

项目名称：点菜宝

文档名称：成本计算过程文档

HUSTZL

侯皓斐 软件2003 班 U202010851

版本：V1.0

文档信息

文件状态： <input type="checkbox"/> 草稿 <input checked="" type="checkbox"/> 正式发布 <input type="checkbox"/> 正在修改	文件标识：	HUSTZL_001
	当前版本：	V1.0
	作者：	侯皓斐
	发布日期：	< 2023.4.23 >

文档更改记录

版本	更改日期	更改人	更改原因	说明
V1.0	2023.4.16	侯皓斐		初版

摘要

本文档旨在为点菜宝系统的成本计算过程提供指导。该系统分为六个子功能：点餐系统、订单管理、餐桌状态管理、后台管理系统、支付系统和保障系统。我们将使用各种方法来分析每个子功能的人天数，以便在自下而上的估算中准确地计算出整个系统的成本。最终，我们将列出一个大表，列出每个子功能的人天数，以及总成本的估算结果。通过本文档，我们将为开发团队提供一个清晰的成本估算框架，以确保项目能够按时、按预算完成。

目录

- 1.引言..... 1
 - 1.1 项目背景..... 1
 - 1.2 目的和范围..... 1
- 2.点菜宝任务分解概述..... 1
- 3.成本估算方法..... 1
 - 3.1 代码行估算法（Code Line Estimation）2
 - 3.2Albrecht 功能点估算（Function Point Analysis）2
 - 3.3 用例点估算方法（Use Case Point Estimation）2
 - 3.4 类比估算（Analogy Estimation）2
 - 3.5 三点估算（Three-point Estimation）2
 - 3.6 专家估算（Expert Estimation）3
- 4.子功能成本估算..... 3
 - 4.1 点餐系统..... 3
 - 4.2 订单系统..... 4
 - 4.3 餐桌状态管理..... 5
 - 4.4 支付系统..... 6
 - 4.5 后台管理系统..... 7
 - 4.6 保障系统..... 8
- 5.自下而上工作量估算..... 9
- 6.结论..... 12

1. 引言

1.1 项目背景

随着餐饮行业的不断发展，消费者对于餐饮服务的需求日益多样化，为了满足消费者的需求并提高餐厅的运营效率，餐饮企业纷纷转向数字化技术以改进服务流程。点菜宝系统作为一种餐饮业数字化解决方案，旨在提供一个一体化的平台，实现从点餐、订单管理到支付等环节的无缝对接，为客户提供更加便捷的用餐体验，同时帮助餐厅提高运营效率和盈利能力。

1.2 目的和范围

本文档的目的是为点菜宝系统的成本计算提供一个详细的过程文档，以便于项目相关人员对项目成本有一个清晰的了解。本文档将对点菜宝系统的各个子功能进行成本估算，包括点餐系统、订单管理、餐桌状态管理、后台管理系统、支付系统和保障系统。

范围涵盖了点菜宝系统的整个开发周期，包括设计、开发、测试、部署和维护等阶段。本文档将采用多种成本估算方法，分析每个子功能所需的人天数，并在文档最后列出一个自下而上的成本估算大表。

2. 点菜宝任务分解概述

点菜宝系统的任务分解主要包括以下六个子系统，分别负责不同的功能模块，包括点餐系统、订单管理、餐桌状态管理、后台管理系统、支付系统和保障系统。以实现整个点菜宝系统的无缝运作。详情请看 WBS 分解图。

3. 成本估算方法

在本节中，我们将简要介绍六种成本估算方法，以评估点菜宝系统各个子功能的人天数。这些方法包括代码行估算法，Albrecht 功能点估算，用例点估算方法，类比估算，三点估算和专家估算。

接下来，我们将根据上述方法，为点菜宝系统的六个部分进行自下而上的成本估算，最后列出一个大表。通过对这些数据的分析，我们可以得出一个更准确的点菜宝系统的成本估算。

3.1 代码行估算法（Code Line Estimation）

代码行估算法是通过计算开发所需的代码行数来估算成本的方法。这种方法根据历史项目的经验数据以及每个子功能的具体需求来估算代码行数。为了得到更准确的估算结果，我们需要考虑开发语言、开发环境、开发人员的技能等因素。

3.2 Albrecht 功能点估算（Function Point Analysis）

Albrecht 功能点估算是一种通过评估软件系统的功能复杂性来估算成本的方法。这种方法将功能划分为五个基本类型：外部输入、外部输出、外部查询、内部逻辑文件和外部接口文件。为了得到功能点数，我们需要计算每个类型的数量并根据其复杂程度进行加权求和。最后，根据历史项目的经验数据，我们可以将功能点数转换为人天数。

3.3 用例点估算方法（Use Case Point Estimation）

用例点估算方法通过分析系统的用例来估算成本。首先，我们需要识别系统的所有用例，然后根据用例的复杂性分为简单、中等和复杂三个级别。接着，为每个级别的用户分配权重并计算总的用例点数。根据历史项目的经验数据，我们可以将用例点数转换为人天数。

3.4 类比估算（Analogy Estimation）

类比估算是一种通过比较类似项目的历史数据来估算成本的方法。在这种方法中，我们需要收集与点菜宝系统类似的历史项目数据，然后根据这些数据计算每个子功能的人天数。为了提高估算的准确性，我们可以考虑项目的规模、技术难度、开发人员的技能等因素。

3.5 三点估算（Three-point Estimation）

三点估算是一种根据最乐观、最可能和最悲观的情况来估算成本的方法。对于每个子功能，我们需要估算在最乐观（最短时间），最可能（正常情况下）和最悲观（最长时间）三种情况下的人天数。然后，根据特定的公式计算出最终的估算值。

3.6 专家估算（Expert Estimation）

专家估算是一种依赖于具有丰富经验的专家对项目成本进行估算的方法。在这种方法中，我们需要收集一组具有类似项目经验的专家，并向他们提供关于点餐系统各个子功能的详细信息。专家们将基于他们的经验和知识，对每个子功能的人天数进行估算。

4. 子功能成本估算

4.1 点餐系统

下面我们将使用用例点估算方法对点餐系统进行成本估算：

1. 计算未调整的角色权值 UAW：

用户（User）：权重 = 3

系统管理员（System Administrator）：权重 = 3

$$UAW = 3 + 3 = 6$$

2. 计算未调整的用例权值 UUCW：

使用简单（S）、中等（M）和复杂（C）对每个用例进行分类，然后分别乘以对应的权重（S=5, M=10, C=15）。

- 1.1. 菜单浏览：S (5)
- 1.2. 点餐功能：M (10)
- 1.3. 结账功能：M (10)
- 1.4. 多人点餐功能：C (15)
- 1.5. 特殊要求处理：M (10)

$$UUCW = 5 + 10 + 10 + 15 + 10 = 50$$

3. 计算未调整的用例点 UUCP：

$$UUCP = UAW + UUCW = 6 + 50 = 56$$

4. 计算技术和环境因子 TEF:

评估一系列技术因子 TCF 和环境因子 ECF(如硬件限制、性能要求、团队等), 两者相乘得到 TEF 值。

假设我们的 TEF 值为 1.65, 可以根据具体项目对各因子进行评估。

5. 计算调整的用例点 UCP:

$$UCP = UUCP * TEF = 56 * 1.65 = 92.4$$

6. 计算工作量 (man-hours):

使用一个系数, 将 UCP 转换为工作量。这个系数可能因组织和团队而异。假设每个 UCP 需要 20 个 man-hours, 那么:

$$\text{工作量} = UCP * 20 = 92.4 * 20 = 1848 \text{ man-hours}$$

根据上述计算, 我们估计点菜宝点餐系统的成本为 1848 个 man-hours。

4.2 订单系统

下面我们将使用用例点估算方法对订单系统进行成本估算:

1. 计算未调整的角色权值 UAW (Unadjusted Actor Weight):

根据项目需求, 我们可以确定以下角色: 系统管理员, 顾客, 收银员/服务员。假设这三个角色的权重分别为: 简单 (1), 普通 (2) 和复杂 (3)。那么 $UAW = 1 + 2 + 3 = 6$ 。

2. 计算未调整的用例权值 UUCW (Unadjusted Use Case Weight):

使用简单 (S)、中等 (M) 和复杂 (C) 对每个用例进行分类, 然后分别乘以对应的权重 (S=5, M=10, C=15)。

2.1. 订单打印: S (5)

2.2. 订单查询: M (10)

2.3. 订单修改: H (15)

那么 $UUCW = 5 + 10 + 15 = 30$ 。

3. 计算未调整的用例点 UUCP (Unadjusted Use Case Points):

$$UUCP = UAW + UUCW = 6 + 30 = 36$$

4. 计算技术和环境因子 TEF (Technical and Environmental Factors):

评估一系列技术因子 TCF 和环境因子 ECF（如硬件限制、性能要求、团队等），两者相乘得到 TEF 值。

假设我们的 TEF 值为 1.65，可以根据具体项目对各因子进行评估。

5. 计算调整的用例点 UCP (Use Case Points):

$$UCP = UUCP * TEF = 36 * 1.65 = 59.4$$

6. 计算工作量 (man-hours):

为了将 UCP 转换为工作量，我们需要一个转换系数，通常在 20 到 40 之间。假设我们使用 20 作为转换系数，那么工作量 = $UCP * 20 = 59.4 * 20 = 1188$ man-hours。

4.3 餐桌状态管理

下面我们将使用 Albrecht 功能点估算方法对餐桌状态管理进行成本估算：

1. 计算未调整功能点计数 UFC

我们统计餐桌状态管理子系统的外部输入，输出，查询，接口文件，内部逻辑文件如下。

显示餐桌状态（空闲/占用）：输入功能点（IFP）

餐桌状态更改（如顾客入座、离开）：外部输入（EI）

同步更新餐桌状态信息至系统：外部输出（EO）

状态更新异常处理：外部查询（EQ）

分配服务员至餐桌：外部输入（EI）

显示服务员信息：输入功能点（IFP）

更换服务员：外部输入（EI）

同步服务员信息变动至系统：外部输出（EO）

接下来，我们将根据功能点的数量和复杂度，为每个功能点分配权重。这里我们假设所有功能点的复杂度均为简单，因此每个外部输入（EI）的权重为 3，外部输出（EO）的权重为 4，外部查询（EQ）的权重为 3，输入功能点（IFP）的权重为 3。

计算未调整功能点计数总数：

$$UFC = (3 + 3 + 4 + 3) + (3 + 3 + 3 + 4) = 26$$

2. 计算 TCF 技术复杂度因子：

$$TCF = 0.65 + 0.01(\sum(F_i)) : F_i: 0-5, \quad TCF: 0.65-1.35$$

F_i 为每个通用系统特性的影响程度， i 表示每个通用系统特性，取值 1-14

技术复杂度因子			
F1	可靠的备份和恢复	F2	数据通信
F3	分布式函数	F4	性能
F5	大量使用的配置	F6	联机数据输入
F7	操作简单性	F8	在线升级
F9	复杂界面	F10	复杂数据处理
F11	重复使用性	F12	安装简易性
F13	多重站点	F14	易于修改

我们采用估算 $TCF = 0.65 + 0.01(14 \times 3) = 1.07$

3. 计算工作量 (man-hours)：

如果：PF=20 工时/功能点。

那么最终的工作量 = $UFC * TCF * PF = 26 * 1.07 * 20 = 556.4$ man-hours。

4.4 支付系统

代码行估算方法相对简单易懂，适用性广泛，结果较为准确。而且支付系统成熟，是调用支付宝和微信接口封装的系统，代码行数量非常好估算。假设我们使用的是 java 编程语言。

下面我们将使用代码行估算方法对支付系统进行成本估算：

1.1. 选择在线支付方式（如支付宝、微信支付等）：约 400 行代码，8 个工作小时

1.2. 跳转至支付平台：约 100 行代码，2 个工作小时

1.3. 支付过程确认与安全验证：约 200 行代码，4 个工作小时

1.4. 支付结果回调与处理：约 150 行代码，3 个工作小时

1.5. 异常处理与提示：约 300 行代码，6 个工作小时

在线支付总工作量：20 个工作小时

- 2.1. 选择现金支付方式：约 50 行代码，1 个工作小时
 - 2.2. 确认支付金额：约 100 行代码，2 个工作小时
 - 2.3. 收银员收款并找零：约 100 行代码，2 个工作小时
 - 2.4. 系统记录现金支付信息：约 100 行代码，2 个工作小时
 - 2.5. 异常处理与提示：约 100 行代码，2 个工作小时
- 现金支付总工作量：10 个工作小时
- 支付系统总工作量：20 + 10 = 30 man-hours.

4.5 后台管理系统

点餐宝后台管理系统十分复杂多样，系统输入输出复杂，功能点用例点数目多，为了便于分析，下面我们将使用类比估算方法对后台管理系统进行成本估算：

为了估算点餐宝后台管理系统的工作量，我们首先分析了以下几个类似的系统以便进行类比估算：

系统名	语言	团队大小	系统规模	工作量
餐馆预订系统	Java	5 人	中等	3000man-hours
外卖点餐平台	Python	8 人	大	5500 man-hours
连锁餐馆管理系统	C#	6 人	中等	3500 man-hours
酒店后台管理系统	PHP	4 人	小	2000 man-hours

通过对比类似系统的特点，我们发现点餐宝后台管理系统与系统 A 和系统 C 更相似。因此，我们将以这两个系统为基础进行类比估算。

点餐宝后台管理系统包含两大功能模块：菜品管理和餐桌管理。从功能上看，这个系统的规模介于系统 A 和系统 C 之间。因此，我们可以根据这两个系统的工作量对点餐宝后台管理系统的工作量进行估算。

以系统 A 和系统 C 的平均工作量为基准：

$$(3000 \text{ man-hours} + 3500 \text{ man-hours}) / 2 = 3250 \text{ man-hours}$$

然后根据点餐宝后台管理系统的具体需求和功能，我们可以对上述基准进行微调。考虑到点餐宝后台管理系统的功能相对较为简单，我们可以适当减少工作量估算。因此，我们将工作量估算为：

$$3250 \text{ man-hours} * 0.9 = 2925 \text{ man-hours}$$

4.6 保障系统

同样是由于保障系统十分复杂，我们为了便于计算，下面我们将使用三点估算方法对保障系统进行成本估算：

为了使用三点估算方法对保障系统的工作量进行估算，我们需要先确定每个任务的最乐观估计(Optimistic Estimate, O)，最可能估计(Most Likely Estimate, M)和最悲观估计(Pessimistic Estimate, P)。然后，我们可以使用以下公式来计算每个任务的估算值： $TE = (O + 4M + P) / 6$ 。最后，我们将所有任务的估算值相加，得到整个保障系统的工作量估算。

例如对于 6.1.1 优化系统性能，我们的估计如下。最乐观估计 (O)：40 man-hours。最可能估计 (M)：60 man-hours。最悲观估计 (P)：100 man-hours。

$$TE = (40 + 4 * 60 + 100) / 6 = (40 + 240 + 100) / 6 = 380 / 6 \approx 63.33 \text{ man-hours}$$

然后我们需要为每个任务分别进行三点估算。

任务编号	最乐观估计 (O)	最可能估计 (M)	最悲观估计 (P)	估算值 (TE)
6.1.1	40	60	100	63.33333
6.1.2	30	50	80	51.66667
6.1.3	20	35	60	36.66667
6.1.4	30	45	75	47.5
6.2.1	50	80	120	81.66667
6.2.2	40	60	100	63.33333
6.2.3	40	65	110	68.33333
6.2.4	25	40	70	42.5
6.3.1	35	55	90	57.5
6.3.2	30	45	75	47.5
6.3.3	20	35	60	36.66667
6.3.4	45	70	110	72.5
6.4.1	30	50	80	51.66667
6.4.2	25	40	65	41.66667
6.4.3	20	35	60	36.66667
6.4.4	15	30	50	30.83333
6.5.1	20	35	60	36.66667
6.5.2	15	25	40	25.83333
6.5.3	10	20	35	20.83333
6.5.4	15	25	45	26.66667
6.6.1.1	10	20	35	20.83333
6.6.1.2	10	15	25	15.83333

6.6.1.3	10	15	25	15.83333
6.6.1.4	10	15	25	15.83333
6.6.2.1	10	20	35	20.83333
6.6.2.2	10	15	25	15.83333
6.6.2.3	10	15	25	15.83333
6.6.2.4	10	20	35	20.83333
6.7.1.1	15	25	40	25.83333
6.7.1.2	5	10	20	10.83333
6.7.1.3	10	15	25	15.83333
6.7.1.4	10	15	25	15.83333
6.7.2.1	20	35	60	36.66667
6.7.2.2	15	25	40	25.83333
6.7.2.3	10	20	35	20.83333

然后将所有任务的 TE 值相加。

得到整个保障系统的工作量估算 1333 man-hours

5. 自下而上工作量估算

点菜宝点菜系统			Man-hours	小计
F1. 点餐系统				
	F1.1. 菜单浏览			185
		按菜品类别浏览	51	
		菜品详情查看	42	
		菜品搜索功能	49	
		菜品推荐功能	43	
	F1.2. 点餐功能			370
		菜品数量选择	88	
		加入购物车	86	
		购物车修改	117	
		确认点餐	79	
	F1.3. 结账功能			370
		查看订单详情	61	
		选择支付方式	113	
		支付确认	76	
		支付结果反馈	120	
	F1.4. 多人点餐功能			554
		添加/删除点餐成员	137	
		合并购物车	142	
		区分个人点餐记录	114	
		按成员分开结账	160	
	F1.5. 特殊要求处理			370

		菜品口味选择	143	
		食材添加/去除	82	
		其他特殊要求备注	145	
	F2. 订单管理			
	F2.1. 订单打印			198
		选择打印设备	70	
		打印订单详情	38	
		打印小票	47	
		打印异常处理	43	
	F2.2. 订单查询			396
		根据订单号查询	122	
		根据顾客信息查询	80	
		根据时间范围查询	121	
		根据订单状态查询	74	
	F2.3. 订单修改			594
		修改订单状态	185	
		修改订单详情	182	
		修改顾客信息	101	
		修改支付方式	126	
	F3. 餐桌状态管理			
	F3.1. 餐桌状态更新			278
		显示餐桌状态	71	
		餐桌状态更改	99	
		同步更新餐桌状态信息至系统	46	
		状态更新异常处理	62	
	F3.2. 餐桌服务员信息 显示			278
		分配服务员至餐桌	63	
		显示服务员信息	48	
		更换服务员	83	
		同步服务员信息变动至系统	84	
	F4. 支付系统			
	F4.1. 在线支付			20
		选择在线支付方式	8	
		跳转至支付平台	2	
		支付过程确认与安全验证	4	
		支付结果回调与处理	3	
		异常处理与提示	6	
	F4.2. 现金支付			10
		选择现金支付方式	1	
		确认支付金额	2	
		收银员收款并找零	2	
		系统记录现金支付信息	2	

		异常处理与提示	2	
	F5. 后台管理系统			
	F5.1. 菜品管理			1870
		添加新菜品	329	
		修改菜品信息	316	
		删除菜品	664	
		菜品分类管理	560	
	F5.2. 餐桌管理			1777
		添加新餐桌	673	
		修改餐桌信息	488	
		删除餐桌	328	
		餐桌布局管理	288	
	F6. 保障系统			
	F6.1. 快速响应			200
		优化系统性能	63	
		减少资源消耗	52	
		合理分配服务器资源	37	
		优化网络传输	48	
	F6.2. 高可用性			256
		系统冗余设计	82	
		负载均衡	63	
		故障切换	68	
		实时监控	43	
	F6.3. 高并发处理			216
		优化数据库查询	58	
		缓存技术应用	48	
		异步处理	37	
		分布式系统设计	73	
	F6.4. 数据安全性			162
		数据加密	52	
		访问控制	42	
		安全审计	37	
		定期安全检查	31	
	F6.5. 数据备份			111
		定期数据备份	37	
		备份数据加密	26	
		备份数据完整性检查	21	
		备份数据存储策略	27	
	F6.6. 日志记录与异常处理			142
		日志记录	68	
		异常处理	73	

	F6.7. 多语言、多平台支持			152
		多语言支持	68	
		多平台支持	83	

通过对点菜宝系统的六个子系统：点餐系统，订单管理，餐桌状态管理，后台管理系统，支付系统，和保障系统的自下而上进行成本分析，我们得出总工作量为 8412 man-hours。

6. 结论

通过对点菜宝系统的六个子系统：点餐系统，订单管理，餐桌状态管理，后台管理系统，支付系统，和保障系统的成本计算过程进行分析，我们得出以下结论。

在进行成本估算时，使用自下而上的估算方法对每个子功能的人天数进行分析，这将有助于更准确地预测项目的总体开发时间和成本。我们使用了使用多种方法（如专家判断法、类比估算法和参数估算法）进行分析，以降低预测误差。

在估算过程中，我们未能充分考虑各个子系统间的交互和集成问题。这些问题可能会增加项目的复杂性和开发难度，从而导致成本和时间的增加。为了确保系统的顺利开发和部署，应该提前规划相应的集成和测试阶段。

各子系统的成本和人天数可能会因项目需求、团队技能水平、技术难度和其他外部因素的变化而发生波动。我们没有在估算过程中加入一定的缓冲以应对这些不确定性。

后续应该优先关注具有较高成本和人天数的子系统，以便在项目初期就对关键任务进行充分的规划和调整。这将有助于提高整个项目的效率和成功率。后续应该定期对成本估算进行审查和更新，以确保项目在整个开发周期中始终保持在预期的时间和成本范围内。这将有助于及时发现潜在问题，并采取相应的措施进行调整。

总之，对于点菜宝系统的成本计算过程，我们采用自下而上的估算方法，结合多种估算技术，确保项目在预期的时间和成本范围内顺利完成。同时，后续应该重视子系统间的交互和集成问题，合理规划项目进度，并根据实际情况进行调整。通过这些措施，我们相信点菜宝系统将能够成功地投入使用，为用户提供高

效、便捷的点餐服务。