

SISTEMA DE IDENTIFICACIÓN AUTOMÁTICA DE CLIENTES PARA DESPACHO DE TAXIS Y ADMINISTRACIÓN DE INFORMACIÓN HISTÓRICA.

Automatic Customer Identification System for Cab Dispatch and Historical Information Administration.

RESUMEN

En este documento se presenta a manera de ejemplo la sistematización del proceso de despacho de taxis en la empresa PrimerTax S.A, desarrollada con base en la *Metodología Ágil XP*; se mostrará la aplicación de las diferentes fases propuestas en la metodología, igualmente el desarrollo de los documentos de soporte sugeridos durante la práctica.

PALABRAS CLAVES:

Sistematización, proceso, empresa PrimerTax S.A despacho, Metodología Ágil XP, diferentes, fase, Cliente, Cliente-Servidor, practica, desarrollo.

ABSTRACT

This document introduces an example about the systematization of a cab dispatch process in the company PrimerTax S.A. Developed on the extreme programming (XP) methodology, the application of different phases in it will be shown, as well as the development of support documents suggested on practice stages.

KEYWORDS:

systematization, process, company PrimerTax S.A., dispatch, methodology programming (XP) methodology, different, phases, client, practice, development

DIXON FERNANDO CANO LARGO

Estudiante de Ingeniería de Sistema y Computación.
Universidad Tecnológica de Pereira
webalcaldia@gmail.com

LINA MARÍA MUÑOZ TORRES

Estudiante de Ingeniería de Sistema y Computación.
Universidad Tecnológica de Pereira
Anilator19@gmail.com

GUILLERMO ROBERTO SOLARTE MARTINEZ

Ingeniero de Sistemas,
Profesor Auxiliar
Universidad Tecnológica de Pereira
gsolarte294@gmail.com

1. INTRODUCCIÓN

Las tecnologías de la informática han desempeñado un importante papel a lo largo de la historia como facilitadores de procesos. Consiguiente a esto, unos de los procesos dentro de una empresa de mayor impacto son la toma de decisiones y ofrecer un buen servicio a los clientes.

Dentro de la toma adecuada de decisiones en una empresa es fundamental la calidad y consistencia de la información y dentro de la calidad del servicio se encuentra un importante factor como lo es la agilidad en el servicio prestado.

En una sociedad tan competitiva, las empresas buscan mejorar sus procesos y dichos procesos están ligados al modelo de negocio, es por esto que una implementación de un software a la medida, que satisfaga unos requerimientos precisos de gestión de información y de mejoramiento en los tiempos de servicio puede ser de gran ayuda para la obtención oportuna de información

necesaria para el proceso administrativo y el mejoramiento de los servicios ofrecidos por la empresa.

Cada día los sistemas de información ofrecen mayores capacidades a bajo costo, y el uso de los mismos es fundamental en el desarrollo de actividades administrativas y operativas, ya que agilizan el trabajo y permiten un mayor control en el uso y almacenamiento de la información referente al negocio.

Algunas empresas de despacho de taxis no cuentan con sistemas de información que permitan prestar servicios y que al mismo tiempo guarde de manera adecuada la información referente a los servicios prestados. Sin embargo sí tienen la infraestructura necesaria para poder obtener la información desde consolas de identificación de llamadas que pueden ser conectadas a equipos de cómputo a través del puerto serial para luego procesar esta información. Igualmente se ha determinado la posibilidad de que estas empresas puedan adquirir equipos con capacidades medias ya que

Fecha Recepción: 9 de Septiembre de 2010

Fecha aceptación: 15 de Noviembre de 2010

el sistema no requiere de características demasiado exigentes.

Por estas razones expuestas nos hemos dado a la tarea de investigar (en nuestro proyecto de grado) al respecto con el fin de realizar una herramienta de software que cumpla con los requerimientos y brinde solución a los problemas de información que tienen estas empresas. Otra característica esencial será desarrollar dicha herramienta con una alta facilidad de uso para que el rendimiento, en el préstamo de servicio de despacho de móviles, no sea afectado de manera negativa para el negocio en el periodo de adaptación al sistema por parte de los operadores.

2. FASES DE LA METODOLOGÍA XP

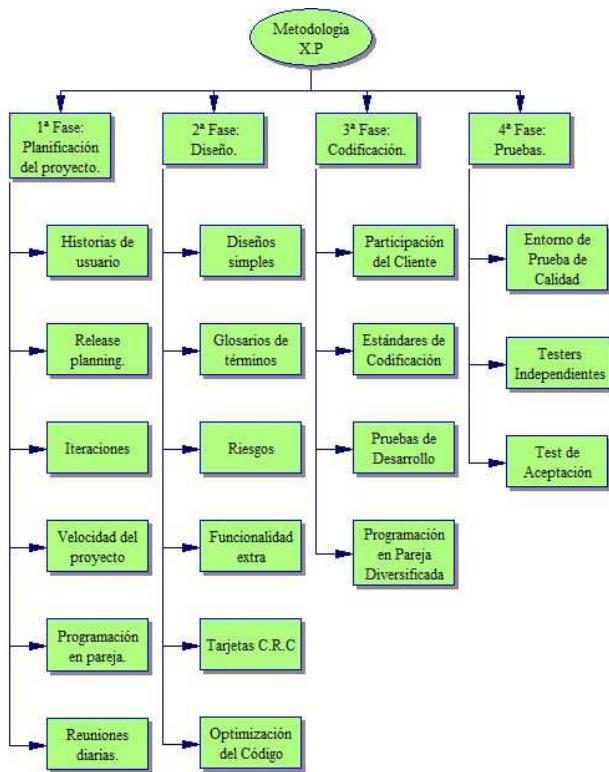


Figura 1. Fases de la Metodología XP[1].

2.1 PRIMERA FASE

En esta fase, se plantean a grandes rasgos las Historias de Usuario que son de interés para la primera entrega del Producto. Las historias de usuario son construidas con la información proporcionada por los clientes y el equipo de desarrollo comienza a familiarizarse con las herramientas, metodología y prácticas que serán usadas para realizar el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo uno o varios

prototipos iniciales. Esta primera fase toma pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con el diseño [2].

2.1.1 Historias de Usuario.

Las historias de usuario fueron diligenciadas por alguno de los desarrolladores, con el fin de que el cliente pudiera concentrar su atención en el análisis del requerimiento o en el caso de que se estuviera evaluando el diseño o una entrega de iteraciones. Por otro lado se contó con la participación de los usuarios finales (operadores) para la redacción y recolección de información para algunas historias de usuario. Pese a que el cliente no fue quien escribió y diligenció la historia de usuario, siempre se contó con su revisión previa antes de finalizar la reunión.

Desde el punto de vista de nivel de detalle, las historias de usuario se realizaron de manera jerárquica en la que las historias de usuario más específicas contenían los aspectos importantes de diseño, pero no se profundizaban en procesos o con descripciones de carácter técnico, las historias de usuario más generales solo mencionan los requerimientos con el mínimo nivel de detalle.

2.2 SEGUNDA FASE

En esta fase se establece la prioridad de cada Historia de Usuario y, correspondientemente, los programadores establecen una estimación de esfuerzo necesario para cada una de ellas. Se realizan acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.

Las estimaciones de esfuerzo asociado a la implementación se establecen con una medida en puntos. Un punto, equivale a una semana ideal de programación.

2.2.1 Iteraciones.

Se realizaron dos reuniones iniciales, en la primera nos reunimos con el señor Juan Carlos Tamayo, quien nos planteó a grandes rasgos los requerimientos del sistema. La segunda reunión se llevó a cabo en las oficinas de PrimerTax S.A en donde el Ingeniero Jesús Ernesto Duque Ocampo gerente de PrimerTax S.A. expresó con mayor detalle el sistema que tenía en mente. Adicionalmente se tuvieron en cuenta temas como la experticia de los usuarios finales en el manejo de sistemas de información, el tiempo que disponía el gerente para la realización de las reuniones y se asignaron los colaboradores que para este caso fueron dos operadores que realizarían las pruebas del sistema.

XP recomienda que el orden de las historias implementadas en las iteraciones sean determinadas por el cliente, sin embargo, las historias de usuario entregadas en las iteraciones fueron seleccionadas según la disponibilidad de los sistemas externos (hardware), la dificultad de la implementación y el riesgo de los

cambios; estos aspectos también fueron mezclados con la prioridad del negocio. La razón por la que se tomaban primero las historias de usuario de mayor dificultad fue porque las mejoras podrían ser entregadas en más de una futura iteración, dando más tiempo de perfeccionarlas antes de dar por finalizado el proyecto.

Para aproximar el tiempo de entrega se tuvo en cuenta los siguientes aspectos:

- a) XP recomienda que una semana conste de 40 horas semanales, es decir 8 horas laborando de lunes a viernes. Sin embargo en el proyecto se estableció que la semana estaría compuesta por 30 horas semanales que serían distribuidas en 4 horas de lunes a viernes, 5 horas sábado y domingo, estos horarios de trabajo fueron convenidos debido a conocimiento de que el grupo de desarrollo tenía obligaciones adicionales al proyecto.
- b) Las reuniones se harían en horarios luego de las seis (6) de la tarde, esto para evitar coincidir con horarios críticos en cuanto a la demanda del servicio en la empresa de despacho.
- c) Las reuniones con el cliente para evaluar las entregas realizadas y la evaluación de nuevas historias de usuario se realizaban los días sábado.

2.2.2 Pruebas.

No es fácil seguir este planteamiento de XP por varios motivos. En primero lugar, el framework escogido para el desarrollo no provee una facilidad para realizar las pruebas, ni tampoco se encontró una herramienta, con licencia de libre uso, para este fin.

Se realizaban códigos con pequeñas pruebas para el funcionamiento de las funciones, pero se encontraron muchas limitaciones, por ejemplo al tratar de hacer pruebas a elementos gráficos tales como Ventanas y Botones, o realizar pruebas a partes del código que se comunican con dispositivos externos por medio del puerto serial. Los reportes son otro obstáculo para realizar las pruebas, ¿Cómo probar un reporte que fue diseñado en papel?. O ¿Cómo saber si el programa obtuvo la hora del sistema de forma correcta?, si no es comparándola con la propia hora del sistema.

El carácter privado de muchos métodos representa un obstáculo insalvable para hacerle pruebas, ya que como solo puede ser accedido desde el interior de la clase no puede ser probado independientemente. La solución que se tomó en este sentido fue probarlo a través del método público que hace uso de él. Si el método público pasa determinada prueba, se asume que el método privado también la ha pasado.

Todos los elementos anteriores representaron obstáculos en el desarrollo de las pruebas y plantearon una inquietud importante sobre el alcance del concepto "codificar primero las pruebas". ¿Se trata de codificar SIEMPRE una prueba antes que el código, o solo aquellas clases encargadas de realizar la lógica del negocio?. Debido que XP no tiene una respuesta clara a esta inquietud, el grupo de desarrollo optó por probar solo aquellas clases que ejecutan la lógica del negocio, que en definitiva son las más importantes de las cuales se debe tener garantía de estar bien construidas.

Es importante resaltar las ventajas que representa hacer las pruebas antes que el código de la aplicación. En primer lugar el tiempo que toma escribir determinado código después de haber implementado la prueba es considerablemente menor que si no hubiera escrito la prueba antes. En segundo lugar al hacer la prueba se identifican de manera precisa cuales son los casos especiales y rutas alternas que deben ser consideradas dentro del código haciendo de este un producto más robusto y tolerante a fallos. Finalmente una ventaja no expuesta de forma vehemente por XP es la estética del código de la aplicación. Al tener tal claridad sobre cómo debe ser escrito determinado código, la tarea de realizarlo es más sencilla y el mismo queda organizado de forma más estética.

2.3 TERCERA FASE

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. En la primera iteración se intenta establecer una arquitectura del sistema que pueda ser usada durante el resto del proyecto. Los elementos que deben tomarse en cuenta para la elaboración de una iteración son los siguientes:

1. Historias de Usuario no abordadas.
2. Velocidad del Proyecto.
3. Pruebas de aceptación no superadas en la iteración anterior.
4. Tareas no terminadas en la iteración anterior.[3][4]

2.3.1 Velocidad Del Proyecto

Durante la realización del proyecto no se tuvo en cuenta la estimación de la velocidad del proyecto para la planeación de entregas de versiones o la duración de las iteraciones. El grupo de desarrollo decidió planear las entregas y el número de historias de usuario por cada iteración según el número de puntos asignados a las historias de usuario (el número de puntos asociados a las historias de usuario fueron asignados según la dificultad de la implementación de la historia).

Durante la estimación de los puntos por cada historia de usuario y el orden de la entrega en las iteraciones se tuvo en cuenta la puntuación del riesgo en el desarrollo y la prioridad en negocio, de manera cualitativa y cuantitativa, el riesgo causado por cambios en

requerimientos y por reprocesos en el desarrollo debido a errores de lógica o de comunicación con los componentes externos de hardware se cuantificaron de la siguiente forma:

- a) **BAJO:** corresponde a cambios o reprocesos que modifican los puntos estimados para la historia de usuario en un tiempo no superior a 8 horas.
- b) **MEDIO:** corresponde a cambios o reprocesos que modifican los puntos estimados para la historia de usuario en un tiempo superior a 8 horas y menor a 16 horas.
- c) **ALTO:** corresponde a cambios o reprocesos que modifican los puntos estimados para la historia de usuario en un tiempo superior a 16 horas

El orden dado a la entrega y desarrollo de las historias de usuario se planeó conforme a la importancia de la funcionalidad específica a la que hacía referencia la historia de usuario. De la siguiente manera se calificó la prioridad de las historias de usuario:

- a) **ALTA:** corresponde a funcionalidades que hacen parte del proceso operativo diario del sistema.
- b) **MEDIA:** corresponde a funcionalidades que hacen parte del proceso administrativo periódico del sistema.
- c) **BAJA:** corresponde a funcionalidades que hacen parte de mejoras que no afectan con gran impacto las funcionalidades de alta o mediana prioridad.

2.4 CUARTA FASE

Esta fase de Producción requiere de pruebas adicionales y revisiones de rendimiento, adicionalmente se toman decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante la fase.[5]

A diferencia de las metodologías pesadas, el diseño se realiza durante todo el tiempo de vida del proyecto, siendo constantemente revisado.[6]

Una de las principales piezas de diseño empleada en el proyecto fueron las tarjetas CRC que no sólo sirvieron como columna vertebral de este, sino que también, tal y como lo hace el diagrama de clases en las metodologías tradicionales, fueron la base del modelo Entidad Relación, elaborado para modelar la base de datos.

Cada Tarjeta CRC se convirtió en un objeto, sus responsabilidades en métodos públicos y sus colaboradores en llamados a otras clases, además se decidió hacer una modificación a lo que aconseja XP y fue tener los atributos en la misma sección de las responsabilidades para ayudar a encontrar con mayor facilidad a quien le correspondía tener una nueva responsabilidad.

3. CONCLUSIONES Y RECOMENDACIONES

El desarrollo e implementación del software desarrollado fue exitosamente concretado, y adicionalmente permitió adquirir conocimientos del negocio que permitirán en un futuro desarrollar herramientas más competitivas y prácticas que complementen la actual y proliferen los resultados obtenidos actualmente.

El cliente mostró interés por el desarrollo ininterrumpido del proyecto, antes de dar inicio al desarrollo del proyecto, el cliente manifestó la necesidad de poder tener control o información del avance del proyecto; esto fue posible gracias a las entregas realizadas al finalizar cada iteración, el cliente se encontraba muy satisfecho gracias a que podía tener la posibilidad de apreciar el avance y manifestar libremente los cambios que deseara en el sistema.

La distribución del conocimiento del código es una ventaja significativa, ya que existen funcionalidades en las que se puede recurrir a lluvias de ideas y el entendimiento mutuo del código facilitó que las ideas plasmadas en lógica de programación fueran correctamente contextualizadas y entendidas por todos los integrantes del equipo de desarrollo.

La metodología XP permitió que el cliente y los usuarios finales afinaran el proyecto conforme iba avanzando el desarrollo, a medida que surgían nuevos requerimientos los riesgos asociados al proyecto aumentaban, sin embargo, se estableció con el cliente un mecanismo de análisis de riesgos que permitían establecer si un requerimiento se encontraba o no dentro del alcance del proyecto. Sumado a esto, cada cambio solicitado y que se convenía implementar era transformado en una historia de usuario, esto conllevaba a la planeación de un tiempo asociado a esta implementación y una prioridad.

El concepto de Spike Solution se convierte en una práctica que puede ser usada en cualquier metodología para la solución de problemas técnicos, ya que es la oportunidad que brinda a un desarrollador de solucionar un problema o adquirir un nuevo conocimiento abstrayéndose totalmente cualquier particularidad del proyecto. El hecho de trabajar sobre un tema en particular permite llegar a un grado de profundización en el tiempo que no se logra otra forma, dándole al programador la habilidad de inyectar dichos conocimientos al proyecto.[7]

Se debe definir la participación del cliente dentro del proceso, en caso de que el cliente no tenga la suficiente disponibilidad, entonces se puede plantear la posibilidad de tener la asistencia de otra u otras personas autorizadas y seleccionadas por el cliente para dicha participación.

Es importante aclarar al cliente la forma de trabajo de la metodología, de manera que el cliente conozca el mecanismo de trabajo, los horarios y la distribución de las tareas.

El cliente debe ser informado de cualquier novedad presentada en cualquiera de las fases de desarrollo.

4. BIBLIOGRAFÍA

- [1] Wells Dom. (1999, 2000, 2001, 2004, 2009 Extreme Programming: A gentle introduction [Online], Available: <http://www.extremeprogramming.org/rules.html>
- [2] Wells Dom. (1999, 2000, 2001, 2004, 2009 Extreme Programming: A gentle introduction [Online], Available: <http://www.extremeprogramming.org/rules/userstories.html>
- [3] Wells Dom. (1999, 2000, 2001, 2004, 2009 Extreme Programming: A gentle introduction [Online], Available: <http://www.extremeprogramming.org/rules/iterationplanning.html>
- [4] Wells Dom. (1999, 2000, 2001, 2004, 2009 Extreme Programming: A gentle introduction [Online], Available: <http://www.extremeprogramming.org/rules/velocity.html>
- [5] Wells Dom. (1999, 2000, 2001, 2004, 2009 Extreme Programming: A gentle introduction [Online], Available: <http://www.extremeprogramming.org/rules/unittests.html>
- [6] Wells Dom. (1999, 2000, 2001, 2004, 2009 Extreme Programming: A gentle introduction [Online], Available: <http://www.extremeprogramming.org/rules/metaphor.html>
- [7] Wells Dom. (1999, 2000, 2001, 2004, 2009 Extreme Programming: A gentle introduction [Online], Available: <http://www.extremeprogramming.org/rules/spike.html>