \frac{15 - 10}{6} = 0.833$$

这就是说，有 68%的可能规模会在 11.334（12.167-0.833）和 13（12.167 + 0.833）之间。本估算方法的前提是对规模的估算没有偏见，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。

3.4 Sizing By Analogy（类比法）

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，可以使用本估算方法。

本估算法的基本步骤如下：

a) 从软件过程数据库和文档库中找到类似项目的相关估算数据和文档。

b) 列出已完成项目中可类比的功能点和完成这些功能点的代码行数。

c) 标识待估算项目和已完成项目可类比功能点之间的差异。

d) 依据 b、c 步的结果进行估算,形成对软件产品规模的估算。

很明显,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,因此使用这种估算方法要求有一个内容丰富、准确、可靠的软件过程数据库。

3.5 可重用代码

本方法仅适用于对可重用模块的规模估算。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) x Existing Code = Equiv SLOCs

举例说明，如果想重用一个有 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）*10000 = 5000 SLOC

改善项目估算，本方法适合用于估算改善一个项目的规模，改善一个项目的规模和待改善的项目的规模相关，通过将待改善项目的规模乘以改善项目规模系数，可以估算出改善一个项目的规模，可以用如下的公式表示：

规模 = 待改善项目的规模 * 改善项目规模系数

3.6 IFPUG 方法

IFPUG 方法的基本思想是将软件的功能进行不断分解，识别每个基本功能，判断其复杂度（低/中/高），乘以相应的权重，得到该功能的功能点数，计算总的功能点数。

IFPUG 方法中包括五种基本功能：

a） 数据功能：

1） ILF（Internal Logical File）

2） EIF（External Interface File）

b） 事务处理功能：

3） EI（External Input）

4） EO（External Output）

5） EQ（External Inquiry）

方法的流程见图 3：

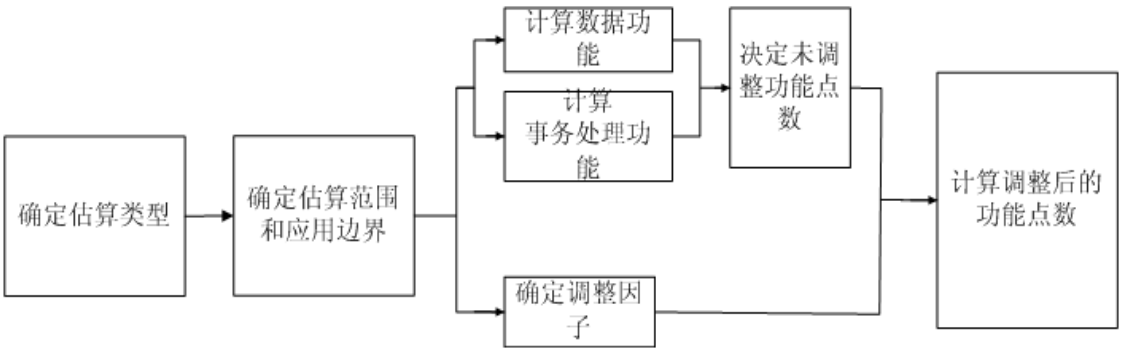

图 3 方法流程图

a） 确定估算类型

有三种估算类型：新开发项目功能点计算、升级项目功能点计算、应用程序功能点计算。其中，新开发项目功能点计算测量的是项目完成时，第一次交付安装的版本的功能点数，反映项目第一次开发的规模。升级项目功能点计算测量的

是对现有应用程序的修改，即，新增功能+改变的功能+删除的功能。应用程序，也称基线功能点计算，测量的是已安装的应用程序，为最终用户提供的实际功能点数。

b） 确定估算范围和应用边界

确定计算范围是计算全部的功能，还是某类特定的功能。应用边界指的是该应用程序与其它外部程序/用户域之间的边界。这是从用户角度来看的，并且是基于业务功能，而不是基于技术方面来考虑。

c） 计算数据功能

数据功能也称文件功能，指的是为满足用户的内部和外部数据需求而提供的功能（储存数据的容器）。是从商业用户的角度进行数据建模，来观察、识别的。在一定程度上与 E-R 图中的实体/关联对应。

数据功能包括 ILF（内部逻辑文件）和 EIF（外部接口文件）两种。

1） ILF：用户可识别的一组逻辑上相关的数据组或控制信息，并且在应用程序内部维护。

2） EIF：用户可识别的一组逻辑上相关的数据组或控制信息，它们被应用程序所使用，但是是在其它应用程序内部维护的。

3） 数据功能的复杂度（低/中/高），由该功能相关的 DET 和 RET 的数量（范围）决定。

4） DET （Data Element Type） ：用户可以识别的唯一的、不可重复的字段或属性。

5） RET （Record Element Type） ：数据功能中用户可识别的数据元素子集

数据功能复杂度矩阵：

	1 to 19 DET	20 to 50 DET	51 or more DET
1 RET	Low	Low	Average
2 to 5 RET	Low	Average	High
6 or more RET	Average	High	High

图 4 数据功能复杂度矩阵

对 ILF：

Functional Complexity Rating	Unadjusted Function Points
Low	7
Average	10
High	15

图 5 对 ILF

对 EIF：

Functional Complexity Rating	Unadjusted Function Points
Low	5
Average	7
High	10

图 6对 EIF

计数 DET 时的一些常用规则：

1） 同一个数据功能（ILF/EIF），在不同的应用程序中，具有不同的 DET；

2） 数据功能间的关联（相当于 foreign key）计为一个 DET；

3） 由于技术实现的原因，在一个数据功能中出现多次的 DET，只计为一个 DET；

4） 格式相同、具有相似意义的多个字段。

d） 计算事务处理功能

事务处理功能，指应用程序所提供给用户的用来处理数据的功能。它们都必须是一个基本处理，即，对用户有意义的、最小的活动单位，并且使程序的业务逻辑处于一致性状态。包括 EI（外部输入）、EO（外部输出）和 EQ（外部查询）三类。

1) EI：应用程序处理来自本应用程序边界之外的数据或控制信息。

2) EI 的基本目的是为了维护一个 ILF 或者改变系统的行为。EI 的一些例子：新增用户、修改用户信息、删除用户、将用户分配到某个项目。

3) EO：应用程序向边界之外提供数据或控制信息。

4) EO 的基本目的是向用户提供经过处理逻辑加工的信息（除了检索之外）。至少包含一个数学公式或计算、或生成衍生数据、或改变 ILF 或系统行为。EO 的一些例子：报表、多数柱状图/饼图、多数“打开文件”操作。

5) EQ：是一个向应用程序边界之外发送数据或控制信息查询的基本处理。

6) EQ 的基本目的是向用户展示提取的数据或者控制信息。逻辑处理里面不包含数学公式、计算、衍生数据的生成，不维护 ILF，不引起系统行为的改变。EQ 的一些例子：查询、浏览、不包含计算的简单报表、下拉列表（注意）、登录屏幕、帮助。

事务处理功能的复杂度（低/中/高），由该功能相关的 DET 和 FTR 的数量（范围）决定。

1) DET（Data Element Type）：用户可以识别的唯一的、不可重复的字段或属性。

2) FTR（File Type Referenced）：该事务处理功能所涉及到的文件功能（读取、维护 ILF 或 EIF）。

事务处理功能的复杂度矩阵：

EI 类型：

	1 to 4 DET	5 to 15 DET	16 or more DET
0 to 1 FTR	Low	Low	Average
2 FTRs	Low	Average	High
3 or more FTRs	Average	High	High

图 7EI 类型

EO/EQ 类型：

	1 to 5 DET	6 to 19 DET	20 or more DET
0 to 1 FTR	Low	Low	Average
2 to 3 FTRs	Low	Average	High
4 or more FTRs	Average	High	High

图 8 EO/EQ 类型

事务处理功能的功能点数：

EI/EQ 类型：

Functional Complexity Rating	Unadjusted Function Points
Low	3
Average	4
High	6

图 9EI/EQ 类型

EO 类型：

Functional Complexity Rating	Unadjusted Function Points
Low	4
Average	5
High	7

图 10EO 类型

计数 DET 时的一些常用规则

1） 进入/退出边界的每个用户可识别的字段计为一个 DET

2） （EO/EQ）既进入又退出应用程序边界的字段，只计一个 DET

3） 未穿越系统边界的字段不是 DET

4） 引发一个基本处理的多种方法，只计为一个 DET

5） 出错信息，确认操作成功等信息只计一个 DET

6） （EO/EQ）报表标题，栏标题等不是 DET

7） （EO/EQ）页码信息和系统标签（例如时间标记）不是 DET

8） 从下拉列表中选择一项，或选择某个单选按钮

9） 饼图、柱状图常包含 2 个 DET

e) 确定调整因子

UFP 只考虑到了每个具体的功能及其复杂程度，而没有考虑项目的总体特征。根据 14 个系统基本特征 GSC（General System Characteristic），计算出调整因子 VAF（Value Adjustment Factor），来对 UFP 加以调整。

注意：调整因子争议较多，功能点分析国际标准（ISO-14143）已删除了这一步。因此这一步是可选的。通常的做法：放到成本估算（COCOMO II 模型）中去考虑。FPA 只反映功能需求（Functional Size），不反映质量需求或技术需求。

14 个 GSC：

1) 数据通讯（Data Communications）
2) 分布式数据处理（Distributed Data Processing）
3) 性能（Performance）
4) 使用强度高的配置（Heavily Used Configuration）
5) 处理速度（Transaction Rate）
6) 在线数据输入（Online Data Entry）
7) 最终用户的效率（End-User Efficiency）
8) 在线更新（Online Update）
9) 复杂的处理（Complex Processing）
10) 可重用性（Reusability）
11) 安装的简易性 （Installation Ease）
12) 运行的简易性（Operational Ease）
13) 多场地（Multiple Sites）
14) 允许变更（Facilitate Change）

对每个 GSC 进行打分，分数取值范围为 0（无关）~5（强关联）。

$$VAF = 0.65 + [\sum_{i=1}^{14} Ci / 100]$$

f) 决定未调整功能点数

将全部的数据功能、事务处理功能的功能点数相加，得到未调整的功能点数UFP（Unadjusted Function Points）。

g) 计算调整后的功能点数

用 VAF 对 UFP 进行调整，最终得到调整后的功能点数 AFP（Adjusted Function Points）。

1） 对新开发项目的计算：AFP＝(UFP＋CFP)×VAF

2） 对升级项目的计算：AFP＝[(ADD＋CHGA＋CFP)×VAFA]＋(DEL×VAFB)

3） 典型情况下（升级后 VAF 不发生变化）简化为 AFP＝(ADD＋CHGA＋DEL＋CFP)×VAF

4） 对应用程序功能点数的计算：

初始：AFP＝ADD×VAF

升级后：AFP＝[(UFPB＋ADD＋CHGA)－(CHGB＋DEL)]×VAFA

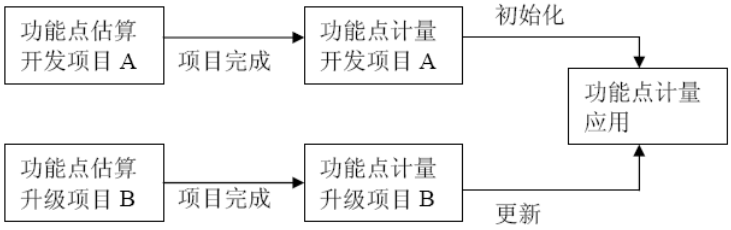

图 11对应用程序功能点数的计算

3.7 快速功能点估算方法

常见的情况：早期需求不完整

几种快速估算方法：

a) 将全部事务处理功能的复杂度默认为“中”，数据功能的复杂度默认为“低”

b) UFP=35*ILF 数+15*EIF 数 (UFP、ILF 和 EIF 相关知识参考 3.6 章节)

c) 根据历史数据

4 工作量估算的方法

关于工作量估算的方法有很多，除了 Delphi 方法外，下面简单介绍其中的几种方法，但使用时并不限于这几种方法，估算时可以使用其他的方法：

4.1 运算法（ALGORITHMIC COST MODELS）

运算法是一种简单直观的估算方法，它根据规模估算的结果和相应的系数运算得到工作量估算：

总工程工作量 wt = p*s*c*h

wt=软件模块或产品的总工程工作量

p=软件模块或产品的规模（SLOC）

s=开发人员生产力系数（1/每人月的代码行）

c=软件模块/产品复杂度

h=软件模块/产品技术难度

运算计算出的是软件模块或产品的总工作量，该软件模块在各个阶段的阶段工作量（如需求分析阶段的工作量、设计阶段的工作量等）或管理工作量（QA、CM 等活动）需要根据在该阶段中的工作量比例进行推算：

阶段工作量 wst = wt*a

wst=阶段工作量

a =某阶段工作量比例/某种类型活动的工作量比例

项目总工作量=项目工程工作量+ 项目管理工作量

4.2 专家判断法（EXPERT JUDGEMENT）

本估算方法需要使用者有类似系统的经验，结合规模估算的结果和 WBS 综合分析，形成对工作量的估算，可以按照下面的步骤进行：

a) 结合 WBS，根据经验和规模来估算 WBS 中每项功能的工作量；

b) 根据每个功能点实现难度和风险的大小，适当调整工作量的估算；

按阶段合计需要的工作量，形成总的工作量的估算。