



UNIVERSIDAD DE CARABOBO
FACULTAD EXPERIMENTAL DE CIENCIAS Y TECNOLOGÍA
DIRECCIÓN DE POSTGRADO
ESPECIALIZACIÓN EN DESARROLLO DE SOFTWARE



ESPECIFICACIÓN DE REQUISITOS PARA REDUCIR COSTOS DE UN PROYECTO DE SOFTWARE. CASO: EMPRESA DE TELECOMUNICACIONES

AUTOR: Lic. Darwin Rocha

TUTOR: Dra. Mirella Herrera

Trabajo Especial de Grado presentado
para optar al título de Especialista en Desarrollo de Software

Valencia, Marzo de 2022



UNIVERSIDAD DE CARABOBO
Facultad Experimental de Ciencias y Tecnología
Dirección de Postgrado FACYT



**ACTA VEREDICTO DEL TRABAJO ESPECIAL DE GRADO
PROGRAMA: ESPECIALIZACIÓN EN DESARROLLO DE SOFTWARE**

Quienes suscribimos, profesores Mirella Herrera C.I.V-8.044.677, Dinarle Ortega C.I.V-8.611.660 y José Canache C.I.V-15.607.108, integrantes del Jurado designado por el Consejo de Postgrado de la Facultad Experimental de Ciencias y Tecnología de la Universidad de Carabobo, en su reunión ordinaria virtual N° 02/2022 de fecha 14/03/2022, para conocer y decidir acerca del trabajo especial de grado titulado: “ESPECIFICACIÓN DE REQUISITOS PARA REDUCIR COSTOS DE UN PROYECTO DE SOFTWARE. CASO: EMPRESA DE TELECOMUNICACIONES.”, el cual fue aprobado luego de realizado el arbitraje doble ciego juicio de pares, para ser presentado oralmente en el IV Congreso Nacional de Investigación e Innovación en Ciencias Económicas y Sociales “Hacia La Nueva Visión del Planeta”, organizado por la Facultad de Ciencias Económicas y Sociales de la Universidad de Carabobo. Asimismo, el trabajo presentado por el Licenciado Lic. Darwin Rocha C.I.V-18.082.538 se realizó bajo la tutoría académica de la Profesora Mirella Herrera C.I.V- 8.044.677, como requisito para optar al título de Especialista en Desarrollo de Software, todo ello conforme a lo estipulado en los artículos 79, 128 y 136 del Reglamento de los Estudios de Postgrado de la Universidad de Carabobo (REPUC). En este sentido, como jurado evaluador dejamos constancia de lo siguiente:

1. Según se evidencia en los recaudos consignados, el trabajo especial de grado corresponde a una producción intelectual aceptada en el IV Congreso Nacional de Investigación e Innovación en Ciencias Económicas y Sociales de la Universidad de Carabobo, a efectuarse del 29 al 31 de marzo. Cabe señalar, que este evento científico utiliza arbitraje doble ciego juicio de pares, para la aceptación de los trabajos.
2. El trabajo presenta un caso de estudio sobre la experiencia práctica llevada a cabo en una empresa de Telecomunicaciones, para la cual se aplicaron un conjunto de herramientas y técnicas propias de la ingeniería de requisitos. Específicamente, el estudio se realizó durante el proceso de recopilación, análisis, especificación y verificación de las necesidades del cliente y la

correspondiente medición del impacto en términos cuantitativos en la reducción de costos de los proyectos de software.

3. En este artículo se observa la aplicación del Cuerpo de Conocimientos de la Ingeniería del Software específicamente a lo atinente a la Ingeniería de Requisitos, de una forma integral y confiable. Todo ello, aplicado a un caso real, que es el objetivo ulterior de un trabajo final en el programa de Especialización en Desarrollo de Software.

Cumplida la revisión de rigor del trabajo antes mencionado por cada uno de los miembros del Jurado, decidimos por UNANIMIDAD que cumple con los requisitos exigidos para su aprobación, por lo que emitimos el veredicto APROBADO, todo conforme a lo dispuesto en la normativa legal vigente.

En fe de lo cual se levanta y firma la presente Acta a los veintiún días del mes de marzo de dos mil veintidós.

Mirella Herrera
C.I.V- 8.044.677
Coordinador del Jurado (Tutor)
FACYT- UC

Dinarle Ortega
C.I.V- 8.611.660
Miembro del Jurado
FACYT - UC

José Canache
C.I.V- 15.607.108
Miembro del Jurado
FACYT- UC



Especificación De Requisitos Para Reducir Costos De Un Proyecto De Software.

Caso: Empresa De Telecomunicaciones.

Darwin Rocha^{1,2}, Francisca Grimón³, Mirella Herrera³

¹ correspondencia: darwinrocha85@gmail.com

² estudiante de Postgrado, FACYT-UC

³ profesora del programa de postgrado FACYT-UC

Resumen

La especificación de requisitos, es un área central en el desarrollo de software, no solo porque conlleva entender con claridad los requerimientos del cliente o del usuario final y su nivel de satisfacción, sino que como se evidencia en este estudio, tiene un impacto significativo en los costos de los proyectos. El presente trabajo muestra un análisis cuantitativo del impacto de la aplicación de técnicas para la educación de requisitos de software, recolectando información fiable y medible de una fábrica de software en una empresa de telecomunicaciones, radicada en Bogotá Colombia. Los resultados reportan una mejora substancial superior al 75% de reducción de horas hombre de programación y un 91.5% de disminución en los costos con tan solo aplicar técnicas básicas en la documentación y pruebas de los requisitos. Se concluye que con la aplicación de técnicas, métodos, herramientas para gestionar los requisitos, redundan en un impacto altamente positivo en los tiempos de entrega, costos del proyecto y por ende en la satisfacción de los equipos de desarrollo y calidad, así como en el cliente y usuario final.

Palabras Clave: ingeniería de requisitos, elicitación de requisitos, trazabilidad, documentación de requisitos, pruebas de requisitos.

Introducción

En el proceso de desarrollo de software, la Ingeniería de Requisitos (IR) forma parte de las etapas tempranas de un proyecto. El cuerpo de conocimientos de la Ingeniería del Software (SWEBOK) contempla un conjunto de fases diferenciadas entre las que se cuentan: elicitación, análisis, especificación y validación, las cuáles refieren a un proceso completo que permitirá entregar un conjunto de requisitos especificados de manera tal, que el programador realice su trabajo de forma eficiente y el producto logrado cumpla con las expectativas del cliente haciendo un uso eficiente de los recursos.

Sin embargo, es bien conocido el alto porcentaje de fracaso de los proyectos debido a que, en la práctica, los desarrolladores dedican poco tiempo a la formalización de los requisitos lo que acarrea la incursión de costos adicionales, incremento del presupuesto del proyecto que el cliente no está dispuesto a asumir, retardo en las entregas, clientes insatisfechos con el producto logrado, entre otros.

Este artículo plantea un conjunto de técnicas y herramientas utilizadas para la fase de especificación de los requisitos, en el desarrollo de un proyecto de software para una empresa colombiana comercializadora de servicios de telecomunicación. El caso de estudio plasma las incidencias y fallas en el desarrollo del software y la notable mejora después de adoptar herramientas para la especificación y validación de requisitos, tomando en consideración que el desarrollo se realiza utilizando una metodología ágil.

Asimismo, presenta un marco teórico y metodológico aplicado a un caso de estudio, métricas y variables tomadas en cuenta antes y después de formalizar y estandarizar la especificación de los requisitos. Finalmente, conclusiones y trabajo futuro.

Marco Teórico

Para contextualizar al lector en el ámbito del estudio, se presentan algunas definiciones:

La Ingeniería de Requisitos de acuerdo con Roger s. Pressman (Pressman ,2015) se refiere a un proceso de especificación que permite dar respuesta a las necesidades del cliente cumpliendo con sus expectativas. En este sentido, identifica siete actividades fundamentales: inicio, elicitación, elaboración, negociación, especificación, validación y administración de requisitos. Otro reconocido autor, Ian Sommerville en (Sommerville, 2005), define la IR como el proceso de desarrollar una especificación del software, la cual tiene el propósito de servir como puente entre las necesidades del sistema del cliente y los desarrolladores. Describe la IR en 4 pasos, estos comprenden el estudio de viabilidad, la obtención y análisis, especificación y validación.

En el mismo orden, la acepción de IR según IEEE hace referencia a “las actividades en la cual se intenta comprender las necesidades exactas de los usuarios del sistema, para traducir éstas en instrucciones precisas y no ambiguas las cuales podrían ser, posteriormente, utilizadas en el desarrollo del sistema” (IEEE,1999).

El término **requisito**, según lo planteado por la ISO 9000 del año 2000, refiere a “la necesidad o expectativa establecida, generalmente implícita u obligatoria” (ISO, 2000). Asimismo, el glosario estándar de la terminología en Ingeniería del Software de la IEEE define requisito como la “Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. “(IEEE: Standard Glossary of Software Engineering Terminology). Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal” (IEEE, 1990).

De la misma manera, el SEI (Software Engineering Institute) plantea en el glosario del modelo de madurez CMMI que un requisito es: (1) Una condición o capacidad necesitada por un usuario para solucionar un problema o lograr un objetivo. (2) Una condición o capacidad que debe cumplir o poseer un producto o componente de producto para satisfacer un contrato, un estándar, una especificación u otros documentos impuestos formalmente. (3) Una representación documentada de una condición o capacidad como en (1) o en (2) (SEI, 2010).

Clasificación de requisitos

De acuerdo con Sommerville (Sommerville ,2005), los requisitos se categorizan en:

- Requisitos del usuario: son declaraciones, en lenguaje natural y en diagramas, de los servicios que se espera que el sistema proporcione y de las restricciones bajo las cuales debe funcionar.
- Requisitos del sistema: establecen con detalle las funciones, servicios y restricciones operativas del sistema.

Para la Software Engineering Body of Knowledge (SWEBOK), tal como se presenta en la Figura 1, existen dos grandes categorías para clasificar a los requisitos (ISO/IEC, 2005a), estas son:

- Requisitos Funcionales: especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física.
- Requisitos No Funcionales: son propiedades o cualidades que el producto debe tener, también son conocidos como atributos de calidad. Específicamente, aquellas propiedades o características que hacen al producto atractivo, usable, rápido o confiable; normalmente están vinculados a los requisitos funcionales.

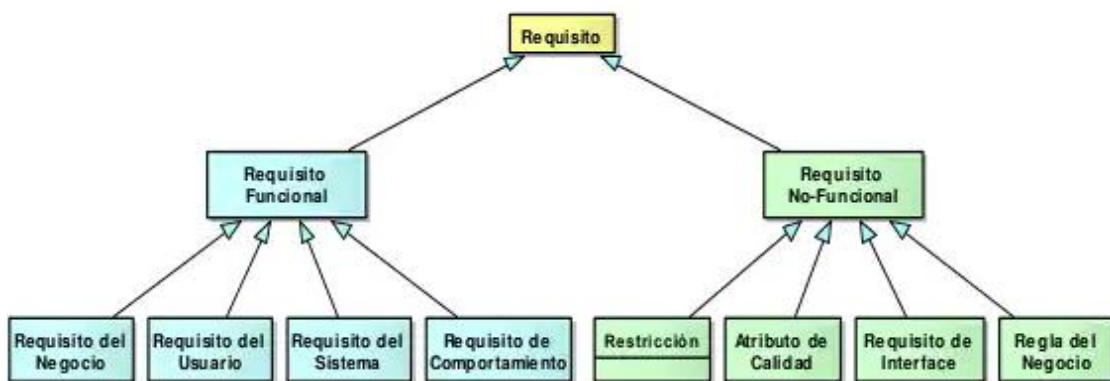


Figura 1. Clasificación de los Requisitos, SWEBOK. 2005

De acuerdo con Niño, Y. y Silega, N. (2018), entre las tres fuentes principales de requisitos de seguridad, una fuente se deriva de evaluar los riesgos para la organización, tomando en cuenta la estrategia general y los objetivos de esta. Otra fuente son los requisitos legales, reguladores, estatutarios y contractuales que tienen que satisfacer una organización, sus socios comerciales, contratistas y proveedores de servicio; y su ambiente sociocultural. Y una tercera es el conjunto particular de principios, objetivos y requisitos comerciales para el procesamiento de la información que una organización ha desarrollado para sostener sus operaciones (ISO/IEC, 2005b), (Mentor, 2017).

Ciclo de Vida de los Requisitos

En la Ingeniería de Requisitos se incluyen cuatro actividades de alto nivel según SWEBOK V3 (Bourque y Farley, 2014).

- **Educción:** En esta actividad se interactúa con los stakeholders para descubrir sus requisitos, también los requisitos del dominio y documentación existente.
- **Análisis:** En esta actividad se toma la recolección de requisitos, se clasifican los requisitos relacionados, se definen los servicios que debe proporcionar el sistema, desempeño, requerido, restricciones de hardware, etc.
- **Especificación:** Los requisitos se documentan y se comprueba la consistencia, integridad, verificabilidad, comprensibilidad, rastreabilidad y adaptabilidad de los requisitos.
- **Validación:** Debido a que en la actividad anterior intervienen diversos participantes, los requisitos entran en conflicto. Esta actividad se preocupa de validar y verificar que el ingeniero de software ha entendido los requisitos.

La especificación de requisitos se concentra en formatos o plantillas que son utilizados por las empresas desarrolladoras para documentar sus requisitos. En un estudio realizado por la organización para el Organización para la Cooperación y el Desarrollo Económicos, en adelante OECD, las empresas desarrolladoras de software de países latinoamericanos, durante la segunda mitad del 2015 y principios del 2016,

específicamente en: Chile, Argentina, Perú, Brasil, Colombia, Ecuador, Costa Rica y México; se obtuvo que un 33.3% de los participantes crearon un formato propio; el 13.3% utilizan plantillas de historias de usuarios; un 6.8% utilizan plantillas de casos de uso; un 13.3% utiliza un documento de especificación de requisitos; el 13.3% reconoce no utilizar ningún formato o plantilla. Cabe agregar que un 10% utiliza alguna herramienta para la especificación de requisitos y otro 10% no respondió la pregunta. (Carrizo y Rojas, 2016)

Esta situación reportada, indica que existe una clara ausencia de dedicación en la especificación de requisitos, destinándose poco tiempo en la descripción funcional y vista de la arquitectura o vista física de la solución. En diversas ocasiones, una vez que un proyecto se ha terminado y cuando se cree que se cumplió con todo lo que debe hacerse, según la experiencia y lo exigido por el cliente que lo solicitó, los usuarios más de una vez se enfrentan al hecho de que el resultado logrado no es lo que ellos esperaban. La historia está llena de proyectos fracasados por incumplimiento de plazos y presupuestos, con funcionalidades que no cubren las necesidades de los usuarios, con el consecuente impacto en los negocios de las organizaciones.

Técnicas para la especificación de requisitos

Las estrategias recomendadas para la especificación de los requisitos del software están descritas por la norma IEEE 830-1998. Este estándar describe las estructuras posibles, contenido deseable, y calidades de una especificación de requisitos del software

Existe una amplia variedad de técnicas y herramientas para la especificación de los requisitos de un sistema. En este caso particular se seleccionó un subconjunto a saber:

1. Plantillas para el Documento de Especificación de Requisitos adaptado a partir del material que provee el Marco de Desarrollo de La Junta de Andalucía España.
2. Diagramas de Modelado de Procesos del Negocio BPMN

3. Escenarios de Casos de Uso

Marco Metodológico

El estudio de caso

Los estudios de caso son métodos observacionales. Se basan en la observación de una actividad o proyecto durante su curso. Son utilizados para monitorear proyectos o actividades y para investigar entidades o fenómenos en un período específico. El nivel de control de la ejecución es menor en los estudios de casos que en los experimentos. Esto se debe principalmente a que en los estudios de caso no se controla, sólo se observa, contrario a lo que ocurre en los experimentos.

Los estudios de caso son muy útiles en el área de Ingeniería de Software, se usan en la evaluación industrial de métodos y herramientas. Además, son fáciles de planificar, aunque los resultados son difíciles de generalizar y comprender. Los estudios de caso no manipulan las variables, sino que éstas son determinadas por la situación que se está investigando. Al igual que las encuestas, los estudios de caso pueden ser clasificados como cualitativos o cuantitativos dependiendo de lo que se quiera investigar del proyecto en curso. A continuación, se hará una breve descripción de las situaciones observadas.



Figura 2: Métodos de Observación. Fuente: Barbosa J. (2013)

Metodología de Desarrollo de Software: SCRUM.

Proceso Observado: Desarrollo de componentes de software del Módulo Célula 10. Encargado de la planificación de visitas a los clientes y cobranza, por el servicio de instalaciones nuevas y/o modificaciones de los servicios: conexión a internet, televisión por cable y telefonía fija, siendo estos todos los servicios prestados.

Problemática actual: Inconformidad del cliente, dueño del producto de software, con los productos presentados, retrabajo en los productos desarrollados por mal funcionamiento, errores en la operación de los módulos por falta de documentación, frustración en el grupo de desarrollo por rechazo de los productos por parte del cliente, excesiva cantidad de horas extras de los programadores por rechazo de productos.

Objetivo: Determinar el impacto del uso de técnicas para la especificación de requisitos, desde tres aspectos: especificación adecuada de requisitos, entregas aceptadas sin modificación (número de revisiones del programador por incidencias), tiempo de desarrollo medido en horas de trabajo.

Resultados

Contexto del Caso de Estudio: célula 10 del proyecto agendamiento

Ubicación: Una empresa del sector de comercialización de servicios de telecomunicaciones en Bogotá, Colombia.

Descripción del proceso de agendamiento

El proceso inicia con la venta del producto (cualquier combinación de los descriptos previamente), esta venta puede ser vía teléfono (audio-contrato) o de forma presencial en los centros de atención y ventas (contrato impreso), el asesor que realiza la venta debe validar que en la zona donde se hará la instalación hay cobertura, esto lo hace colocando la dirección exacta del lugar, esta dirección se valida contra, un subsistema que dada un dirección, devuelve las coordenadas (x, y) de la ubicación, y un campo llamado *address_id*, que es el identificador único para esa dirección, si la dirección es un conjunto residencial o un edificio, el sistema toma como referencia la entrada principal y el nodo de conexión más cercano al lugar.

El tipo de producto que se desea instalar, se debe segmentar en tipo de red, tipo de trabajo y subtipo de trabajo, teniendo estos tres datos e incluyendo el nodo, se lista en un calendario, los próximos 7 días, mostrando la disponibilidad para realizar la instalación, es importante que la disponibilidad tenga esta información: franja del día (hora de inicio y hora de fin), día (fecha) y cupos disponibles (cantidad de técnicos para realizar la instalación).

Una vez seleccionado el día y la franja, se procederá al agendamiento, se debe enviar los datos del cliente, los datos de la dirección, los datos de capacidad (día, franja y disponibilidad), los datos del producto y observaciones, el sistema deberá asignar la cantidad de técnicos requeridos en base a la duración del tipo de trabajo y este será restado de la capacidad disponible, un técnico solo puede trabajar 9 horas al día, si la instalación de un producto requiere varios días, el sistema deberá tomar en cuenta el máximo de horas disponible del técnico.

En el proceso de agendamiento se incluye un número único, denominado número de orden, un cliente puede tener varios números de orden asignado, para instalaciones en

el mismo lugar o en diferentes lugares, en adelante diremos agendamiento u orden, refiriendo al conjunto de información descrita en el párrafo anterior.

Un agendamiento puede ser actualizado, pero no se podrá cambiar ni el cliente, ni el número de orden.

Un agendamiento se puede modificar una vez realizado por cambio de fecha y hora de la instalación, para esto se debe especificar: número de orden y el motivo de acuerdo a las causas desplegadas en el sistema. Un agendamiento se puede cancelar, aquí solo se cambia el estado de la orden a “cancelado”, para esto se debe especificar: número de orden y el motivo y el motivo de acuerdo a las causas desplegadas en el sistema.

Los estados de la orden o agendamiento son: agendada, reagendada y cancelada, una vez que sale del estado agendada no podrá volver a estar en ese estado. Toda la información de la orden debe estar guardada en base de datos, no se puede modificar salvo por el método actualizar, o que este cancelada y se agenda nuevamente, para eso se validará que el número de la orden ya existe o no, en caso de que no exista crea una orden nueva, en caso de que exista, actualiza la información de la orden existen y su estado será reagendado.

Los métodos reagendar y cancelar, solo modifican el estado de la orden.

Captura de requisitos del proyecto

En el corazón de SCRUM, el *backlog* o pila del producto consiste en la lista de los requisitos del producto, los cuales no están en excesivo detalle, pero sí deben estar priorizados. Este *backlog* o “pila del producto está en constante evolución y abierta a todos los roles” (Pérez, 2011), pero es el propietario o cliente del producto el responsable y quien decide sobre está, bien sea para agregar, modificar o eliminar elementos de la lista.

Lo anterior no ocurre necesariamente en la práctica, por lo general la priorización de los requisitos cuando está en manos del cliente, y éste no es un experto en el área del

software o no cuenta con las herramientas para transmitir de forma correcta los requerimientos o requisitos del producto a elaborar, trae consecuencias negativas en los tiempos de desarrollo. Todo ello sin contar con que no siempre dispondrá del tiempo suficiente para dedicarle a su intervención en el desarrollo del producto. (Hernández, 2019)

Situación previa a la aplicación de técnicas de especificación de requisitos

Como se mencionó anteriormente, SCRUM hace énfasis en documentación “blanda”, es decir sin excesivo detalle y la constante interacción del equipo con el cliente, sin embargo, en el caso de estudio se observó cómo estos dos aspectos, aumentaron los costos de producción. A continuación, las causas detectadas:

1. Documentación blanda: tiene como objetivo evitar la excesiva documentación, pero en contra, tiende a omitir detalles que pudieran resultar importantes causando un retrabajo al detectar el efecto de las omisiones al momento del desarrollo.
2. Prioridad asignada por el cliente: si bien es cierto que el cliente debe ser parte fundamental de todo el proceso de desarrollo, existen casos en los que el cliente no tiene el conocimiento necesario para establecer prioridades de desarrollo. Ciertamente, el cliente conoce las prioridades del negocio pero pudiera estar omitiendo las prioridades establecidas por el desarrollo como por ejemplo, procesos de codificación complejos o la interacción entre los requisitos. Una prioridad mal establecida trae costos extras atribuibles a la reorganización de las prioridades.
3. Trabajo directo sobre el producto: si bien ayuda a avanzar en el desarrollo sobre el producto final, pudiera suceder que, al modificar un requisito de software, el impacto sobre el proyecto completo sea difícil de estimar hasta realizar la validación correspondiente. Si el impacto es mínimo, trae como consecuencia el costo extra de cambiar el requerimiento, pero a medida que aumenta el impacto, estos costos se van sumando al proyecto.

Como consecuencia de los tres aspectos anteriormente mencionados y su impacto directo en el costo de los proyectos que lleva la célula 10, se recolectó un conjunto de datos que ilustran el comportamiento detectado.

- La célula 10 está compuesta por 8 desarrolladores, en este ejercicio se asignará un costo por hora de programación 10\$/programador (precio ficticio), 2 personas encargadas de hacer las pruebas y validaciones, en adelante QA (Aseguramiento de la calidad), para ellos supondremos un costo por hora de 7\$/probador y un líder técnico para el cual se asignará un costo de 15\$ la hora.
- El número de requerimientos recibidos en los 5 últimos sprint, son 57 en total. Cada requerimiento consume un promedio de 7 horas de desarrollo y 2 horas de pruebas.

A continuación, en la Tabla 1 se identifican los costos asignados.

Tabla 1. Relación de costos dólares/hora de los últimos 5 sprint

Número de requisitos	Tiempo Total dedicado por desarrollador (horas)	Tiempo Total dedicado por probador QA (horas)	Costo Total por hora de desarrollo (\$/hora)	Costo Total por hora de probador QA (\$/hora)
57	399	114	3900	798\$

Fuente: Elaboración propia

De los 57 requerimientos, el equipo de QA rechazó 23 por que no cumplían con lo esperado (el desarrollador entendió algo diferente al QA, o bien no cumplió con los requerimientos solicitados), 10 tuvieron que verificarse con el cliente por que habían surgido dudas sobre su funcionamiento, y solo 24 recibieron el visto bueno. Sin embargo, de los 24 aceptados por el cliente, este último rechazó 4 por que no se ajustaban a su solicitud. Es decir, de los 57 requerimientos solo 20 fueron aceptados sin modificaciones mientras que 37 tuvieron algún tipo de revisión.

A continuación, la Tabla 2 presenta el impacto del rechazo de los 37 requisitos y sus costos.

Tabla 2. Relación de costos dólares/hora de los últimos 5 sprint para los 37 requisitos rechazados

Número de requisitos rechazados	Tiempo Total dedicado por desarrollador (horas)	Tiempo Total dedicado por probador QA (horas)	Costo Total por hora de desarrollo (\$/hora)	Costo Total por hora de probador QA (\$/hora)
37	148 horas	74 horas	1480 \$	528 \$

Fuente: Elaboración propia

Es de hacer notar, que el costo de revisión supera el 40% del costo total del producto, sin incluir el costo del líder técnico ni del cliente.

Asimismo, se observa que el modelado de los procesos de negocio, no está al alcance de los desarrolladores, quienes solamente conocen de las interrelaciones y flujo de los procesos por ensayo y error durante la programación. Lo mismo ocurre con la arquitectura del software, se reconoce el uso de un modelo de módulos no fuertemente relacionados, para el que cada módulo del software cumple una función específica del negocio y se dividen en dos sub módulos: EAP y EAF. EAP integra los servicios REST y EAF se encarga de consumir los servicios y mostrar la interfaz de usuario, configurando de esta manera una arquitectura de micro servicios. En cuanto a las pruebas, si bien cada desarrollador debe aplicar un conjunto de pruebas, no hay una política clara sobre las mismas, ni se genera ningún tipo de evidencias, salvo las de QA, que se entregan al cliente.

Situación con la aplicación de técnicas de especificación de requisitos

En esta sección se muestran los cambios, producto de la aplicación de técnicas en la especiación de requisitos, y los resultados sobre el desempeño de la célula.

La especificación de requisitos se hizo en tres etapas:

1. Reunión para la concertación de requisitos: se realiza una reunión donde el cliente manifiesta los requisitos que desea sean abordados por la célula, si bien esta reunión siempre había sido realizada, la mejora fundamental consistió en

que los desarrolladores registraron a través de notas sus observaciones y al cliente se le requirió el envío de sus requisitos a través del correo con las especificaciones correspondientes para la posterior trazabilidad.

2. Descripción escrita y detallada de los casos de prueba: los programadores hacen una lista de casos de prueba con las variantes posibles. Esto permitió probar casos extremos o flujos de comportamiento, este punto fue totalmente nuevo.
3. Doble validación de los requisitos: una vez el programador establece los puntos a ser evaluados, se realiza una nueva reunión para validar la correcta interpretación de los requisitos planteados, este punto es totalmente nuevo.

Aplicando estas tres etapas, se obtuvieron los resultados mostrados en la Tabla 3:

Tabla 3. Resultados en tiempos de ejecución y requerimientos, aplicando las técnicas de captura de requisitos

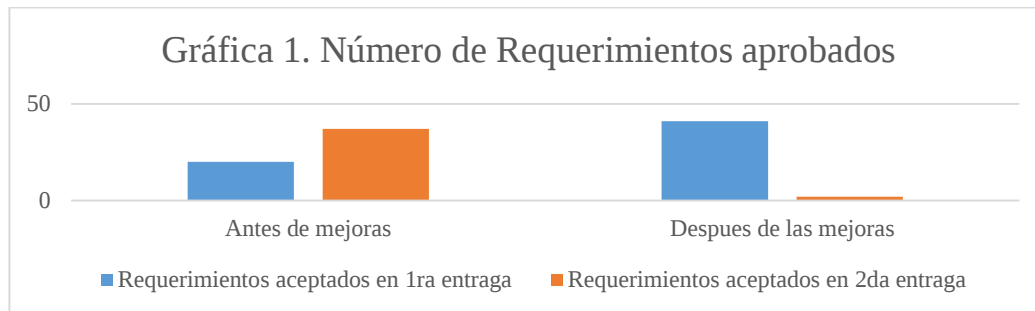
Número de requisitos	Número de requisitos aceptados en 1ra entrega	Tiempo Total dedicado por desarrollador (horas) 1ra entrega	Número de requisitos aceptados en 2da entrega	Tiempo Total dedicado por desarrollador (horas) 2da Entrega
43	41	258	2	9

Fuente: Elaboración propia

Como se puede observar, hubo una aceptación del 95% de las entregas, lo cual repercutió directamente sobre los costos del proyecto. En este sentido, comparando los datos observados antes de las mejoras, donde las revisiones representaban un 40% de los costes del proyecto, se puede apreciar una dramática disminución a tan solo 3,4% del coste, es decir, una reducción del 91.5%. Indudablemente, el impacto altamente positivo, demuestra la efectividad de las técnicas aplicadas y urge la necesidad de la aplicación de las técnicas implantadas, sobre el resto de los proyectos. Cabe mencionar, que otro aspecto impactado positivamente fue la trazabilidad de los requerimientos a partir de la documentación levantada. De los 37 requisitos correctamente documentados en todas las fases: descripción, desarrollo, pruebas unitarias, pruebas QA y entrega al cliente; se cuantifica una mejora del 97% en

cumplimiento de criterios de aceptación, en base a la misma cantidad observada en los Sprint anteriores.

La Gráfica 1 a continuación muestra la efectividad de la aplicación de las técnicas de levantamiento de requisitos en dos sprints, medida a partir de la aceptación y/o rechazo de los requisitos.



Fuente: Elaboración propia

Conclusiones

Con base en la documentación descrita en este trabajo, más la evidencia empírica del trabajo de campo realizado, se observa claramente, que las técnicas de ingeniería de software en especial las de captura de requisitos, aun aplicando las más básicas, conllevan a una mejora sustancial no solo en los tiempos de entrega sino también en los costos del proyecto, mejorando de forma significativa la experiencia del equipo de desarrollo, del cliente y otros actores involucrados.

Las empresas de desarrollo de software deben mantener una investigación continua alrededor de mecanismos que les permitan aumentar la confiabilidad de sus entregas, disminuir los riesgos y los sobrecostos en el proceso de desarrollo. La investigación se debe orientar al uso de nuevas técnicas y enfoques que fortalezcan características tales como la agilidad en el tratamiento de los requisitos, la disminución de los conflictos entre los participantes, el reconocimiento oportuno de los errores o problemas en la identificación y especificación de los requisitos y el establecimiento de controles en su evolución durante las diferentes fases del ciclo de vida.

Para finalizar y como parte de la reflexión, se elaboró un inventario de las principales causas que afectaron al proyecto y que produjeron fallas con sus consecuentes costes en el mismo. Algunas de ellas son: ♣ Los clientes no están obligados a tener una comprensión clara de los requisitos del sistema, incluyendo el alcance del mismo, las principales características funcionales y los atributos no-funcionales. ♣ Las necesidades de los usuarios cambian constantemente. ♣ Los ingenieros de software no tienen suficiente acceso al conocimiento y a la experiencia del dominio. ♣ El cronograma del proyecto es demasiado estrecho como para permitir una adecuada interacción y un periodo de aprendizaje entre el cliente y el equipo de desarrollo. ♣ Los líderes de proyectos no tienen el tiempo para apropiarse de conocimientos técnicos y del dominio. ♣ Faltan vínculos de comunicación entre clientes, analistas y desarrolladores. ♣ Falta un dominio estandarizado de definición de datos y de interfaz del sistema y del entorno.

Con base en los resultados, se sugieren las siguientes mejoras a las prácticas actuales de Ingeniería de Requisitos: ♣ Mejorar el proceso de gestión de proyectos para facilitar la comunicación, la documentación y el control, así como la gestión de cambios. ♣ El conocimiento del dominio y los prototipos son elementos necesarios para el éxito de la práctica de la IR. ♣ Hacer que el cliente sienta que los requisitos y el sistema futuro son de su propiedad y por lo tanto su responsabilidad. ♣ Ser proactivo en los procesos de IR y ser capaz de predecir posibles cambios y necesidades futuras. ♣ Enlazar los requisitos con las pruebas y adoptar un proceso de diseño basado en pruebas. ♣ Aun cuando el tiempo es escaso, debe estar claro qué desarrollar antes de cómo desarrollarlo. ♣ Desarrollar herramientas de IR que se ajusten mejor a las necesidades reales de los clientes y de los ingenieros.

Un modelo de la actividad de especificación no debería enfocarse solamente en mostrar los pasos secuenciales de aplicar una técnica en particular, sino que debería permitir visualizar un proceso desde su contexto, cuya dimensión no es lineal y ofrecer una guía práctica suficientemente clara y detallada para sistematizarlo. Este

último punto enfatiza un problema importante de la IR: existe una gran diferencia entre lo que los investigadores proponen o recomiendan, y lo que realmente aplican los analistas que practican la especificación y documentación en su trabajo, quienes generalmente utilizan la experiencia personal y las técnicas que les son familiares o que les impone una metodología adoptada.

Esta brecha se debe, en parte, a la falta de investigación empírica suficiente que permita a la IR obtener retroalimentación positiva, desde la industria del desarrollo de software y de la transferencia de las propuestas metodológicas novedosas que surgen de la investigación.

A partir de este análisis, se considera de interés citar nuevamente el problema abierto que aborda este artículo, recordando que el mismo se focaliza en que la práctica de procesos de captura de requisitos, adecuados para la organización, generalmente representa una mejora en el proceso de desarrollo, en los costos y en la satisfacción del usuario, los desarrolladores, el dueño del producto, etc.

Referencias

Sommerville, I. (2005). Ingeniería del software. 7ma Edición. United Kingdom: Pearson Education

Roger S. Pressman, B. R. M. (2015). Ingeniería de software. Un enfoque práctico. 8ª edición. New York.

IEEE. (1990). IEEE Terminología estándar de ingeniería de software.

Revista Cubana de Ciencias Informáticas Vol. 12, No. Especial UCIENCIA, Septiembre 2018 ISSN: 2227-1899 | RNPS: 2301 Pág. 205-221 <http://rcci.uci.cu>

SEI. (2010). CMMI® para Desarrollo, Version 1.3: Carnegie Mellon University.

ISO. (2005). ISO/IEC 17799 Tecnologías de la información - Técnicas de seguridad - Código para la práctica de la gestión de la seguridad de la información

ISO. (2000). ISO 9000:2000 Sistemas de gestión de la calidad — Conceptos y vocabulario.

ISO/IEC. (2005a). Ingeniería de Software - Guía del Cuerpo de Conocimiento de Ingeniería de Software (SWEBOK).

ISO/IEC. (2005b). ISO/IEC 27001 Tecnología de la información - Técnicas de seguridad - Sistemas de gestión de seguridad de la información - Requisitos: Switzerland.

IEEE(1999). Std 830-1998 IEEE “Recommended Practice for Software Requirements Specifications”Description. Institute of Electrical and Electronics Engineers. Noviembre 1999. Disponible en: <http://www.standards.ieee.org/findstds/standard/830-1998.html>

P. Bourque and R.E. Fairley. “Guide to the software engineering body of knowledge”. IEEE Computer Society Press. Version 3.0. 2014. ISBN: 0769551661 9780769551661

Carrizo, D y Rojas, J (2016). Práctica Industrial en Ingeniería de Requisitos. Un estudio empírico de empresas latinoamericanas. International Conference on Information Systems and Computer Science INCISCOS 2016.

Mentor, A. (Producer). (2017, marzo 12). Aula Mentor. Aula Mentor. Seguridad informática. Normas ISO sobre gestión de seguridad de la información. Retrieved from http://descargas.pntic.mec.es/mentor/visitas/demoSeguridadInformatica/normas_iso_sobre_gestin_de_seguridad_de_la_informacin.html

“Políticas de banda ancha para américa latina y el caribe: un manual para la economía digital.” OECD publishing, París, 2016.

Bournissen, J. M. (1999). La evolución del software, puntos de crisis (Parte II). Enfoques, 11(1 y 2), 130-140.

Pérez, O. A. (2011). Cuatro enfoques metodológicos para el desarrollo de Software RUP–MSF–XP–SCRUM. Inventum, 6(10), 64-78.

Lavalle, Y. F., & Tabares, C. Z. E. M. FUNDAMENTOS PARA LA GESTIÓN DEL REQUISITO NO FUNCIONAL ESCALABILIDAD DEL SOFTWARE VERSAT SARASOLA.

Bourque, P., & Fairley, R. E. (2014). SWEBOK v3. 0: Guide to the software engineering body of knowledge. IEEE Computer Society, 1-335.

Barrera, J. A. H. (2015). Integración de marcos de trabajo para desarrollo de software: SCRUM, PSP e ISO 25000 [Integrating software development frameworks: SCRUM, PSP and ISO25000]. Ventana Informática, (32).

Godoy, D. A., Belloni, E. A., Kotynski, H., Santos, H. D., & Sosa, E. O. (2014, October). Simulando proyectos de desarrollo de software administrado con SCRUM. In XVI Workshop de Investigadores en Ciencias de la Computación.

Hernández, G., Martínez, Á., Jiménez, R., & Jiménez, F. (2019). Scrum y Peopleware: elementos clave para la gestión en la construcción de software. Revista Ibérica de Sistemas e Tecnologías de Informação, (E19), 265-277.

Niño, Y. y Silega, N. (2018). Requisitos de Seguridad para aplicaciones web Security. Revista Cubana de Ciencias Informáticas Vol. 12, No. Especial UCIENCIA, pp. 205-221

copia del Acta de Aprobación del Proyecto



Ciudadano(a) Investigador(a)
Darwin Rocha, Mirella Herrera y Francisca Grimón
Presente

Nos complace informar a usted que el trabajo titulado «**ESPECIFICACIÓN DE REQUISITOS PARA REDUCIR COSTOS DE UN PROYECTO DE SOFTWARE. CASO: EMPRESA DE TELECOMUNICACIONES**», fue **APROBADO** luego de realizado el arbitraje doble ciego juicio de pares, para ser **presentado oralmente** en el **IV Congreso Nacional de Investigación e Innovación en Ciencias Económicas y Sociales “Hacia La Nueva Visión del Planeta”**, organizado por la Facultad de Ciencias Económicas y Sociales de la Universidad de Carabobo. Los detalles sobre la presentación oral de su trabajo a través de ZOOM (día, hora) le serán comunicados oportunamente por la Comisión Académica en fecha previa al Congreso.

El Congreso se estará realizando del 29 al 31 de marzo de este año 2022 y su lema **“Hacia La Nueva Visión del Planeta”** da continuidad a la voluntad manifiesta de nuestra gestión decanal y directiva por transitar los senderos de construcción de un mundo mejor, donde haya posibilidad de ver alcanzados los **17 Objetivos de Desarrollo Sostenible ODS de la Agenda 2030 de la Organización de Naciones Unidas**.

Le expresamos el agrado de la Universidad de Carabobo a través de su Facultad de Ciencias Económicas y Sociales y el Comité Organizador del Congreso, por contar con su participación y le recordamos que este trabajo, previa revisión y aprobación del Comité de Publicaciones y realizados de su parte los ajustes necesarios recomendados por los evaluadores, podrá formar parte de los textos digitales editados por el Comité Editorial del congreso.

En Valencia a los 8 días del mes de marzo de 2022.

Atentamente,



Dr. Williams Aranguren
Coordinador Comisión de Arbitraje