

A Novel Activity-Based Costing for Modern Software Development

—An Implementation of Cloud-Based Application

Rich Lee^{1,2}

¹IBM, Taipei

²National Taipei University of Technology, Taipei

Email: richchihlee@gmail.com

Received: Aug. 28th, 2014; revised: Sep. 10th, 2014; accepted: Sep. 17th, 2014

Copyright © 2014 by author and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

There are quite a few approaches to software development. The IT staffs usually govern and manage the tasks of software development project. Even though, the management accountant could understand the detail and the nature of software development; it is still not easy, without embedding a scheme in among the development processes, to suggest a more proper budget plan before the development begins, and to be able to remind the IT staffs any unusual costly activities during the development processes. This article proposes a series of processes co-existing with the current daily practice of software engineering. This scheme not just can help the IT staff to manage the task management well, but also can let the management accountant be aware of the imposed costs of the development activities by applying the Activity-Based Costing, and feedback professional comments to the IT staffs and the key executives. Based upon these comments, they can take appropriate measures to mitigate the potential project risks sooner.

Keywords

Management Accounting, Activity-Based Costing, Software Development Lifecycle, Cost Estimation, Cloud Computing

基于作业成本分析法之新颖式软件成本计算

—应用于云计算软件系统

李 智^{1,2}

¹IBM, 台北市

²国立台北科技大学, 台北市

Email: richchihlee@gmail.com

收稿日期: 2014年8月28日; 修回日期: 2014年9月10日; 录用日期: 2014年9月17日

摘要

软件开发工法各异, 项目管理工作亦常受信息专业人员所专擅主导, 纵使管理会计人员了解软件开发工法细节, 但倘无一套作业机制运作于软件开发过程中, 也不易针对成本事前之估算与过程中各项工作支出之合理性, 提出专业意见。本文设计一套符合当前软件工程项目开发管理工作日常作业机制, 一方面满足信息专业人员对项目管理之要求, 另一方面结合作业成本分析法, 使得管理会计人员能随时了解各项工作支出情形, 反馈专业意见给信息专业人员与关键主管阶层, 适时找出相应对策尽早解决软件开发潜在风险。

关键词

管理会计, 作业成本分析法, 软件开发生命周期, 成本估算, 云计算

1. 引言

软件开发成本估算有许多先天信息不足与估算障碍影响其精确度。软件开发可粗分为两类: 1) 软件产出物为一商品, 有清楚产品规格, 软件开发为该产品之生产制程, 如图 1 所示, 市场对你开发软件商品有一定需求, “产品经理”依据产品定位探索市场需求, 提出产品功能需求建议, 供“设计经理”评估其可行性, 经充分讨论后再具体规划细部功能规格, 由“研发经理”按功能规格加以产品化, 最后“销售经理”将产品推销至市场, 再由“产品经理”探求用户及潜在用户对产品之期许与建议, 做为下一版产品提升之依据, 倘若市场对产品需求愈多或是修正愈速, 则产品变动愈大致使开发工作亦愈多; 及 2) 原因于一项目要求, 项目中之“利益关系人”(Stakeholders)因应商务需求, 主导软件产品规格, 软件开发为其满足商务需求之生产制程, 如图 2 所示, “利益关系人”于项目初期仅有概略需求, “系统分析员”透过访谈分析探求“利益关系人”显性及隐性真实具体需求[1], 发展需求规格书, 并与“系统设计员”充分讨论, 由“系统设计员”具体规划细部系统功能规格, 交由“研发经理”按功能规格设计制作产出物, 再由“利益关系人”测试, 当产出物与需求认知差异时, 则衍生修正需求再续前述流程, 直至认知差异一致为止, 倘若修正需求始终无法收敛一致, 项目开发必然升高成本与增加失败风险。软件商品开发虽有清楚产品规格, 但常因竞争对手商品规格变化, 或因使用者要求而克制化, 致使需求不定; 而软件项目开发虽有概略需求范围, 但因开发期间外在商务环境改变, 或因利益关系人主观意识, 亦致使需求不定, 也正因此致使软件开发估算困难, 同时在开发过程中发生多余设计活动以弥补需求不定所造成之重工(Reworks), 重工便产生估算非预期成本。

云计算在国际信息服务大厂(如: Google, Amazon, IBM, Oracle, Microsoft 等)及各式新颖应用服务(如: Facebook, Twitter, Salesforce, Dropbox 等)推波助澜下正方兴未艾, 简分三种模式: 1) 基础架构层(Infrastructure-as-a-Service, IaaS)——即“基础架构即服务”, 是以系统资源虚拟化(Resource Virtualization)为手段, 将运算、储存和网路等资源抽象化, 实现系统部署流程自动化和资源管理优化, 透过种种动态灵活基础架构管理机制与服务, 大幅降低系统资源妥备成本; 2) 平台层(Platform-as-a-Service, PaaS)——即

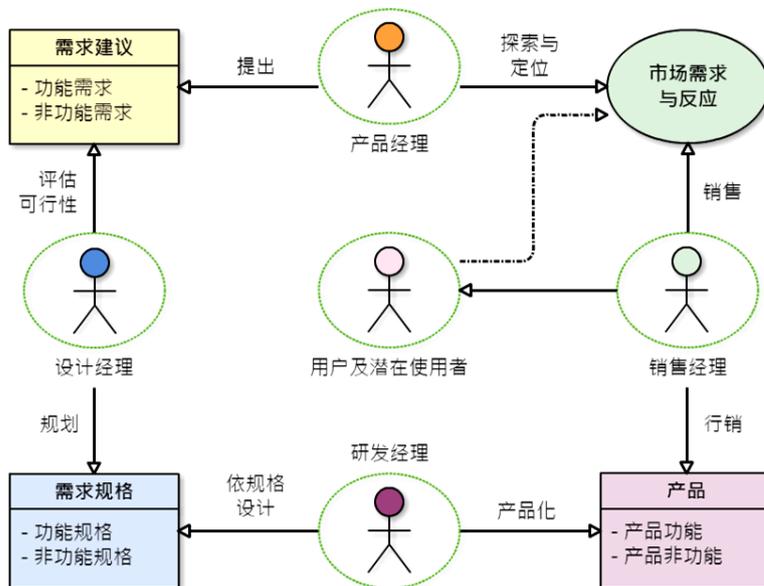


Figure 1. Software production workflow
图 1. 软件产品化流程

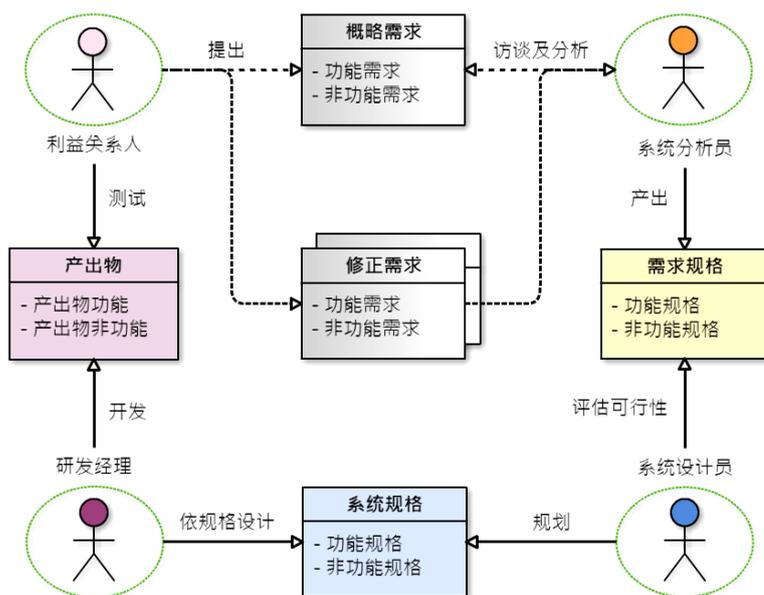


Figure 2. Software development project workflow
图 2. 软件项目开发流程

“平台即服务”，为云计算服务应用软件提供了开发、运行、管理与监控环境，使设计者仅需关注于服务设计，透过平台商提供之“程序整合开发工具” (Integrated Development Environment, IDE)将软件建构于此平台之上，运用平台层所提供之各项优化“中间件服务” (Middleware Services)，以提高服务应用之扩充性、可用性与安全性等能力；及 3) 应用层(Software-as-a-Service, SaaS)——即“软体即服务”，用户透过各式网络中端设备直接操作应用层所提供之商务相关软体组合，此类应用均已架构于基础架构层提供之各种资源，以及平台层提供之优质运行环境上，根据实际使用程度与所点选之服务加以付费等。由前述可知，使用基础架构层服务之软件开发，目前居于多数应用，仍需要布署其他运行所需之应用服务

器、数据库，与支持软件等；使用平台层服务之软件开发，虽无需布署运行支持软件，但必须调校监控管理平台所提供之各项资源；而仅使用应用层者则无关软件开发，非本文所探讨范围。云计算之软件开发所使用之编程语言与设计框架均属稳定，并不随应用服务差异而改变，因此相应设计相关活动单位产值与所耗用时间比亦应变异不大，适合以活动记录为核心之管理方式，以提高工作绩效与降低潜在风险，同时活动历史资料可进一步了解开发团队产值变化情形，将有助于未来软件估算更为准确。

软件开发过程是一系列活动所组成，包涵：1) 需求规画与分析；2) 系统分析与设计；3) 测试规画与实施；4) 系统上线与培训；及 5) 系统运维与效能监控等五大活动类型。而常被用以降低或消除价值生成(Value Creation)作业所需时间与资源之“作业基础管理”(Activity-Based Management, ABM)亦为一系列成本记录与控制活动所组成，搭配“作业成本分析法”(Activity-Based Costing, ABC)用以解析活动所产生之真实发生成本，并便于衡量与制订各项作业活动细项之积效指标。本文提出一套作业机制将软件开发过程与“作业基础管理”相结合，透过明确定义软件开发过程中之“成本驱动因子”(Cost Drivers)，配合“作业基础管理”信息系统便于搜集成本记录，提供短期进行中项目各“成本驱动因子”已发生成本与事前估算之差异分析，并提供软件开发中长期成本趋势与变异分析。

2. 软件需求本质

需求不定与难以捕捉纪录是软件开发不确定因素源之一，也是成本估算难以准确主因，软件需求包括两类：1) 功能性需求(Functional Requirements)——指软件输出及相关操作流程与数据处理等；及 2) 非功能性需求(Nonfunctional Requirements)——指软件运行效能、系统反应时间，安全性与使用性(Usability)等[2]。软件需求书为软件开发所需之规格，其完整性与正确性将直接影响软件开发时程与质量，“研发经理”倚赖完整正确需求规格，方能制定后续实施计划，因此软件需求规格必须具备两大类属性，其中“内部属性”包括：1) 语意明确性(Unambiguous)，2) 正确性(Correct)，3) 完整性(Complete)，4) 可理解性(Understandable)，5) 可验证性(Verifiable)，6) 内部一致性(Internal Consistent)，7) 可修正性(Modifiable)，8) 标示相对重要性(Annotated by Relative Importance)，9) 标示相对确定性(Annotated by Relative Stability)，10) 标示版本(Annotated by Version)，11) 精确性(Precise)，12) 版次间追溯性(Traced)，及 13) 可追踪性(Traceable)等；而“外部属性”则包括：1) 可获取性(Achievable)，2) 电子储存(Electronically Stored)，3) 简明性(Concise)，4) 设计无关性(Design Independent)，及 5) 再利用性(Reusable)等。由此可见，正确地记录软件需求并不容易，即使针对相同事务不同人或甚至是于另一时间点均有可能发生解读偏差，因此造成软件开发过程中发生重工机率极高之主因。

同时，整个软件开发乃是一个“不同人之间”(Human-to-Human)思考沟通与对问题之了解及“人与计算器间”(Human-to-Machine)转换与解译之过程，软件品质就受两种因素所制约；甚至对同一问题之理解也因时间不同而相异，这尚未包括因无法掌握之外部因素影响所强加之需求变化。另外，软件于设计之初，由于商业规模不明，因此对未来系统运行模式与来自用户应力(User Stress)预估必须有所假定，依据此假定系统定义其强度(Design Strength)以设计整体架构，使得在运行期间设计强度始终大于用户应力；假使错估用户应力或设计强度，均将影响软件开发活动与成本未预期之增加。因此有必要进一步分析与了解致使软件开发成本偏高原因，并据以修正下一个软件项目成本估算与潜在风险。

由此可见，在软件开发早期时期，概略需求受外部因素干扰十分不稳定，欲充分纪录软件开发所需规格，并兼顾前述需求规格属性，根本不切实际，而欲据此精确成本估算更是缘木求鱼；同时整个软件开发过程为一系列活动所构成[3]，活动类型包括：1) 与外部沟通，2) 建立内部共识，3) 需求规划，4) 系统设计，5) 功能测试与验证，6) 系统导入，7) 系统维护，及 8) 撰写文件等活动；假使在软件开发过程中不将各种活动成本计入，不但会低估软件开发真实成本，也无从得知何种活动成本异常。

3. 软件开发成本估算

在软件工程(Software Engineering)领域中亦有探讨软件开发成本估算议题,除了依过去经验或是专家评估外,其他估算方法尚有:1) 功能点(Function Points)估算法——从“利益关系人”概略需求为观点,依照软件项目类型,如:软件项目开发、系统增修项目开发,或是软件产品项目开发等,决定计点方式与权重,评估项目开发数据与交易功能复杂度与实施范围,套入功能点数依计算公式再乘以每功能点数之单位开发成本,即取得项目成本估算[4];及2) 构造性成本模型(Constructive Cost Model, COCOMO)——依照软件项目产出物特性,运算环境特性,人员及项目属性等,订出不同成本因子,在项目发展不同进程中,如:需求及初部设计、细部设计、程序设计与单元测试、及整合测试等分别套用公式求出所需人月数,再乘以单位人月开发成本,即取得项目成本估算[5]。上述软件项目开发估算方法,乃至后续项目实施成本及费用之发生,均与会计程序脱钩,以致于会计部门难以管理会计(Management Accounting)制度,综观项目绩效及发现潜在问题,发挥项目预警及监督之能[6]。运用“作业成本分析法”在软件开发生命周期不同进程中,将软件开发作业水平与各项成本变动原因纳入考虑,揭露开发活动隐性成本,并为各类活动订出绩效指标,进而指出绩效关键活动,透过有价值成本信息,提供具体软件开发可改善之建议[7]。本文探讨运用“作业成本分析法”,并提出具体做法,将软件开发活动成本计入,一方面兼顾软件工程项目管理需求,另一方面将软件开发纳入日常管理会计正轨,同时做为日后类似开发成本估算依据,并在项目实施中早期发现成本趋高或发生变异之活动,提醒项目开发团队及早因应。

4. 软件开发成本驱动因子

软件开发运用“作业成本分析法”是否成功之关键因素受四个面向影响:1) 组织行为模式如:高阶主管是否支持,是否与经营策略间连动,绩效评估与奖惩制度是否合理,权责是否清楚,资源是否充足,项目成员对“作业成本分析法”训练是否到位,及对成本管理目标是否具有共识等;2) 实施技术性如:是否具备软发生命周期相关知识与经验,是否有软件协助成本作业,及是否能有效收集“成本驱动因子”等;3) 开发组织结构:软件开发目的在于产品取得市场竞争,或产出物能满足“利益关系人”期望,软件开发组织结构是否够敏捷能反应需求变化,及部门间运作是否效率等;4) 组织文化如:组织是否为成果与团队合作导向并关注软件开发细节,成本作业参与者态度是否正面,及认知成本作业是否为软件开发管理带来效益等[8]。因此综合以上成功关键因素,软件开发欲运用“作业成本分析法”,则必须考虑软件工程现行实务变动不大下,软件开发主管及其团队理解“作业成本分析法”所能带来之管理效益,使软发生命周期自然无缝纳入标准成本会计制度,同时也让会计人员能提醒项目开发潜在风险。

衍生自软件图 1,产品化及图 2,项目开发流程,进而延伸导入相应之作业成本分析法如图 3 及图 4 所示,“填写项目活动纪录表”为导入作业成本分析法流程之核心活动,软发生命周期所有参与者均须填写,而此核心活动与目前软件开发模式,包括:计划导向(Plan-Driven)及敏捷程序(Agile Process),并不违背,仅需纳入目前工作核销及回报作业即可[9]。

“项目活动纪录表”范例如表 1 所示,填表人可依其在项目中角色,于会计结账许可日之前,每日或每周填写一次,纪录表活动记录包括(4-1~4-7)所有字段:1) 项目代码——指软件开发项目内部所使用之简称;2) 预算科目代码——指此软件开发项目相应之会计预算科目代码,所有成本发生均归于此科目项下;3) 员工代码——指活动发生者,即填表人之代码;4-1) 活动类码——指该活动在软发生命周期类型;4-2) 活动子类码——指活动在该类型下之细分类型;4-3/4) 日期/时间起止——指该活动开始即结束时间;4-5) 工作内容——纪录活动过程中关键事物或是其他文件指引信息等,不必然巨细靡遗,足够提示主管即可;4-6) 工作产出——指活动后产出物,如:访谈纪录、程序代码,或测试结果均是;4-7) 主管签核——活动主管逐项审视纪录是否记载正确,特别是活动字段(4-1~4-4)将直接影响项目成本计算

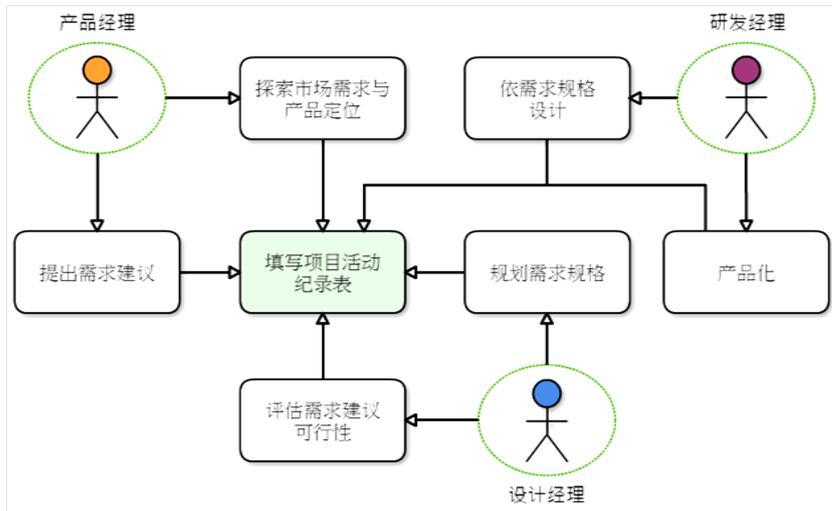


Figure 3. Software production workflow with ABC

图 3. 软件产品化导入作业成本分析法流程

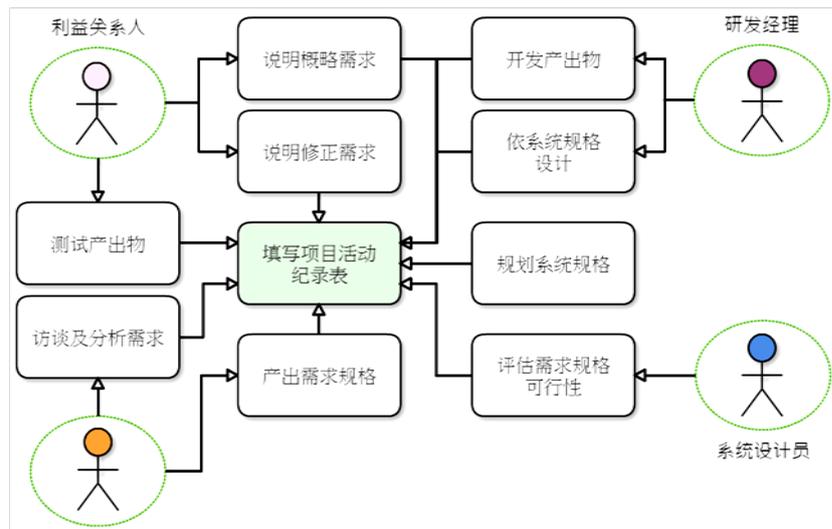


Figure 4. Software development workflow with ABC

图 4. 软件项目开发导入作业成本分析法流程

Table 1. Task activity records sample form

表 1. 项目活动记录表范例

项目代码	(1)	预算科目代码	(2)	员工代码	(3)
活动	日期/时间		工作内容	工作产出	主管签核
类码	子类码	(起)	(止)		
(4-1)	(4-2)	(4-3)	(4-4)	(4-5)	(4-6)
入账日期	(5-1)	会计	(5-2)	稽核	(5-3)

公式；5-1) 入账日期——由会计及稽核人员填写，表示此记录表所有活动数据均已经成本部门确认；5-2) 会计员工代码；及 5-3) 稽核员工代码。

“项目活动记录表”范例提供软件开发活动成本记录格式，使用者可依其软件类型与各阶段开发生

命周期所需，加以细化为各式活动纪录表，使其用途更明确减少使用时发生疑义或是降低导入此机制之培训成本。为方便随处使用与正确及时记录，信息人员更可将各式活动纪录表以行动运算(Mobile Computing)形式撰写成应用程序，以扩大多阶层相关人员共同参与，以减轻因使用“作业成本分析法”所带来过多信息与复杂性。管理会计人员可依此“项目活动纪录表”制作传票，展开各开发项目之成本分析，揭露各“成本驱动因子”所发生之变异，及点出潜在项目实施知各项风险。同时藉由软件估算值与实际发生成本间差异，持续地调整各项权重，进而使未来件估算更为科学化与准确化。

5. 个案研究

本文观察与追踪已导入所建议“作业成本分析法”之某云计算软件开发商，分析五个指标型大型开发项目与二个自行开发软件产品之“项目活动纪录表”资料，并透过“焦点访谈”(Focus Group)方式了解导入“作业基础管理”过程中之各方得失意见，整理后简说获致经验如后：1) 在实施“作业成本分析法”前，软件开发成本估算仍采取软件工程合适之估算方法，但依照“功能性需求”与“非功能性需求”展开各项“成本驱动因子”；2) 依“成本驱动因子”设计“项目活动纪录表”，经过培训后由信息人员填写；3) 透过信息系统整理“项目活动纪录表”资料，由管理会计人员审核与后续分析；4) 在软件开发估算之初，依照商务需求设定“功能性需求”与“非功能性需求”间预算之比例，此安排用意在于控制信息人员投入需求设计适配资源，假使于项目结束后统计实际发生成本，调整下一个类似软件项目之比例；5) 管理会计人员每周产出软件项目各项“成本驱动因子”统计数字外，亦揭露其他软件项目工作相近但却成本异常之执行人与该工作差异，由相关信息人员说明；6) 项目结束后依照“成本驱动因子”统计数字推演简易软件开发估算公式如下，其中 i 为软件项目类型， ω_θ 为“成本驱动因子”权重：

$$\text{Estimation}_i = \alpha * \text{FuncReq}_i + \beta * \text{NonFuncReq}_i \quad \alpha, \beta \text{ 为权重, 且 } \alpha + \beta = 1 \quad (1)$$

$$\text{FuncReq}_i = \sum_{\theta=1}^m (\omega_\theta * \text{CostDriver}_{\theta,i}) \quad \text{NonFuncReq}_i = \sum_{\theta=1}^n (\omega_\theta * \text{CostDriver}_{\theta,i}) \quad (2)$$

7) 在搜集各 CostDriver_θ 资料之后，计算其平均数与变异数如下，找出变异数大之“成本驱动因子”，由信息人员拟定对策如何降低工作变异，通常此类变异产生之原因为“成本驱动因子”设计粒度过粗、信息人员“项目活动纪录表”培训不足，或是某信息人员执行工作效率发生变异等：

$$\text{AvgCostDriver}_\theta = \sum_{i=1}^q (\text{CostDriver}_{\theta,i}) / n \quad (\text{平均数}) \quad (3)$$

$$\text{VarCostDriver}_\theta = \sum_{i=1}^q (\text{CostDriver}_{\theta,i} - \text{AvgCostDriver}_\theta)^2 / n \quad (\text{变异数}) \quad (4)$$

及 8) 依照“功能性”与“非功能性”需求实际发生成本，及软件项目特性指定或依 $\text{AvgCostDriver}_\theta$ 求算各“成本驱动因子”权重 ω_θ ，此权重之用意在于强调一般软件项目成本对某些“成本驱动因子”敏感，并能及早揭露潜在风险所设计。由于本文所建议之方法属于依时间滚动方式，该软件开发商随时可依项目状况而机动调整，使得软件开发估算愈来愈稳定，即便少数信息人员因流动而造成之变异，也因为其他人员对于“作业基础管理”逐渐熟稔，透过培训及同仁相互提醒而迅速适应。

个案中依云计算软件开发各生命周期设计多种“项目活动纪录表”，包括：1) 需求范围界定活动纪录表——用于搜集软件竞争产品信息，进行市场调查，与探索用户期望功能等成本计算；2) 关键技术研发活动纪录表——用于研发软件新技术，开发共享型关键组件，与软件新架构规画测试验证等成本计算；3) 培训活动纪录表——用于内外部培训成本计算，举凡与软件开发相关之专业知识、新颖技术，与项目管理等培训活动；4) 需求分析活动纪录表——用于搜集、规划、记录、分析，疑义澄清与访谈各种“功能性”与“非功能性”需求之活动所发生之各项成本计算；5) 系统设计活动纪录表——依“功能性”与

“非功能性”及“功能点”之需求规划，用以搜集、规划、记录、分析，疑义澄清与访谈系统设计相关之活动所发生之各项成本计算；6) 程序设计活动纪录表——依系统设计规划，用于程序设计相关活动如：程序撰写、各种软件测试，疑义澄清与访谈等所发生之各项成本计算；7) 数据库活动纪录表——用于规划、记录、分析，设计数据库储存取用机制，如：建立与维护数据表格、物件关联对映(Object-Relational Mapping, ORM)，与数据库效能调校等所发生之各项成本计算；8) 系统布署活动纪录表——用于规划、记录、分析，与设置系统运行所需环境之相关活动如：安装与调校虚拟机(Virtual Machine, VM)、应用系统服务器、数据库、其他支援软件、网络与其他硬件设备，及布署新软件产品等所发生之各项成本计算；9) 新系统导入辅导活动纪录表——用于规划、记录、疑义澄清与访谈，及培训辅导新软件产品使用之相关活动所发生之各项成本计算；10) 系统运维活动纪录表——用于新软件产品上线导入与辅导后，举凡后续需求增修调整、相应系统程序与数据库设计，系统布署与导入辅导等相关活动所发生之各项成本计算。

6. 结论

软件开发估算之困难在于开发过程中未预期活动可能性高，而此未预期活动又以因需求不定之影响最巨，软件工程中所以建议知估算方法均与成本会计程序脱钩，使得管理会计人员无法适时提出专业观点，提醒或分析之异常开发活动，为此本文运用“作业成本分析法”与软件开发流程相结合，透过完整成本发生记录，加以分析何种开发活动最容易发生成本变异，更进一步地可推估类似软件开发之成本，尤其重要的是使信息人员了解“作业成本分析法”要义与其效益，同时也使管理会计人员了解软件开发过程，俾能适时提醒信息人员采取相应修正作为以降低软件开发之潜在风险。本文着重于系统功能需求活动之成本记录，目的在于掌握及早期暴露因需求不定(包括需求不确定性、多样性，与人员解读需求能力等)所致之软件开发成本变异，至于后续各项软件设计布署与导入活动，依据软件开发项目绩效原因分析相关文献[10] [11]及个案实证发现，在需求明确后其成本变异不大，并在软件开发过程中均可控制。同时本文虽以“作业成本分析法”记录各项活动发生成本，然其背后用意则在于尝试改良软件成本估算变异太大问题，并兼顾现行软件工程方法，提出整合成本会计理论提出具体可行之软件开发成本机制。

参考文献 (References)

- [1] Mohamed, A.H. (2010) Facilitating tacit-knowledge acquisition within requirements engineering. *Proceedings of the 10th WSEAS International Conference on Applied Computer Science*, Stevens Point.
- [2] Summers, J.D., Joshi, S. and Morkos, B. (2014) Requirements evolution: Relating functional and non-functional requirement change on student project success. *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Buffalo.
- [3] Highsmith III, J.A. (2000) Adaptive software development: A collaborative approach to managing complex systems. Dorset House Publishing, 392.
- [4] Balaji, N., Shivakumar, N. and Ananth, V.V. (2013) Software cost estimation using function point with non algorithmic approach. *Global Journal of Computer Science and Technology*, **13**.
- [5] Sharma, T. (2011) Analysis of software cost estimation using COCOMO II. *International Journal of Scientific & Engineering Research*, **2**, 1-5.
- [6] Clinton, B.D. and White, L.R. (2012) Roles and practices in management accounting. *Management Accounting*, **94**, 37-43.
- [7] Kaplan, R.S. and Anderson, S.R. (2007) Time-driven activity-based costing: A simpler and more powerful path to higher profits. Harvard Business School Press, Boston.
- [8] Fei, Z.Y. and Isa, C.R. (2010) Factors influencing activity-based costing success: A research framework. *International Journal of Trade, Economics and Finance*, **1**, 144-150.
- [9] Kakar, A.K. (2012) A theory of software development methodologies. *Proceedings of the Southern Association for Information Systems Conference*, Atlanta.

- [10] Liu, J.Y.-C., Chen, H.-G., Chen, C.C. and Sheu, T.S. (2011) Relationships among interpersonal conflict, requirements uncertainty, and software project performance. *International Journal of Project Management*, **29**, 547-556.
- [11] Letier, E., Stefan, D. and Barr, E.T. (2014) Uncertainty, risk, and information value in software requirements and architecture. *36th International Conference on Software Engineering*, Hyderabad.