

一种嵌入式软件项目估计方法

陈佳豫^{1,2} 李 杨^{1,2} 刘金国¹ 赵 莹^{1,2} 孔德柱¹

(中国科学院长春光学精密机械与物理研究所 长春 130033)¹ (中国科学院研究生院 北京 100039)²

摘 要 软件项目估计是 CMM2 级软件项目策划 KPA 的基础,是软件开发中的一个重要环节。合理的估计是保证软件项目符合预算和进度要求的前提条件。描述了基于 CMM 的软件估计的过程,介绍了一种基于嵌入式软件项目的估计方法,并结合具体实例对规模估计、工作量/成本估计、关键计算机资源估计和进度估计的内容进行了细致的阐述。该研究为有效地规划和管理嵌入式软件项目,制定合理可行的软件开发计划提供了有力的支持。

关键词 CMM,软件项目策划,KPA,软件项目估计

Estimation Method of Embedded Software Project

CHEN Jia-yu^{1,2} LI Yang^{1,2} LIU Jin-guo¹ ZHAO Ying^{1,2} KONG De-zhu¹

(Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China)¹

(Graduate School of the Chinese Academy of Sciences, Beijing 100039, China)²

Abstract Software project estimation is the base of software project planning KPA of CMM level 2, and it is one of important parts in the process of software development. Proper estimation is the precondition which ensure budget and schedule of software project. Described the process of CMM-based software estimation, and introduced an estimation method of embedded software project. Moreover, combining an example, illuminated in detail the contents of size estimation, effort/cost estimation, key computer resource estimation and schedule estimation. The research provides strong supports for effectively planning and managing software project, and making reasonable software development plan.

Keywords Capability maturity model(CMM), Software project planning(SPP), Key process area(KPA), Software project estimation

1 引言

随着软硬件技术的快速发展,嵌入式软件产业在国民经济中扮演着越来越重要的角色,特别在航空、航天、航海等高风险领域,对嵌入式系统的功能需求越来越多。同时,由于嵌入式软件开发的整个过程都是软硬件并行交互进行,因此嵌入式软件开发已经成为一项很复杂的系统工程,开发计划很难制定。就目前发展来看,在嵌入式软件项目的开发过程中,对项目作出一个合理的估计已成为确保项目能够按照预定的成本、进度顺利完成的必备前提条件。自上个世纪七十年代以来,出现了 CMM^[1], ISO9001, ISO/IEC12007, Botstrap 和 Spice 等众多的软件过程质量模型或软件质量保证标准,虽然上述标准的侧重点不同,但这些标准或模型都强调通过实施软件估计来为制定实施软件工程和管理软件项目的合理计划做准备,其中以卡内梅隆大学软件工程研究所(SEI)开发的软件能力成熟度模型(capability maturity model,即 CMM)影响最为深远。本文结合当前实施 CMM 的软件工程实践,描述了基于 CMM 的软件估计过程,结合具体嵌入式软件项目实例给出了一种基于嵌入式软件项目的估计方法,有针对性地介绍了嵌入式软件项目的规模估计、工作量/成本估计、关键计算机资源估计和进度估计的估计过程。

2 软件项目估计

软件项目估计,就是结合目前各种实际情况,提供项目中的软件规模、工作量和人力成本、设备资源以及进度的最可能合理的模型。它是一门科学,更是一门艺术,这个重要的活动不能以随意的方式来进行,因为它是所有其他项目计划活动的基础^[2]。如果低估软件项目估计会造成人力、成本预算低估,日程过短,最终人力资源耗尽,成本超出预算,为完成项目不得不赶工,影响项目质量,甚至导致项目失败,所以对于任何软件开发来说,好的估计异常重要。

适当的软件项目估计方法是 CMM2 级软件项目策划 KPA 的要求,CMM3 级的集成软件管理 KPA 也需要有好的估计方法来为项目制定计划^[3],具体估计的对象包括规模、工作量/成本、关键计算机资源和进度^[4]。

(1) 规模估计

规模,就是一个工程量化的结果,是用具体数字来体现项目的描述。软件的规模是决定编写软件需花费的工作量的主导因素。估计小系统的规模要容易得多,可以将系统分成一些更小的部分进行软件规模估计^[3]。规模估计的信息来源是用户需求,当需求不够清晰时,可采用快速功能点进行估计;需求清晰后,可采用功能点分析法进行估计^[4,5]。

陈佳豫 博士研究生,主要研究方向为软件工程、计算机控制;李 杨 博士研究生,主要研究方向为软件可靠性;刘金国 研究员,硕士生导师,主要研究方向为 CCD 图像传感器应用技术研究;赵 莹 博士研究生,主要研究方向为软件工程、数字图像处理;孔德柱 实习研究员,硕士研究生,主要研究方向为航天相机控制器。

(2)工作量/成本估计

工作量估计是对开发软件所需工作时间的估计,可以理解为单个人代表完成某个固定任务所花费的时间,它与组织的整体生产率紧密相关。成本估计包括人力、物质、有形及无形地支出成本估算。工作量和成本估计是根据规模估计值进行估算,具体包括自顶向下估计法和自底向上估计法两种策略^[3]。如果有历史数据,采用历史经验数据分析法估计;如果没有历史数据,采用简单生产率法估计^[4]。

(3)关键计算机资源估计

关键计算机资源估计是根据需求、工作产品的规模,确定和标识项目中的关键计算机资源,如存储器容量、处理器能力和通信通道容量等^[4]。

(4)进度估计

进度是工作量、人员以及人员技能综合因素的函数,以不同阶段的里程碑作为标志^[4]。进度估计需要充分考虑对客户承诺的进度、项目各阶段的工作量以及参与人员的综合因素,来合理安排人员,最终估算出各阶段所需的有效工作时间并制定进度计划。

3 基于嵌入式软件项目的估计

下面以相机控制软件为例,说明基于嵌入式软件项目的具体估计方法。

相机控制软件需要完成的功能包括:通讯、参数实时计算、数据管理、指令采集处理、调焦控制、相机工作控制、成像控制及辅助处理等。其功能框图如图1所示。

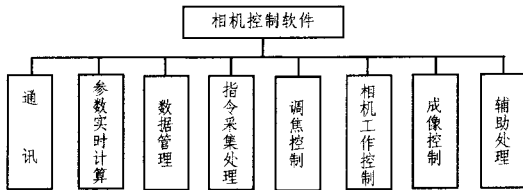


图1 相机控制软件系统功能框图

为使估算准确,估算方法采用1+4+1方法。公式为:

$$R=(1 \times A+4 \times B+1 \times C) / 6 \quad (1)$$

即对于规模、工作量/成本、关键计算机资源和进度估计过程中的每一项估计值均由三名具有不同经验的软件开发与设计人员分别进行估算,将三者的估算结果按由大到小顺序(用A,B,C分别表示)分别乘以权1,4和1然后除以6,所得结果作为最终的估算结果(R)。为便于说明,下列估算过程中使用的估算值均为已经按照公式(1)计算后取整的估算结果加以使用。

3.1 软件规模估计

3.1.1 功能分类

对相机控制软件进行功能分类,确定其内部逻辑文件ILF、外部接口文件EIF、外部查询EQ、外部输入EI和外部输出EO这5种功能类型的功能数^[4,5]。具体如下:

ILF:指所有需要在软件系统内部维护的数据或控制信息。确定相机控制软件的ILF为2个,包括相机数据文件和控制信息文件。

EIF:指所有由软件系统引用,不在系统内维护的数据或控制信息。确定相机控制软件的EIF为2个,包括通过总线接收的相关信息文件。

EQ:指包括输入和输出构件的基本处理过程。对于相机

控制软件的EQ为0个。

EI:指数据由外向内跨越边界的基本处理过程。确定相机控制软件的EI为14个,包括通讯接收的各种指令、数据以及各种控制成像、调焦的信息。

EO:指导出的数据由内向外跨越边界的基本处理过程。确定相机控制软件的EO为2个,包括相关的输出数据和控制序列。

3.1.2 计算功能点

如果需求清晰,采用功能点分析法估计;需求不够清晰,采用快速功能点法进行估计。下面分别介绍两种计算功能点的方法。

(1)功能点分析法

首先,计算未调整的功能点UFP(Unadjusted Function Points)^[4]。根据相机控制软件各功能类型的不同复杂度的权值和数据量计算出未调整功能点数,如表1所示。

表1 计算未调整功能点

功能	简单	中等	复杂	合计
内部逻辑文件	0×7	1×10	1×15	25
外部接口文件	1×5	1×7	0×10	12
外部查询	0×3	0×4	0×6	0
外部输入	5×3	9×4	0×6	51
外部输出	0×4	1×5	1×7	12
UFP				100

然后评估技术复杂度^[4]。对14个系统特征值(GSC)进行评估,得到对“未调整的功能点”的影响因子。该评估主要是通过考察系统的技术和运行特点等与非功能需求和设计约束相关的信息,确定14个GSC($F_i, i=1,2,\dots,14$)。将各评估值相加得总的的影响度,即 $TDI=\sum F_i (i=1,\dots,14)$,其中 F_i 的取值为0~5, TDI 的取值为0~70。通过计算得到相机控制软件 $TDI=45$ 。

最后,计算功能点数FPC。根据经验公式^[4]

$$FPC=UFP \times (0.65+0.01 \times TDI) \quad (2)$$

计算相机控制软件的FPC为110功能点。

(2)快速功能点法

通过考察相机控制软件系统的技术和运行特点,取其总的系统特征值调整因子 $VAF=1.10$,具体快速功能点法的计算方法如表2^[4]所示。

表2 快速功能点法功能点分析表

总的系统特征值	无	轻微	中度	一般	重大	极强
调整因子VAF=	0.65	0.80	0.90	1.10	1.20	1.35
功能类型	功能数	权值	UFP=功能数 × 权值		FPC=UFP × VAF	
ILF	2	10	20		22	
EIF	2	7	14		15.4	
EQ	0	4	0		0	
EI	14	4	56		61.6	
EO	2	5	10		11	
合计	—	—	100		110	

3.1.3 主要产品规模估计

根据嵌入式软件的特点,选取功能点与文档规模、测试用例的转化经验值分别为:需求文档规模为70页/100功能点;设计文档规模为100页/100功能点;测试用例为200个/100功能点。则相机控制软件的需求文档和设计文档页数分别估计为77页和110页;测试用例数估计为220个。

3.1.4 代码规模估计

相机控制软件使用汇编语言进行开发,根据经验取功能点与编程语言的转化值(FP-LOC)为 100 LOC/FP。因此将相机控制软件的功能点数 FPC 转化为代码规模: $LOC = FPC \times 100 = 11000$ 条。

3.2 工作量/成本估计

3.2.1 工作量估计

工作量 E 的计算公式为:

$$E = FPC/P \quad (3)$$

其中 FPC 为功能点数; P 为生产率。

生产率 P 是指一人一天能完成功能点的个数。根据嵌入式软件开发的实际经验,结合项目实际情况,在取 P 值时,应该考虑一个人每天在工作时间内不仅要从事软件工作,还有一些其它工作,如开会、讨论、学习以及与硬件部分协商等活动。纯从事软件工作的有效时间约为 3 小时/天。考虑上述因素,取 $P \approx 0.0856$ 功能点/人天。由式(3)计算相机控制软件的工作量 E 为 1285 人天。

根据嵌入式软件要求高可靠性的特点,将相机控制软件项目各阶段的比例和工作量折算如表 3 所示。

表 3 各阶段工作量分配表

阶段	工作量分配比(%)	工作量 E(人天)
任务分析	10	128.5
需求分析	25	321.5
设计	20	257
编码和单元测试	12	154
组装与测试	33	424
总计	100	1285

3.2.2 成本估计

结合嵌入式软件项目的具体情况,将成本分为 5 类(其中前 3 类成本属软件成本):

(1)开发与支撑环境成本:包括含维护在内的硬件和软件费用、应用的方法和过程所形成的一部分成本以及开发软件项目使用的工具。

(2)人员成本:包括支付给软件开发人员的费用、差旅费、人员的培训费用。

(3)资源消耗成本:包括办公场所使用成本、水、电、空调等费用,会计、文秘等辅助人员的费用,网络、通信以及资料等费用。

(4)风险成本:占总软件成本的 5%~10%,包括诸如人员变动、项目开发周期变动等风险的费用。

(5)不可预见成本:占总软件成本的 5%~10%,用来应对事先没有预料到的工作开销。

按照上述分类方法,将相机控制软件项目的各项软件成本的成本率 C_i ($i=1,2,3$)估算如表 4 所示。则总软件成本率 $C = \sum C_i = 1450$ RMB/人天。

成本的计算公式为:

$$I = E \times C \quad (4)$$

其中 E 为工作量, C 为成本率。则:

$$I_{\text{软件成本}} = E \times C = 1285 \times 1450 = 1863250 \text{ RMB}$$

$$I_{\text{风险成本}} = I_{\text{软件成本}} \times 8\% = 149060 \text{ RMB}$$

$$I_{\text{不可预见成本}} = I_{\text{软件成本}} \times 10\% = 186325 \text{ RMB}$$

各项成本估算如表 4 所示。

表 4 成本估算表

成本名称	成本率 C_i (RMB/人天)	成本 $I = E \times C_i$ (RMB)
开发与支撑	500	642500
环境成本	—	—
人员成本	500	642500
资源消耗成本	450	578250
风险成本	—	149060
不可预计成本	—	186325
总计	—	2198635

3.3 关键计算机资源估计

项目本身存在的计算机资源:相机控制器的微处理器选用 8086,硬件配置主要有引导程序存储器 PROM、应用程序存储器 EEPROM、数据存储器 SRAM、中断控制器、看门狗定时器、时间片定时器、总线控制器、A/D 转换器、电子开关、通用异步串行控制器、秒脉冲选通器、行信号选通器、各种控制锁存器等。其中比较关键的计算机资源限制包括相机存储器容量、处理器能力和通信通道的容量。

3.4 进度估计

进度估计时需要考虑人员技能,各阶段人员分配,对客户承诺的进度等多种因素,因此可以采用如下公式计算项目各阶段的进度^[4]:

$$T = E \times (1 - Q) / n \quad (5)$$

其中 T 为进度, E 为工作量, n 为分配人数, Q 为人员技能综合因素权值(由人员技能综合因素的各项和除以 100 得到,其中人员技能综合因素的确定参见表 5^[4])。

表 5 人员技能综合因素检查表

检查项	权值	权值标准说明
人\组编码熟练程度	-40~40	由低到高(低:新手;高:高手)
人\组文档编写熟练程度	-30~30	由低到高(低:新手;高:高手)
人\组接受工作任务的程度	-10~10	由消极抵制到主动积极
人\组团队协作程度	-15~15	由各自为政到全员配合
人\组团队规模	-5~5	由多人到一人(多人:人员超过 4 个)

设定相机控制软件项目组成员共 8 人,其中有丰富经验者 3 人(A_1, A_2, A_3),有些经验者 3 人(B_1, B_2, B_3),有点经验者 2 人(C_1, C_2)。先依据表 5 为每一成员取相应权值,再将参与工作的每个人的权值相加取平均即为每一检查项的权值,最后将权值取和即为项目人员技能综合因素。设项目的任务分析阶段的工作由 A_1 完成,即 $n_1 = 1$,项目人员的综合因素估计为 75;需求分析阶段的工作由 A_2 和 B_2 共同完成,即 $n_2 = 2$,项目人员的综合因素估计为 50;设计阶段的工作由 A_3, B_1 和 B_3 共同完成,即 $n_3 = 3$,项目人员的综合因素估计为 40;编码和单元测试阶段的工作由 A_2, B_2 和 C_1 共同完成,即 $n_4 = 3$,项目人员的综合因素估计为 45(此时 A_2, B_2 的技能综合因素有所提高);组装与测试这三个阶段的工作由 A_1, B_1 和 C_2 共同完成,即 $n_5 = 3$,项目人员的综合因素估计为 45(此时 A_1, B_1 的技能综合因素有所提高)。因此分别对应的人员技能综合因素权值 Q 为:

$$Q_1 = 75/100 = 0.75; Q_2 = 50/100 = 0.50;$$

$$Q_3 = 40/100 = 0.40; Q_4 = 45/100 = 0.45;$$

$$Q_5 = 45/100 = 0.45.$$

则各阶段工作进度计算如下:

$$T_{\text{任务分析}} = E_{\text{任务分析}} \times (1 - Q_1) / n_1 \approx 32.5 \text{ 天}$$

$$T_{\text{需求分析}} = E_{\text{需求分析}} \times (1 - Q_2) / n_2 \approx 80.5 \text{ 天}$$

(下转第 275 页)

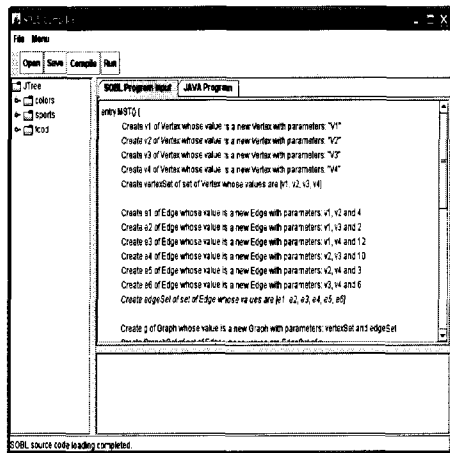


图5 SOBL 程序执行界面

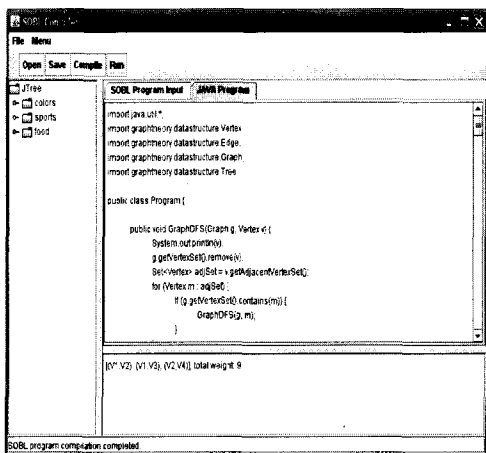


图6 SOBL 程序编译后得出结果

结束语 本文通过以 SemanticObjectsTM为核心技术,并在此基础上研究出一种类自然语言即 SOBL(语义对象行为

语言),来为自然语言和机器进行交互所面临的巨大鸿沟搭起一座桥梁。本文详细介绍了 SOBL 语言的语法结构,阐述了 SOBL 语言的核心技术及它的软件开发方法,并且通过实例验证了 SOBL 语言运行在 SemanticObjectsTM框架中的有效性。结果表明 SOBL 语言是一种较好的面向非专业程序员的类自然语言开发语言,能够大大地简化软件开发的难度。

进一步的工作,主要研究 SOBL 语言作为一种可信编程语言在可信语义软件工程中的具体应用。研究 SOBL 语言在语义服务中作为一种语义服务语言的具体应用等等。

参考文献

- [1] Semantic Objects Getting Started, Copyright 2003. NEC Soft, Co. Ltd.
- [2] Bolognesi T, Brinksma E. Introduction to the ISO Specification Language LOTOS. Computer Networks and ISDN Systems, 1987, 14 (1):25-59
- [3] Drayton L, Chetwynd A G, Blair G S. An Introduction to LOTOS through a Worked Example. Computer Communications (Special Issue on FDTs in Communications and Distributed Systems) Butterworth-Heinemann, 1992, 15(2):70-85
- [4] Ryan M D, Huth M. Logic in Computer Science: Modelling and Reasoning about Systems. ISBN: 0521656028, Cambridge University Press, 2000; 387
- [5] Garcez H E A, Rosenstiel W. CVF-Coverification Framework // Proc. Brazilian Symposium on Integrated Circuit Design. 1998; 103-106
- [6] Strehl K, Thiele L. Symbolic Model Checking of Process Networks Using Interval Diagrams Techniques // Proc. IC-CAD. 1998; 686-692
- [7] Balarin F, Hsieh H, Jurecska A, et al. Formal Verification of Embedded Systems based on CFSM Networks // Proc. DAC. 1996; 568-571
- [8] <http://www.sklse.org/en/en/research1.htm>

(上接第 238 页)

$$T_{\text{设计}} = E_{\text{设计}} \times (1 - Q_3) / n_3 \approx 51.5 \text{ 天}$$

$$T_{\text{编码和单元测试}} = E_{\text{编码和单元测试}} \times (1 - Q_4) / n_4 \approx 28.5 \text{ 天}$$

$$T_{\text{组装与测试}} = E_{\text{组装与测试}} \times (1 - Q_5) / n_5 \approx 78 \text{ 天}$$

最后,根据估算的项目总工期及客户的要求,制定项目进度计划如表 6 所示。

表 6 各阶段、任务的进度计划表

任务名称	持续时间(天)	开始时间	结束时间
任务分析	32.5		
需求分析	80.5		
设计	51.5		
编码和单元测试	28.5		
组装与测试	78		

注:表中所示的时间为有效工作时间。

结束语 本文给出了一种嵌入式软件项目估计方法,为有效地规划和管理嵌入式软件项目、制定合理可行的嵌入式软件开发计划、提高嵌入式软件项目的成功率提供了有力的支持。并结合具体实例对规模估计、工作量/成本估计、关键计算机资源估计和进度估计的内容进行了细致的阐述。

将利用本文方法进行估计的结果,与以往相类似软件实

际结果比较,各项数值的误差均在-15%至+15%之间,并且随着信息量的增加以及不断的细化,估计的误差会逐渐变小,确认了上述方法的实际效用。亚里斯多德曾说:“应该满足于事物的本性所能容许的精确度,当只能近似于真理时,不要去寻求绝对的准确”。估计的基本目标就是产生一个能够在大部分时间内发挥作用的合理的估计。另一方面,计划基于估计,没有估计而制定的开发计划是盲目的计划。因此,我们应尽早估计,经常估计。

参考文献

- [1] Paulk M C, Weber C V, et al. Key Practices of the Capability Maturity Model. SM, Version 1.1. Technical Report, CMU/SEI-93-TR-025, ESC-TR-93-178. 1993
- [2] Pressman R S. Software Engineering A Practitioner's Approach. Fifth Edition. McGraw-Hill, 2001
- [3] Jalote P. CMM in Practice: Processes for Executing Software Projects at Infosys. Pearson Education, Inc., 1999
- [4] 田媛,等. 长春光学精密机械与物理研究所 GJB 5000 过程文件. 中国科学院长春光学精密机械与物理研究所, 2006
- [5] 周汉兵,关昕,马力. 功能点度量在软件开发中的应用. 计算机工程与设计, 2006, 27(3):525-527,529