



Universidad Tecnológica de Torreón

Estándares y Métricas para el Desarrollo de Software

Proyecto Final Estimación de Puntos de Función

MC. Irelina Díaz Luis

TICS - 4"B"

- Andrés Salvador Rodríguez Mejía
- Amaury Rodríguez Ruiz
- Lucero Alhely Barraza Cedillo
- Luis Fernando Martínez Moya

01/11/2020

Contenido

INTRODUCCION	3
CAPÍTULO I. DESCRIPCIÓN GENERAL DEL PROYECTO.....	5
1.1. Sistema de estudio.....	5
1.2. Objetivos	5
CAPÍTULO II. MARCO TEÓRICO	6
2.1. ¿Qué es la técnica de estimación de Puntos de Función?.....	6
2.2. ¿Para qué se utiliza la técnica de estimación de Puntos de Función?	7
CAPÍTULO III. DESARROLLO DEL PROYECTO	9
3.1. Obtener información del sistema.....	9
3.2. Identificar los componentes del sistema.....	9
3.2.1. Entrada Externa (EI)	9
3.2.2. Salida Externa (EO).....	9
3.2.3. Consulta Externa (EQ).....	9
3.2.4. Archivo Lógico Interno (ILF).....	10
3.2.5. Archivo de Interfaz Externo (EIF).....	10
3.3. Cálculo del número de elementos y su complejidad	10
3.4. Cálculo de los Puntos de Función sin ajustar (PFSA)	10
3.5. Cálculo de los Puntos de Función Ajustados (PFA)	10
3.5.1. Determinación de los niveles de influencia	10
3.5.2. Cálculo del factor de ajuste.....	11
3.5.3. Cálculo de los PFA	11
3.6. Estimación del esfuerzo	11
3.7. Estimación de la duración del proyecto	11
3.8. Cálculo del presupuesto del proyecto	11
3.9. Conclusión final del estudio del proyecto	11
CONCLUSIONES.....	12
BIBLIOGRAFÍA	13

INTRODUCCION

Todo proyecto de desarrollo de software comienza con una etapa de estimación y planificación, en la cual debemos determinar, a partir de los requerimientos del software, cuánto esfuerzo, personal, recursos materiales, tiempo y en última instancia dinero tomará construir el sistema o producto.

Frente a esto, se han desarrollado métodos como el de puntos de función, donde estimamos las funcionalidades usando fórmulas matemáticas basadas en parámetros como el tipo de componente, complejidad, factores del entorno, entre otros aspectos.

¿Qué es el método punto de función?

Es una técnica de estimación de software desarrollada originalmente por Allan Albrecht en 1979 mientras trabajaba para IBM, quien definió conceptos para medir el software a partir de valoraciones de funcionalidades entregadas al usuario y no a partir de aspectos técnicos, con la intención de producir valoraciones independientes de la tecnología y fases del ciclo de vida utilizado.

IFPUG-FPA realiza las valoraciones a partir de la funcionalidad del sistema, primero clasificándolas, luego asignando una complejidad y ponderación a cada una según unas tablas predefinidas, determinando así el valor de puntos de función.

¿Para qué se utiliza?

- Estimación de proyectos de software
- Validar la calidad de las especificaciones funcionales
- Seguimiento y control de proyectos
- Medir la productividad y calidad de nuestro proyecto o servicio de ingeniería de software

Sistema de estudio

El aprendizaje es un proceso de adquisición de habilidades y conocimientos, que se produce a través de la enseñanza, la experiencia o el estudio. Respecto al estudio, puede decirse que es el esfuerzo o trabajo que una persona emplea para aprender algo.

Por otra parte, una técnica es un procedimiento cuyo objetivo es la obtención de un cierto resultado. Supone un conjunto de normas y reglas que se utilizan como medio para alcanzar un fin.

Por lo tanto, una técnica de estudio es una herramienta para facilitar el estudio y mejorar sus logros.

Objetivos de la Técnica Puntos de Función y su importancia en el desarrollo de proyectos de software

Los puntos de función permiten estandarizar las mediciones del tamaño del software, obteniendo estimaciones de mayor exactitud, frente a las que obtendríamos al basarnos solamente en nuestra experiencia y aproximación inexacta.

Objetivo del proceso de medición

- Ser una medida consistente
- Consistente: Dos profesionales analizando un mismo proyecto llegan al mismo resultado
- Simple para minimizar el esfuerzo de la medición

A la hora de desarrollar software empresarial, el éxito o fracaso del proyecto suele medirse en función del desempeño de este frente a las expectativas establecidas inicialmente, de allí la importancia de la exactitud de las mediciones de tamaño del software.

CAPÍTULO I. DESCRIPCIÓN GENERAL DEL PROYECTO

1.1. Sistema de estudio

Este proyecto se basa en una necesidad de un cliente que lo pidió de manera personal, con el objetivo de manipular todo el sistema de su empresa, pidiendo algunos puntos como prioridad, con la finalidad de poder enriquecer el sistema, aprovechándolo al máximo, y sin generar algún error de cálculo.

La función de este sistema es mejorar el manejo de información, cumpliendo con las necesidades dichas y que las satisfaga, manipulando totalmente la información y contactos de esta empresa.

Cualquier tipo de cliente que busque un programa parecido a este, le podría servir, dándole un uso empresarial, que sirva de manera general, aplicándole cambios mínimos.

Existe la posibilidad de crear nuevos módulos, este software fue desarrollado para la interacción del usuario con la empresa, haciendo pedidos desde casa, y como trabajador, tener el manejo de información general, pedidos y reportes, pero podemos ampliar esto, metiendo los proveedores, las recetas y productos más usados, ya que este software solo fue diseñado para el manejo de pedidos, pero podemos centrarlo más en el uso general de un software.

1.2. Objetivos

- Como usuario, puedes tener acceso a pedir alguno de los paquetes que maneja la empresa, creándote un usuario, dándole como datos de entrada, tu nombre, tu número de teléfono y tu dirección.
- Como administrador, tienes acceso total al manejo de la empresa como trabajador, viendo las ventas las diarias y mensuales, con algunas consultas que te ponen al día de cómo se está manejando la empresa, sabiendo mes con mes, los ingresos y gastos totales.
- Tener un reporte la cantidad de ingresos exactos mensuales, semanales y diarios, cumpliendo con todos los requerimientos que podría tener un software básico.
- Mejorar la forma en que un usuario maneja sus pedidos, ya que para algunos puede ser confuso la forma de pedir los paquetes.

CAPÍTULO II. MARCO TEÓRICO

2.1. ¿Qué es la técnica de estimación de Puntos de Función?

Los Puntos de Función constituyen una indicación cuantitativa (métrica) del tamaño y complejidad de un sistema de información, basada en la funcionalidad de ese sistema que puede percibir el usuario. Esa medida cumple con las características más significativas con que debe contar una buena medida del producto final del desarrollo de sistemas, a saber:

- Indicación de la funcionalidad entregada
- Facilidad de comprensión por técnicos y por usuarios
- Independencia con respecto a la tecnología utilizada
- Posibilidad de aplicación al comienzo del ciclo de vida del desarrollo
- Consistencia y auditabilidad.

Como se ha dicho antes, los Puntos de Función se pueden usar como referencia para medir productividad. Por ejemplo

Tasa (ritmo) de Desarrollo = $PF / Hs\text{-Hombre de desarrollo}$

Tasa de Mantenimiento = $Hs\text{-H de mantenimiento} / PF \text{ de la Aplicación}$

O para medir la calidad. Por ejemplo

- Índice de Calidad = $Hs\text{-H para reparar defectos} / PF$
- Densidad de Defectos = $Cantidad \text{ de Defectos} / PF$

En esencia, los PF son una métrica calculada que consiste en la suma, ponderada de acuerdo con su complejidad, de las entradas, las salidas, las consultas, los archivos lógicos y las interfaces que corresponden a una aplicación.

Según el manual de IFPUG los objetivos del Análisis con Puntos de Función (APF) son los siguientes:

- Medir lo que el usuario solicitó y recibió
- Medir en forma independiente de la tecnología usada para la implementación
- Proporcionar una medida de tamaño para dar apoyo a análisis de productividad y de calidad
- Proporcionar un medio para la estimación del esfuerzo requerido para el desarrollo de software
- Proporcionar un factor de normalización para realizar comparaciones de software.

El APF mide una aplicación en dos etapas. En la primera se determina una cantidad de PF no ajustados que reflejan la funcionalidad específica que recibe el usuario. En la segunda etapa se evalúa un factor de ajuste que refleja la funcionalidad general producida para el usuario.

Es fundamental tener en cuenta que la funcionalidad producida por una aplicación es evaluada en función de qué es lo que entrega la aplicación, y no cómo lo entrega. Sólo se deben contar los componentes que han sido solicitados por los usuarios. Los componentes que se consideran son cinco (salidas, entradas, consultas, archivos, interfaces) y se dividen en dos grupos denominados Tipos de Función, de la siguiente manera:

Datos:

- Archivos Lógicos Internos (Internal Logical Files - ILF)
- Archivos de Interface Externos (External InterfaceFiles - EIF)
- Transacciones:
- Entradas externas (External Inputs - EI)
- Salidas Externas (External Outputs - EO)
- Consultas Externas (External Inquiries - EQ)

A cada uno de esos Tipos de Función se le asigna un peso, extraído de una tabla (según se verá más adelante) según su complejidad funcional: Baja, Media, Alta. La complejidad funcional de los Tipos de Función "Datos" se define en base a la cantidad de Tipos de Registro (Record Types - RET) y de Tipos de Datos (Data Element Types - DET). La complejidad de los Tipos de Función "Transaccionales" se basa en la cantidad de Tipos de Archivos Referenciados (File Type References - FTR) y de Tipos de Datos (Data Element Types - DET).

2.2. ¿Para qué se utiliza la técnica de estimación de Puntos de Función?

La utilización de Puntos de Función ha ido aumentando en forma importante con el pasar de los años; su difusión puede comprobarse con la creciente cantidad de asistentes a las reuniones del IFPUG año a año. Capers Jones suele repetir en cada oportunidad que se le presenta que actualmente los PF son ya más utilizados que LOC como medida de tamaño de software (como lo escuchó decir el autor de este trabajo a fines de 1993 en la Cuarta Reunión Internacional de Applications of Software Measurement, ASM '93, en Orlando, EE.UU. Aunque en esa ocasión hubo no pocos comentarios en contra de esa opinión y en favor de la no perdida primacía de LOC como métrica aun más usada de tamaño de software). Pero en última instancia, además de los potenciales problemas de definición en el uso de LOC, existe el problema de la medición de productividad. La productividad se puede definir como "la cantidad de bienes o servicios producidos por unidad de insumos", donde el principal insumo en el caso del software es la mano de obra.

En este momento una pregunta válida podría ser ¿Por qué medir la productividad? La respuesta podría tener varios componentes, por ejemplo:

- Para medir cuanto (producto) se le entrega a los usuarios

- Para saber cuán eficientes somos
- Para saber cómo nos comparamos con otros grupos de desarrollo
- Para saber si estamos mejorando
- Para evaluar el efecto del uso de nuevas herramientas o técnicas
- Para poder estimar proyectos futuros
- Para poder determinar cuándo reemplazar aplicaciones.

Cuando se utilizan LOC producidas por mes-hombre, o por hora-hombre, debe tenerse cuidado en que no se produzcan cambios en el entorno tecnológico para que esa medida siga siendo válida. Por ejemplo, y como bien saben los que se desempeñan en la actividad de manufactura de productos, cuando existen costos fijos y se reduce la cantidad de unidades producidas, el costo por unidad sube. El desarrollo de software tiene asociado un porcentaje significativo de costos fijos, o por lo menos inelásticos, que no están vinculados con la programación. Así es que cuando se utilizan, por ejemplo, lenguajes de programación más poderosos, el resultado es que se reducen las "unidades" que se producen para un determinado programa, o sistema, y por lo tanto el costo por unidad aumenta. Esto puede crear problemas con el uso de una medida de productividad basada en LOC por hora-hombre. Justamente, la idea de A.J. Albrecht de IBM al presentar en octubre de 1979 (en una reunión conjunta de SHARE/GUIDE/IBM en Monterey, California, EE.UU.) su nuevo método de medición con Puntos de Función, o Puntos Funcionales, fue la de hacer disponible un método que permitiese independizar la medida de productividad del lenguaje de programación específico utilizado.

Algunas medidas que podrían ser empleadas son las siguientes:

- Productividad Tasa de desarrollo = $PF \text{ (desarrollo)} / \text{Esfuerzo (Horas-hombre)}$
- Calidad Densidad de Defectos = $\text{Defectos} / PF \text{ de la Aplicación}$
- Índice de Calidad = $\text{Horas para Corregir Defectos} / PF$
- Plazos Tiempo = $PF \text{ (desarrollo)} / (\text{Tasa de desarrollo} * \text{Cant.personal})$

Para poder realizar los cálculos requeridos evidentemente es necesario contar con mediciones de una serie de atributos complementarios.

CAPÍTULO III. DESARROLLO DEL PROYECTO

3.1. Obtener información del sistema

Con base en los requerimientos del cliente, se diseñará un punto de venta en el cual los clientes pueden tener acceso a compra de producto que antes ha sido estipulado por el cliente, haciendo uso de plataformas tipo “Uber-Eats” y “Rappi” para tener una idea de que es lo que se desea. El sistema será utilizado por una cantidad indefinida de usuarios, de manera visible, pero, solo una persona tendrá acceso a las modificaciones pertinentes en un entorno único, diseñado en un lenguaje de cuarta generación con utilización de bases de datos en un equipamiento UNIX.

3.2. Identificar los componentes del sistema

3.2.1. Entrada Externa (EI)

- Alta de clientes (1 archivo accedido y 3 atributos) Baja 3
- Modificación de los datos de los clientes (1 archivo accedido y 3 atributos) baja 3
- Alta productos(1 archivo accedido y 3 atributos) baja 3
- Baja de producto (1 archivo accedido y 1 atributo)baja 3
- Alta pedidos de clientes (1 archivo accedido y 6 atributos) baja 3
- Modificación de pedidos de clientes(1 archivo accedido y 3 atributos) baja 3
- Alta de trabajadores (1 archivo accedido y 6 atributos) baja 3
- Baja de trabajadores (1 archivo accedido y 1 atributo)baja 3
- Modificación de datos de productos(1 archivo accedido y 3 atributos) baja 3
- Modificación de los datos de los trabajadores (1 archivo accedido y 6 atributos) baja 3

3.2.2. Salida Externa (EO)

- Reporte de clientes preferentes (1 archivo accedido y 3 atributos) baja 3
- Reporte de productos más vendidos (1 archivo accedido y 2 atributos) baja 3
- Factura de pedidos de clientes (1 archivo accedido y 1 atributo) baja 3
- Reporte de ventas por trabajador (1 archivo accedido y 1 atributo) baja 3
- Reporte diario de ventas (1 archivo accedido y 1 atributo) baja 3
- Reporte mensual de ventas (1 archivos accedido y atributo) baja 3
- Listado de clientes (1 archivo y 4 atributos) baja 4
- Listado de productos (1 archivo y 4 atributos) baja 4
- Listado de trabajadores (1 archivo y 4 atributos) baja 4

3.2.3. Consulta Externa (EQ)

- Seguimiento de pedido de clientes (1 archivo accedido y 3 atributos) baja 3

3.2.4. Archivo Lógico Interno (ILF)

- Tabla de clientes (1 registro lógico y 3 atributos) baja 7
- Tabla de trabajadores (1 registro lógico y 6 atributos) baja 7
- Tabla de productos (1 registro lógico y 3 atributos) baja 7
- Tabla de pedidos (1 registros lógicos y atributos) baja 7

3.2.5. Archivo de Interfaz Externo (EIF)

3.3. Cálculo del número de elementos y su complejidad

Caso de Uso	Cantidad	Peso	Resultado
EI	10	Baja 3	30
EO	9	6 x Baja 3 3 x Baja 4	30
EQ	1	Baja 3	3
ILF	4	Baja 7	28
EIF	0	0	0
Puntos de función sin ajustar			91

3.4. Cálculo de los Puntos de Función sin ajustar (PFSA)

PFSA=91

3.5. Cálculo de los Puntos de Función Ajustados (PFA)

3.5.1. Determinación de los niveles de influencia

No.	Características generales del sistema	Nivel de Influencia
1	Comunicación	5
2	Procesamiento de datos distribuidos	5
3	Rendimiento	1
4	Configuración	0
5	Tasa de trabajos	0
6	Entrada	4
7	Interfaz	5
8	Actualización On-line	0
9	Procesamiento	0
10	Reusabilidad	4
11	Facilidad de implementación	0
12	Facilidad de operación	0
13	Múltiples instalaciones	0
14	Facilidad de cambios	1
TOTAL (Nivel de Influencia)		25

3.5.2. Cálculo del factor de ajuste

Factor de Ajuste = (Nivel de Influencia * 0.01)+0.65

Factor de Ajuste= 0.9

3.5.3. Cálculo de los PFA

PFA = (PFSA * Nivel de Influencia)

PFA = 81.9

3.6. Estimación del esfuerzo

H/H = PFA * Horas PF promedio

H/H = 81.9 * 8

H/H = 655.2

3.7. Estimación de la duración del proyecto

H/H=655.2 / 4Desarrolladores

Horas = 163.8

Días = 27.3

Meses = 1.365 para desarrollar el software de lunes a viernes 6 horas diarias con 4 desarrolladores.

3.8. Cálculo del presupuesto del proyecto

Sueldo mensual Desarrolladores = \$12,000.00

Otros Gastos del Proyecto = \$20,000.00

Costo = \$85,520.00

3.9. Conclusión final del estudio del proyecto

El desarrollo de software de un punto de venta en línea, tendrá una duración de 1.365 meses en una jornada laboral de 6 horas, de lunes a viernes con 4 desarrolladores, y tendrá un presupuesto de \$85,520.00 + IVA

CONCLUSIONES

El análisis es una etapa fundamental dentro de la realización de una aplicación, esta etapa se puede resumir en una sola frase: Entender el problema. Cuando terminamos el análisis tenemos ya una comprensión mayor del problema, sabemos cuáles son las abstracciones claves, y empezamos a estudiar cómo se desenvuelve la aplicación en el tiempo.

También expresan los requerimientos funcionales que los usuarios comunicaron al sistema durante la redacción del pliego de condiciones.

- Comprobar que el sistema cumple dichos requisitos en el momento de la entrega.
- Determinar las fronteras del sistema (a donde podría llegar).
- Escribir la documentación del sistema.
- Confeccionar juegos de test (seguir la ideología establecida).

BIBLIOGRAFÍA

https://www.researchgate.net/publication/291695346_Puntos_de_Funcion

<https://definicion.de/tecnica-de-estudio/>

<http://www.pmoinformatica.com/2015/04/estimacion-puntos-funcion-introduccion.html>