



**TECNOLÓGICO NACIONAL
DE MÉXICO**
Instituto Tecnológico de Apizaco

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

**“ESTRATEGIAS DE MEJORA PARA LA ADMINISTRACIÓN DE
REQUERIMIENTOS”**

TESIS

PARA OBTENER EL GRADO DE:

MAESTRO EN SISTEMAS COMPUTACIONALES

PRESENTA:

BEATRÍZ FERNÁNDEZ TAMAYO

DIRIGIDO POR:

Director:

M. en C. JOSÉ JUAN HERNÁNDEZ MORA

Co-Director:

M. en C. MARÍA GUADALUPE MEDINA BARRERA

APIZACO, TLAX.

AGO 2018.

ESTRATEGIAS DE MEJORA PARA LA ADMINISTRACION DE REQUERIMIENTOS

Tesis presentada por Instituto Tecnológico de Apizaco.

Para obtener el grado:

Maestro en sistemas computacionales

Presenta:

Beatriz Fernández Tamayo

Dirigido por

Director:

M. en C. José Juan Hernández Mora

Co-Director:

M. en C. María Guadalupe Medina Barrera

Tutor:

M. en C. Juan Ramos Ramos

Revisor:

M.C. en C. María Janáí Sánchez Hernández

APIZACO, TLAX

AGO 2018

TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE APIZACO

“ESTRATEGIAS DE MEJORA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS”

Aprobado por:

Director de tesis

Co-director

Tutor

Revisor

Índice

Capítulo 1	Introducción	9
	1.1 Retos.....	10
	1.2 Definición del Problema.....	10
	1.3 Hipótesis.....	11
	1.4 Objetivo General.....	11
	1.5 Objetivos Específicos.....	11
	1.6 Alcances y Limitaciones.....	12
	1.7 Contribuciones Esperadas.....	12
	1.8 Propuesta de Solución.....	12
	1.9 Trabajos Relacionados	12
	1.9.1 Ingeniería de Software.....	12
	1.9.2 Ingeniería de requerimientos.....	13
	1.9.3 Diseño centrado en el usuario.....	16
	1.9.4 Requerimientos Basados en Experiencia de usuario.....	17
Capítulo 2	Ingeniería de Requerimientos con Enfoque en el Diseño Centrado en el Usuario.	19
	2.1 Requerimientos.....	19
	2.2 Diseño Centrado en el Usuario.....	22
	2.2.1 Metodología y Técnicas de DCU.....	23
	2.3 Prototipos.....	25
	2.4 Casos de uso.....	25
	2.5 Estándares de Calidad.....	28
Capítulo 3	Modelo de Recolección de Requerimientos con Enfoque el Usuario	33
	3.1 Gestión del Cambio.....	36
	3.2 Descripción de Procedimiento.....	37
	3.2.1 Asignación de Roles.....	37
	3.2.2 Recolección de requerimientos.....	39
	3.2.3 Desarrollo de Documentos Entregables y Establecimiento de Requerimientos.....	45
Capítulo 4	Sistema Gestor de Incidencias: Caso de estudio	49
	4.1 Recolección de requerimientos.....	49
	4.2 Desarrollo de Documento de Establecimiento de Requerimientos.....	56
	4.3 Prototipado.....	60
Capítulo 5	Resultados	65
Capítulo 6	Conclusiones y trabajos futuros	70
Referencias		73
Anexos		75

Índice de Figuras

Figura 1	Etapas de la fase de requerimientos.....	19
Figura 2	Clasificación de los requerimientos.....	20
Figura 3	Clasificación de requerimientos funcionales.....	20
Figura 4	Actividades relacionadas con la ingeniería de requerimientos.....	21
Figura 5	Proceso del diseño centrado en el usuario.....	22
Figura 6	Historias vividas sobre una interacción exitosa con la aplicación.....	24
Figura 7	Comportamiento de los usuarios finales.....	24
Figura 8	Plantilla de detalle de casos de uso.....	26
Figura 9	Elementos de un diagrama de casos de uso.....	27
Figura 10	Ejemplo de una relación de inclusión.....	27
Figura 11	Ejemplo de una relación de extensión.....	27
Figura 12	Clasificación CMMI.....	29
Figura 13	Niveles de Madurez CMMI-Dev.....	30
Figura 14	Modelo Recolección de requerimientos centrado en el usuario.....	34
Figura 15	Modelo de recolección de requerimientos centrado en el usuario en.... diagrama BPMN.	35
Figura 16	Gestión del cambio.....	36
Figura 17	Roles necesarios en fábrica de software.....	37
Figura 18	Distribución de roles para el desarrollo.....	38
Figura 19	Portada del documento de requerimientos.....	45
Figura 20	Información del documento.....	45
Figura 21	Ejemplo listado de la funcionalidad del sistema.....	46
Figura 22	Ejemplo de un diagrama de casos de uso.....	47
Figura 23	Pantalla inicial SpiraTest.....	48
Figura 24	Test inicial del usuario de GPR.....	51
Figura 25	Test inicial del usuario de QA.....	52
Figura 26	Sketch realizado por los usuarios registro de incidencia.....	53
Figura 27	Sketch realizado por los usuarios seguimiento de incidencia.....	53
Figura 28	Diagrama de casos de uso de SAI.....	59
Figura 29	Diagrama BPMN de SAI.....	59
Figura 30	Matriz de trazabilidad de requerimientos básica con datos exportados... desde SpiraTest	60
Figura 31	Lista de requerimientos generada en la plataforma SpiraTest.....	60
Figura 32	Pantalla de seguimiento de requerimientos de SpiraTest.....	61
Figura 33	Reporte detallado de requerimientos.....	61
Figura 34	Prototipo de pantalla de registro.....	62
Figura 35	Prototipo de pantalla de registro de incidencia.....	62
Figura 36	Prototipo de pantalla Registro de incidencia.....	62
Figura 37	Prototipo de pantalla de reportes.....	62
Figura 38	Prototipo de pantalla de búsqueda.....	62

Índice de Tablas

Tabla 1	Simbología.....	34
Tabla 2	Ejemplo para describir detalladamente un caso de uso	47
Tabla 3	Resultados de las pruebas de mapeo sobre pantalla blanca.....	54
Tabla 4	Listado de funcionalidad.....	54
Tabla 5	Descripción detallada de casos de uso de SAI.....	58
Tabla 6	Registro recuperado del reporte de validación de la empresa.....	63
Tabla 7	Observaciones de resultados.....	65
Tabla 8	Comparación de tiempo invertido con el modelo propuesto vs otros.....	68
	modelos en proyectos pasados	
Tabla 9	Relación de correcciones requeridas por actividad.....	67

Índice de Gráficas

Grafica 1	Comparación de tiempo invertido con el modelo propuesto vs otros.....	68
	modelos en proyectos pasados.	
Grafica 2	Demostrativa de decremento de correcciones por actividad.....	69

Definiciones, Abreviaturas y Acrónimos

AI	Arquitectura de Información
BPMN	Business Process Model and Notation
CMMI	Capability Maturity Model Integration
DCU	Diseño Centrado en el Usuario
DMS	Desarrollo y Mantenimiento de Software
FS	Fabrica de software
UI	Diseñador de Interfaces
UML	Lenguaje Unificado de Modelado
UX	Experiencia de Usuario
ELICITACIÓN	Del latín elicitus, "inducido" y elicere, "atrapar", es un término asociado a la psicología que se refiere al traspaso de información de forma fluida de un ser humano a otro por medio del lenguaje.
PROTOTIPO	Modelos desechables elaborados específicamente para la evaluación de las decisiones de diseño.

Resumen

Este proyecto reúne varias estrategias de administración de requisitos en un modelo de desarrollo que facilita el trabajo del equipo que crea el software, obteniendo un índice alto de información con un esfuerzo mínimo en el establecimiento de requisitos.

Utilizando herramientas de manejo de información como *SpiraTest* para automatizar varios artefactos usando el establecimiento de requisitos, modelado de información con lenguaje UML como *yEdGraphics* y de desarrollo de prototipos como *Balsamiq Mockups*, todo esto apoyó el análisis inteligente y la combinación de modelos de desarrollo enfocados en la mejora de usabilidad de proyectos de software con diseño y experiencia de usuario.

Para este trabajo, fue necesario implementar la metodología ágil para tener un entorno en el que el usuario esté presente en todo momento, para la aplicación de las pruebas y el diseño de los prototipos.

Como resultado de la prueba piloto de la propuesta en un proyecto pequeño, hubo una disminución en las correcciones que no fueron de tipo evolutivo, así como un mayor índice de seguridad en el proyecto por parte de los principales usuarios

Abstract

This project gathers several strategies of requirements administration in a development model that easy the job of the team that creates software, getting a higher index high of information with minimal effort in the requirements establishment.

Using driving tools of information as *SpiraTest* to automate several artifacts using in the requirements establishment, of information modeling whit UML language as *yEdGraphics* and of prototype development as *Balsamiq Mockups*, all this supported whit analysis and the combination of development models focused usability improvement of software projects whit design and user experience.

For this work, it was necessary to implement agile methodology to have an environment in which the user is present at all times, for the application of the tests and design of the prototypes.

As a result of piloted of the proposal in a small project, there was a decrease in corrections that were not of an evolutionary type, as well as a higher security index in the project by the main users.

Capítulo 1

Introducción

Última mente la industria del software ha visto un crecimiento exponencial en cuanto a demanda y tendencias nuevas sobre desarrollo de software que dan soluciones a los problemas de la crisis del software, diversas empresas de desarrollo buscan consolidarse en el mercado buscando un alto nivel de usabilidad en sus proyectos, no solo en cuanto a los resultados del producto sino también aumentar el nivel de usabilidad.

El primer gran problema tratado en la crisis del software es la inconsistencia de los requerimientos, precisamente los requerimientos son las bases de todo proyecto no solo de software, de maquinaria, muebles, dispositivos todo lo objeto creado tiene como objetivo satisfacer una o varias necesidades de algún usuario, de esta idea partimos para establecer que cada cosa que es creada debe ser diseñada para un alguien en específico, pero que pasa con aquellas ideas que no están bien fundamentadas en las necesidades de personas, por ejemplo una aplicación que debe realizar un proceso de facturación para sus proveedores, dicha aplicación es desarrollada y entregada al cliente pero cuando es puesto a prueba resulta que no hay ninguna forma de realizar un reporte sobre los movimientos de egresos, entonces resulta que si el cliente quisiera tener un historial por aclaraciones por ejemplo no podría generar un reporte automático, tendría que buscar y buscar hasta encontrar la información que requiere, entonces en realidad el proceso que ayuda en la contabilidad al usuario no está completo, se omitieron detalles importantes y no soluciona al 100% el problema.

Los problemas que vienen con los errores como el antes mencionado, pueden representar una inversión de tiempo extra y en los peores casos el replanteamiento de todo el proyecto; el mayor problema sobre el proceso de desarrollo de software se divide en dos partes, la primera es sobre el manejo e interpretación de la información, toda la información que se obtiene de las fases de establecimiento de requerimientos debe ser interpretada en características funcionales y visuales, la segunda parte es referente a toda la documentación y pasos que deben ser realizados como lo marcan las normas de calidad como CMMI y MoProsoft para la realización de proyectos de calidad.

El Instituto Nacional Tecnológico de la Comunicación (INTECO.2008) menciona que de acuerdo con el usuario el desarrollo de los mejores sistemas o software son aquellos que son creados por equipos de desarrolladores que definen desde el inicio del proyecto una redacción clara de qué pretenden lograr con dicho proyecto y el camino claro para conseguir tal fin. Es importante recalcar que una apropiada toma de requerimientos al inicio de un proyecto es la base para un resultado satisfactorio, en muchos casos se generan desfases de tiempo y demasiados cambios durante el ciclo de vida del mismo debido a un proceso pobre de toma de requisitos, contando con un apropiado procedimiento de recolección, análisis y administración de características no sólo aseguramos mejor calidad del producto final, permite al personal de la empresa un manejo más simplificado de documentos e incluso la

automatización de los mismos. La ingeniería de requerimientos centrados en el usuario resulta importante por la razón más lógica, cada sistema o software está destinado a ser un apoyo a un individuo o varios, es preciso crear de forma confiable y rápida sistemas que cumplan con características muy específicas y además sean escalables. Cabe resaltar el hecho de que es sumamente viable la creación de metodologías y técnicas para análisis de requerimientos que brinden usabilidad a nuestro sistema o software.

Entonces se tienen varios temas de especial interés, recolección de requerimientos, desarrollo de artefactos de establecimiento de requerimientos, diseño centrado en el usuario y SCRUM, este trabajo realiza un análisis de ellos y como realizar una combinación de ellos para brindar mayor nivel de calidad y usabilidad a los proyectos de software.

1.1 Retos

Los retos principales de este trabajo de investigación serán:

- 1.- La búsqueda de información y aprendizaje de técnicas adecuadas para la obtención de requerimientos.
- 2.- Entendimiento de diferentes maneras de comunicación que permitan llevar a cabo una interacción cómoda, así como constante para el cliente y analista de requerimientos.
- 3.- finalmente el desarrollo de técnicas que faciliten el levantamiento, comprensión, aprobación y administración de requerimientos de un proyecto de software.

1.2 Definición del Problema

Entre los principales problemas a los que se enfrentan los desarrolladores de software esta la comprensión cabal de las necesidades del cliente o solicitante de un proyecto, usualmente hay ciertas incógnitas que terminan por dañar el resultado o producto final. Como organización es importante la adaptación al ritmo de trabajo de sus clientes lo que en ocasiones trae problemas para llevar a cabo todas las prácticas establecidas en el plan de trabajo inicial, en ocasiones, dadas por falta de tiempo o algunos cambios que podrían darse dentro de la empresa o por parte del cliente, omisión total o parcial de características necesarias; cada usuario es distinto por ende vanean los problemas o escenarios y cada percepción de la solución será divergente.

Aun cuando los modelos de calidad definen procesos para la administración de requerimientos se presentan los siguientes problemas:

- Falta de habilidades del equipo en el levantamiento de requerimientos
- Las buenas prácticas para desarrollo de proyectos no son llevadas a cabo correctamente.
- Omisión de prácticas de validación de documentos.
- Cambios en requerimientos funcionales por parte del cliente (en todas las etapas del proyecto).
- Cambios imprevistos de líder de proyecto.
- Falta de comunicación con el cliente.
- Llevar a cabo los documentos entregables se torna tedioso para el equipo.

Este trabajo está enfocado al desarrollo e implementación de estrategias que mejoren la administración de requerimientos, dado que es precisamente aquí donde se cimienta todo el proyecto.

1.3 Hipótesis

¿Un desarrollo de estrategias de mejora centradas en el concepto de diseño centrado al usuario para la administración de requerimientos de proyectos de software brinda mayor calidad en los mismos?

1.4 Objetivo General

Diseño de un modelo de desarrollo que englobe técnicas de elicitación y especificación de requerimientos enfocados en la experiencia de usuario, para mejorar los productos de software en calidad y usabilidad, empleando técnicas de desarrollo centrado en el usuario.

1.5 Objetivos Específicos

- * Retomar las buenas prácticas y normativa de CMMI y Moprosoft.
- * Definición y capacitación del equipo adecuado para recolección y análisis de requerimientos.

- * Desarrollo de un modelo de desarrollo para mejorar el proceso de establecer requerimientos con las siguientes características:

- Aplicabilidad a distintos proyectos.
- Reducir cambios y prever impactos de los mismos.
- Mejorar administración de documentos.

1.6 Alcances y Limitaciones

Este trabajo cubrirá mejoras en el proceso de levantamiento y aprobación de requerimientos de proyectos de software. La principal limitante para el desarrollo y éxito de esta estrategia es la inmensa variabilidad de tipos de clientes, que podría o no llevar a cabo cabalmente el proceso.

1.7 Contribuciones Esperadas

Este trabajo cubrirá mejoras en el proceso de levantamiento - aprobación de requerimientos de proyectos de software. La principal limitante para el desarrollo y éxito de esta estrategia es la inmensa variabilidad de tipos de clientes, que podría o no llevar a cabo cabalmente el proceso.

1.8 Propuesta de Solución

Propuesta de solución es la implementación de un modelo de desarrollo “Modelo de Requerimientos centrados en el usuario” que este apegado a la normativa CMMI y Moprosoft y envuelva características de un diseño basado en la experiencia de usuario para obtener mejores y más útiles resultados en los proyectos futuros.

1.9 Trabajos Relacionados

1.9.1 Ingeniería de Software

1968 Garmich, Alemania durante una conferencia cuyo tema central fue exponer y dar solución al conjunto de problemas que enfrenta el desarrollo de software, surge el concepto “ingeniería de software”, en términos generales la ingeniería de software se refiere al estudio

de todas las formas en que puede ser desarrollado y dar mantenimiento un producto de software.

A continuación, algunas definiciones del concepto mencionado anteriormente:

“La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software”. (Institute of Electrical and Electronics Engineers [IEEE].1993).

“Es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas de software” (Zelkovitz, 1978).

“Es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software” (Bohem, 1976).

“Trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable, que sea fiable y trabaje en máquinas reales” (Bauer, 1973).

“La ingeniería de software se interesa por todos los aspectos de la producción del software, desde las primeras etapas como la especificación hasta el mantenimiento del sistema que se pone en operación” (Sommerville I. 2011).

La mayor parte de los conflictos que pertenecen a la llamada “crisis de software”, están relacionados directamente con los requerimientos del sistema, estos son todas aquellas características que necesita y/o debe tener un producto de software, para que al cumplir con ellas este ofrezca mayor calidad y posea un alto grado de usabilidad para el usuario.

Es por esta razón que, de la ingeniería de software se deriva el concepto “Ingeniería de requerimientos”, que encierra todo lo referente al tema de sobre requisitos, su definición, administración y desarrollo.

1.9.2 Ingeniería de Requerimientos

Como ya se mencionó anteriormente la ingeniería de software es una disciplina que se centra en todos y cada uno de los aspectos de la producción de software, cuyas actividades fundamentales son la especificación de requerimientos, desarrollo, validación y mantenimiento, es decir, la creación y distribución de actividades para la formación del software.

Todos los métodos referentes a requerimientos están enfocados en el sistema de división del problema en partes pequeñas que puedan ser resueltas de forma rápida y más simple, sin embargo debido a la relevancia que pueden tener los requerimientos en un sistema o proyecto de cualquier índole, fue viable crear una nueva sub-disciplina para el análisis puro de los

objetivos del usuario, el nacimiento de esta disciplina puede encontrarse en el año 1990 en los artículos de *Transactions on Software Engineering* (IEEE-TSE, 1991).

Los productos de software deben poder ser vendidos a uno o varios clientes, existen dos tipos, los primeros son productos genéricos, que consisten en sistemas independientes que se producen en una organización y son vendidos en el mercado, siendo accesibles a cualquier cliente, algunos ejemplos de estos son herramientas de administración de proyectos como Microsoft Project, paquetes de dibujo como Corel Drawn, sistemas de contabilidad etc. Los productos personalizados, son sistemas destinados para un cliente y necesidades muy específicas, un desarrollador o equipo realizan los programas con características específicas que solucionan problemas para ese usuario en especial.

En ambos casos el proceso de recolección de requerimientos resulta ser de lo más importante y donde debe centrarse la especial atención de los analistas e ingenieros de software ya que estos resultan ser el soporte de todo producto. Si los requerimientos no están correctamente entendidos o incluso enfocados la probabilidad del fracaso del proyecto es alta.

Como tal el proceso de toma de requerimientos es erróneamente manejado como una fase del ciclo de vida del software, los requisitos brindan información a todas las partes del software, y los descuidos de estos derivan en fallos catastróficos y hasta la completa inutilidad del producto.

El *Project Management Institute* (PMI.2014) refiere que el 47% del fracaso de los proyectos del software es causado por deficiencias en el ejercicio de la ingeniería de requerimientos y los casos más comunes son:

- * No se identificaron varios requerimientos.
- * El producto se entrega sin cumplir con los requerimientos necesarios.
- * La entrega final del producto no satisface al cliente, aunque este fue entregado a tiempo y está dentro del presupuesto.
- * El proyecto incorpora requerimientos que no deben estar en el alcance.
- * La estimación de costo/esfuerzo se hace en base a un alcance equivocado ya que no considera algunas áreas funcionales y procesos de negocio.
- * Fallas de comunicación sobre requisitos que derivan en cambios innecesarios debido a la falta de atención en comprender correctamente las necesidades del cliente al principio.

Entonces, si los requerimientos no están bien identificados en un proyecto y son identificados como fallas y no como cambios que den paso a la evolución, el equipo debe invertir más tiempo en corregir o incluso re plantear todo el proyecto esto conlleva a una decepción para el cliente, en el artículo *software defect radaction Top 10 list* (Boehm B. y Basili V. 2011) se indica que del 40% al 50% del esfuerzo en los proyectos de software se gastan en la corrección de defectos, pero ese no es el mayor problema, el verdadero conflicto llega cuando el equipo de desarrollo intenta corregir los defectos ya que es aquí donde existen de 20% a 50% más pasibilidades de introducir otro u otros errores en el sistema.

El problema principal es comprender los requisitos del cliente, para después conceptualizarlos y construirlos según las expectativas del usuario. Por lo tanto, la sugerencia es conectar al cliente con el proceso para aliviar los problemas en la fase de recopilación de requisitos para el software, para lograr un software de alta calidad (Geogy M. y Dharani D. 2016).

Software Engineering Body of Knowledge (SWEBOK), es el área de conocimientos de requerimientos que está a cargo del proceso de obtención, análisis, especificación y validación.

Sin embargo, las antes mencionadas disciplinas no son las únicas que pueden incluirse o emplearse en el proceso de requerimientos, recientemente en su tutorial de ingeniería de requerimientos Sommerville Ian (2016) califica la actividad de elicitación de requerimientos como fundamental para descubrir y comprender más afondo las necesidades del sistema.

Es preciso recalcar que hay dos tipos de cambios en un proyecto, los primeros se refieren a cambios que permiten la evolución del sistema, tienen un impacto mínimo benefactor y su ajuste sirve para aumentar la calidad y/o usabilidad del proyecto; pero por otro lado se encuentran los cambios que son generados por errores de especificación de requerimientos, pueden deberse a la exclusión de algunos requisitos o que estos estén mal entendidos. Siendo por esta razón que todo proceso de desarrollo debe proveer una buena gestión de cambios, ya que estos de manera inevitable surgirán.

En el artículo sobre retos del sistema de gran escala en la ingeniería de requisitos Safwat y Senousy (2015), se señalan varias limitaciones existentes en las prácticas de especificación de requerimientos para los desafíos en desarrollo de sistemas, pero en especial citaremos tres de las actividades que se mencionan como las más significativas para los últimos tiempos:

Análisis de interesados: Hace referencia a cualquier persona que este influenciando o influenciado por el desarrollo del sistema y puede utilizas el producto directa o directamente, se identifican de acuerdo al grado de influencia o interés sobre el proyecto.

Elicitación de requisitos: El proceso de revisar, documentar, analizar y comprender las necesidades y limitaciones de un sistema. El resultado es un artefacto llamado especificación de requerimientos que contiene toda la lista de requisitos textuales y sus descripciones, casos de uso, diagramas de proceso e interfaces y prototipos.

Priorización de requerimientos: Proceso de descubrir los requisitos que son más importantes ah desarrollar interactuando con las partes interesadas y organizándolas en mayor prioridad de orden.

Las nuevas técnicas descuidan algunas prácticas esenciales asociadas a la ingeniería de requerimientos de otros usuarios o interesados que se obtuvieron en juvias de ideas, talleres o métodos de entrevistas etc. Además, el análisis de impacto de cambios está ausente para analizar el efecto de los cambios en otros requisitos y cambios en la fase de desarrollo.

Kriesia, Blindheim, Bjelland, & Steinert (2016), proponen la utilización de *wayfaring* para encontrar las preguntas correctas, para que con las respuestas a esas preguntas se realice un refinamiento de requerimientos de forma iterativa. Describen las tendencias nuevas de desarrollo de productos organizacionales dirigidas por especificaciones e impulsadas por prototipos, la idea es que el primer prototipo este desarrollado bajo especificaciones predeterminadas y a partir del segundo caso las especificaciones están sujetas a cambios bajo influencia del aprendizaje de los prototipos anteriores; pero como formulamos esos requerimientos predeterminados, Mattmanna Gramlinch y Klobberdanza (2015) refiere que dentro de las consideraciones existentes en los criterios de calidad para la formulación estructurada y la documentación de los requisitos se centran en la completitud de los requerimientos durante el proceso de desarrollo de un producto de software; la efectividad y la eficiencia de los procesos de desarrollo de productos se ven influidas principalmente por una adquisición estructurada y sistemática de requisito y documentación de requisitos para formar una base confiable para todo el proceso de desarrollo y apoyar el desarrollo de productos optimizados en aplicaciones especiales.

1.9.3 Diseño centrado en el usuario

User Centered Desing o diseño centrado en el usuario (DCU) en español hace referencia a la visión filosófica del diseño, en que el proceso esta conducido por información acerca de la audiencia objetiva del producto, principalmente el DCU se distingue de otros enfoques por que el proceso que sigue no es secuencial o lineal, más bien presenta ciclos en los que iterativamente se puede probar el diseño y este es optimizado hasta alcanzar el nivel de calidad y usabilidad que se requiere.

El díselo centrado en el usuario consiste en que el producto sea asentado en base a la información de usuarios reales, esa información se obtiene por medio de encuestas, entrevistas, investigación conceptual, etc.

Pero claro previo a todo esto se necesita que el diseñador le dé una forma digamos tangible al producto para que este pueda ser manejado, es aquí en que se emplea la técnica de personajes y escenarios; en donde los denominados personajes son representaciones de los usuarios con distintos patrones de conducta, objetivos y/o necesidades.

Se pueden utilizar arriba de 5 personajes, para que cada uno tenga características de una parte de la población de usuarios, se llenan fichas con nombres y cargos ficticios, así como una descripción sobre su entorno y patrones de conducta, así como las necesidades que podría tener. Esta información traducida se denomina “modelos mentales” estos sirven para poder agrupar a la población de usuarios de acuerdo a sus preferencias y patrones de conducta.

Los modelos mentales son un concepto psicológico que hace referencia a representaciones internas de una realidad externa, representaciones que somos capaces de construir a partir de nuestras experiencias. En palabras de Norman:

Es la forma conceptual que tenemos acerca de cómo funcionan los objetos, cómo tienen lugar los hechos o cómo se comporta la gente, y son resultado de la tendencia a formar explicaciones de las cosas. Estos modelos son esenciales para comprender experiencias, predecir el resultado de acciones y para manejar situaciones inesperadas. Los modelos mentales se basan en cualquiera que sea el conocimiento existente, real o imaginario, ingenuo o sofisticado. Los modelos mentales a menudo están contruidos sobre evidencias incompletas, sobre un escaso conocimiento acerca de lo que está ocurriendo, y con un tipo de psicología ingenua que postula causas, mecanismos y relaciones, incluso cuando no existen (Montero 2015).

Un modelo mental es una parte esencial del pensamiento sistemático, es decir, es la manera en que se percibe el entorno y sus componentes y así nos guiamos en realizar ciertas actividades de una forma en específica.

Un modelo mental se vale de la eliminación que hace referencia al proceso de deshacernos de cierta información de la mente para mantener un modelo mental, se procede a construir, es decir, agregar información útil al modelo mental. Pero al agregar información se distorsiona o cambia la perspectiva, se amplían algunas cosas y disminuyen otras. Cuando se habla de generalización se refiere a convertir una experiencia en representativa de un grupo mayor o general. Por último, tenemos los prejuicios, esto se refiere a cuando ya se posee un modelo mental que influye mucho sobre otro.

Según Peter Senge (2016) muchas ideas fracasan debido a que, aunque suponen estrategias muy buenas y efectivas casi nunca se realizan, los conceptos sistemáticos nunca se integran a políticas operativas, los modelos mentales no solo determinan el modo de interpretar el mundo, si no el modo de actuar.

En esencia los modelos mentales influyen en las decisiones que toma un usuario, por ejemplo, dos personas con diferentes modelos mentales pueden observar una misma cosa, pero describirlo de forma totalmente distinta por que perciben diferentes detalles, y realizar sobre ese objeto distintas acciones, según los psicólogos se le llama “observación selectiva”. Claro que esto no quiere decir que estas diferentes percepciones sean erróneas o correctas el problema real es que estos modelos sean tácticos, es decir, que estén centrados en ideas o filosofías y no es los usuarios realmente.

1.9.4 Requerimientos Basados en Experiencia de usuario

El primer punto sobre la experiencia de usuario es el cómo se une a él, el concepto de arquitectura de información (AI), cuyo contexto refiere el manejo, análisis, organización y disponibilidad de los datos necesarios para los sistemas de información.

Todo producto de software está diseñado para solucionar una o varias necesidades de una cantidad de usuarios por ello que se dice que se encuentra en un contexto (PSS) sistema de servicio de producto, es importante combinar productos y servicios apropiadamente y que

estén totalmente enfocados en las necesidades del cliente para el análisis de requisitos deben estar muy puntuales y claras las situaciones específicas en el uso del producto deben aclararse teniendo en cuenta el contexto del trabajo del cliente.

De un estudio de varios proyectos de software se encontró que la mayoría de los requerimientos no cumplía las características fundamentales. De la muestra de requerimientos utilizada para la evaluación, se pudo identificar que ningún documento cumple completamente con las características de calidad, solo el 30% de dichos requerimientos cumple con algunas características de un buen requerimiento. Se encontró que del total de defectos reportados el 23% de los errores se dan por inadecuada definición de requerimientos (Rodríguez C. T. 2017).

La calidad del sistema depende de una buena comunicación con el cliente, la forma de levantamiento de requerimientos, la eficacia y exactitud al plasmar las necesidades en los requerimientos. Estos requerimientos dan la oportunidad de generar un documento adecuado de casos de uso con el cual el análisis, diseño y desarrollo del sistema es claro para terminar con un sistema que realmente cumpla con las necesidades del cliente, en tiempos razonables y precio justo (Licon, 2008).

Capítulo 2

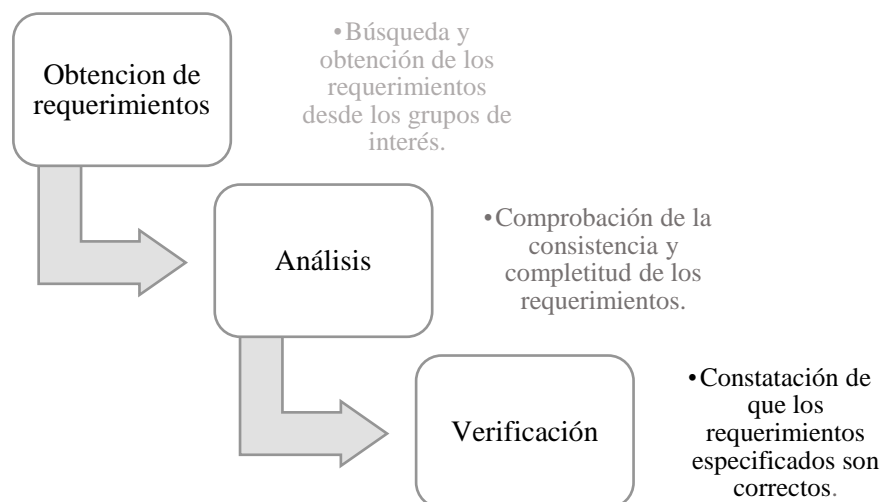
Ingeniería de Requerimientos con Enfoque en el Diseño Centrado en el Usuario.

En la industria del software y el desarrollo de instrumentos de cualquier índole es de suma importancia el buen manejo de los requerimientos de los usuarios, en los problemas y situaciones reales que el producto podría enfrentar, es por ello que en este trabajo es de suma importancia el extenso conocimiento sobre la ingeniería de requerimientos desarrollado bajo los métodos del diseño enfocado en el usuario.

2.1 Requerimientos

Un requerimiento se puede traducir como una necesidad o una característica que contribuye a la solución de un problema. “Los requerimientos son la declaración de los atributos, características, funciones o cualidades que debe cumplir un sistema de software para que pueda decirse que este tiene un buen nivel de usabilidad” (Leandro A.2016).

A continuación, se muestra en la figura 1 el ciclo de establecimiento de requerimientos:



Leandro Alegsa. (2016). Figura 1 Etapas de la fase de requerimientos.
Recuperado de <http://www.alegsa.com.ar>

Como muchas áreas del ciclo de vida del software los requerimientos poseen una clasificación, a continuación, se muestra la clasificación de los requerimientos.



Figura 2 Clasificación de los requerimientos.
 Recuperado de <https://administracionderequerimientos.wordpress.com>.

En donde:

Requerimientos de usuario: Son las características que los usuarios esperan del sistema son dadas en lenguaje natural y/o diagramas.

Requerimientos de sistema: Son establecidos de manera detallada las funciones, servicios y restricciones operativas del sistema.

Requerimientos Funcionales: Engloba las funciones que un sistema debe realizar, por ejemplo, sumar, restar, enviar a otro vinculo etc. O declaran que no debe hacer ejemplo “El sistema no permitirá la impresión de una factura”. A su vez los requerimientos funcionales se clasifican en:

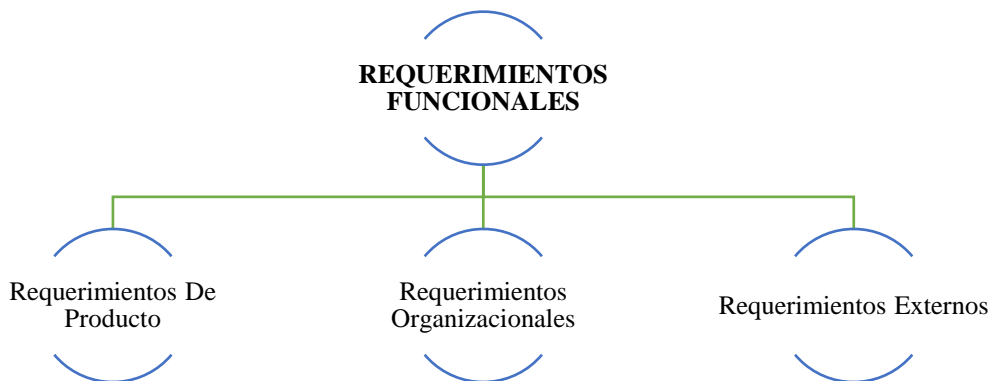


Figura 3 Clasificación de requerimientos funcionales.
 Recuperado de <https://administracionderequerimientos.wordpress.com>

Requerimientos no Funcionales: Son las restricciones que el sistema debe tener referentes a su funcionalidad, por ejemplo, restricciones de tiempo, estándares de desarrollo, fiabilidad, capacidad de almacenamiento, tiempo de respuesta etc.

Las actividades relacionadas con la ingeniería de requerimientos según Roger S. Pressman (2011) son:

Inicio: Se identifica una oportunidad de negocios o se descubre un nuevo mercado o servicio potencial. Los ingenieros de software hacen una serie de preguntas libres de contexto para conocer las necesidades del cliente y determinar si el proyecto es factible.

Obtención: Se inicia la definición de los requisitos de manera más profunda, incluso se pueden hacer modificaciones a los requerimientos planteados durante la etapa de inicio.

Elaboración: Desarrollo un modelo técnico refinado de las funciones, características y restricciones del software. Creación y el refinamiento de escenarios del usuario para describir la forma en que el usuario final interactuara con el sistema.

Negociación: En esta etapa se deberán poner de acuerdo tanto los clientes como los desarrolladores. Para desechar los requerimientos que no sean necesarios o de igual manera los que no estén alcance realizarlos.

Especificación: Desarrollo del documento formal de especificación de requisitos, es la base para las actividades de ingeniería de software subsiguientes.

Validación: Examinar la especificación para asegurar que todos los requisitos de software se han establecido de manera precisa.



Roger S. Pressman (2011).
Figura 4 Actividades relacionadas con la ingeniería de requerimientos.
Recuperado de: Ingeniería de software 7

Gestión de requisitos: Actividades que ayudan al equipo del proyecto a identificar, controlar y rastrear los requisitos y los cambios a estos en cualquier momento mientras se desarrolla el proyecto.

2.2 Diseño Centrado en el Usuario

El origen del todo desarrollo de un proyecto ya sea de software o de cualquier otra disciplina, parte de una idea cuyo objetivo es hacer un proceso más sencillo, es decir están inspirados en ser de utilidad para un usuario, por ejemplo en el sector automotriz cada nuevo modelo de auto posee más características para que la persona que lo conduce lo haga de forma intuitiva un ejemplo más claro podrían ser los botones esta ubicados cada vez en mejor posición para que sean detectados de forma más sencilla.

El diseño centrado en el usuario es definido por algunas instituciones como *Usability Professionals Association* como un enfoque de diseño y creación dirigido por información obtenida de los usuarios. En la década de los 50s la atención de los informáticos estaba enfocada en comprender la manera en que un ordenador trabajada y no en la forma de trabajo de los usuarios.

“Un enfoque de diseño DCU refiere que debe ser el usuario final el que prevalezca, sobre otros factores en la toma de decisiones” (Yusef M. y Sergio. S. 2009). De acuerdo a la norma ISO 13407, podemos dividir este proceso en cuatro fases (Figura 5):

Entender y especificar el contexto de uso: Identificar a las personas a las que se dirige el producto, ¿para qué lo usarán? y en qué condiciones.

Especificar requisitos: Identificar los objetivos del usuario y del proveedor del producto deberán satisfacerse.

Producir soluciones de diseño: Esta fase se puede subdividir en diferentes etapas secuenciales, desde las primeras soluciones conceptuales hasta la solución final de diseño.

Evaluación: Es la fase más importante del proceso, en la que se validan las soluciones de diseño (el sistema satisface los requisitos) o por el contrario se detectan problemas de usabilidad, normalmente a través de test con usuarios.

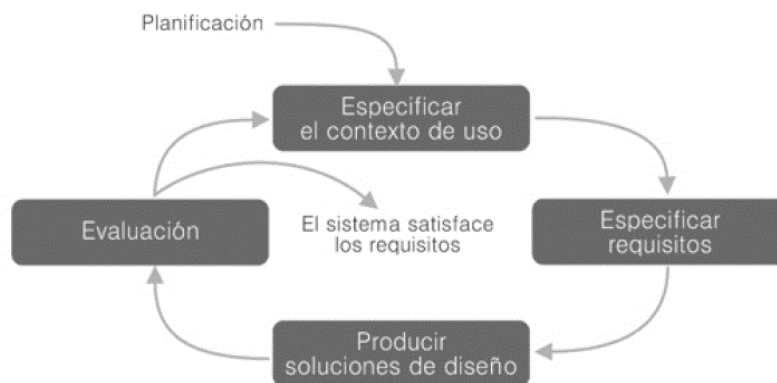


Figura 5 Proceso del Diseño Centrado en el Usuario
 Recuperado: <http://www.nosolousabilidad.com/manual/3.htm>

2.2.1 Metodologías y Técnicas de DCU

El modelo de diseño centrado en el usuario engloba varias metodologías y técnicas que comparten el objetivo de conocer y comprender las necesidades, limitaciones y comportamiento de características del usuario, involucrando en la mayoría de ocasiones al usuario real o potencial. A continuación, se responden algunas preguntas sobre dichas técnicas tales como descripción, procedimiento, ubicación en el ciclo del producto, y qué limitaciones o problemas pueden presentar.

Test de usuario

Esta prueba es la más importante del DCU, esta es la mejor manera de evaluar la usabilidad de un sistema, básicamente estas pruebas se realizan con la observación de un grupo de usuarios realizando pruebas recomendadas por un evaluador, sin dar a conocer el procedimiento para realizarlo de esta forma el usuario determina la forma en que llegara al resultado esperado, con esto se puede conocer el nivel de usabilidad. Sin importar cuanto conocimiento posea un diseñador sobre la usabilidad es de suma importancia saber cómo es percibido el sistema por el usuario.

Procedimiento

Es muy importante planear bien las sesiones de pruebas de usuario por ejemplo Nielsen (2000) dice El número de participantes para una prueba debe ser alrededor de 15, entonces en vez de hacer una prueba con 15 participantes, es mejor llevar a cabo tres pruebas con 5 participantes por cada una, repartidas en diferentes momentos del proceso de desarrollo.

El objetivo de dichas pruebas es mejorar de forma interactiva la usabilidad de la aplicación, por lo que cada prueba con 5 participantes ofrece suficiente información para mejorar la solución del diseño.

Cuando

Estas son pruebas de evaluación, pero eso no significa que deban ser realizadas al final del proceso del producto, la filosofía de DCU es diseñar iterativamente basados en la mejora incremental del producto. Por tanto, cuanto más esperamos para realizar la primera de las pruebas, más costoso resultará la reparación de los errores de diseño que se detecten.

En las etapas más tempranas del proyecto, ya que el producto aún no ha tomado forma, los test de usuarios deben realizarse sobre prototipos.

Limitaciones y problemas

Los test de usuarios es el alto coste que podría implicar tanto el reclutamiento de los participantes, como el tiempo y esfuerzo dedicados a realizar las pruebas y a sintetizar y analizar los resultados.

El otro problema es que, al tratarse de pruebas que se realizan en laboratorio y en las que los objetivos y tareas se les imponen explícitamente a los participantes, la interacción del usuario se encuentra descontextualizada, influyendo en su forma de resolver problemas.

Eye-Tracking

Esta técnica está basada en cómo un usuario percibe o que elementos espera ver y que orden en una interfaz para realizar una interacción más cómoda, por ejemplo, si podemos anticipar de alguna forma o saber que elementos espera el usuario tales como los contenidos navegación, mapas contactos, logos etc. Y son colocados estratégicamente en la interfaz, de una u otra forma se puede guiar la interacción usuario-sistema.

Las pruebas de seguimiento visual o *eye.tracking* son ejecutadas por medio de software y hardware que hacen posible tener un registro de como un usuario visualiza una escena y que zonas posiciona su mirada (Montero Y. Solana H.2007). A continuación, en las figuras 6 y 7 se muestran algunos ejemplos de ejecución de estas pruebas, correspondiendo la número 6 a los recorridos visuales que realizaron los usuarios y la figura número 7 a una muestra de las zonas que son vistas primero más veces por más usuarios.

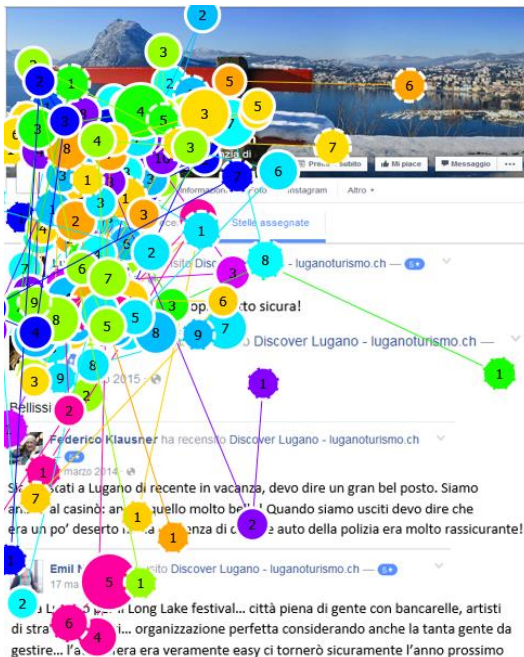


Figura 6 Historias vividas sobre una interacción exitosa con la aplicación
 Recuperado de: <http://www.webatelier.net/eye-tracking-lab>

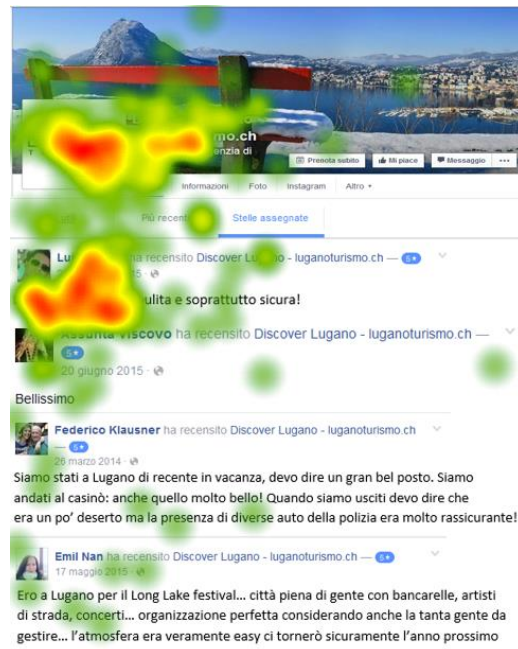


Figura 7 Comportamiento de los usuarios finales
 Recuperado de: <http://www.webatelier.net/eye-tracking-lab>

Para el análisis del comportamiento visual se emplean *heatmaps* o mapas de calor, donde los colores de mayor intensidad señalan las zonas de la interfaz en las que los participantes han fijado su atención con mayor frecuencia (figura 7).

Limitaciones y Problemas

Aún que actualmente los sistemas disponibles de *eye-tracking* son bastante exactos una limitación de los mismos es que son altamente costos, además existe un significativo porcentaje de personas cuyos ojos no pueden calibrarse lo que aumenta el costo.

“Las pruebas de *eye-tracking* requieren del evaluador un conocimiento y esfuerzo considerable en la interpretación de los resultados, por lo que su uso inexperto puede conducir a conclusiones erróneas” (Montero Y.2016).

2.3 Prototipos

Un prototipo es una representación simple pero fiel de la idea principal de un algo, es decir es una forma de representar un proyecto ya sea en menor escala o forma sencilla para encontrar deficiencias en el diseño principalmente.

El uso de los prototipos no sólo sirve para probar las interacciones que los usuarios puedan realizar, son igualmente útiles para otras como la recolección y análisis de requisitos ya que amplía y mejora la información necesaria para el desarrollo del sistema.

Las principales características de los prototipos son:

- Contribuyen a mejorar la comunicación y la participación de los usuarios con los desarrolladores.
- Brindan más informes para toma de decisiones basadas en experiencia.
- Permiten la exploración de opciones en cuanto al diseño sin realizar trabajo arduo antes de establecer el definitivo.
- Son muy útiles para completar los requerimientos de los usuarios y así mejorar la calidad y la usabilidad.
- Ayudan con el desarrollo de documentación

Los prototipos pueden desarrollarse en dos formas, de baja fidelidad y de alta fidelidad los primeros utilizan aspectos del sistema sin demasiados detalles es decir podrían mostrar los recorridos de un sistema y las características visuales que contendrá, pero estas no funcionarían realmente, mientras que los siguientes se muestran aspectos más específicos que detallan el proceso de una o varias tareas. (Saltiveri T.2004)

2.4 Casos de Uso

Un diagrama de casos de uso forma parte del lenguaje UML y es útil para representar la funcionalidad o interacciones del usuario y las respuestas del sistema de manera coherente, los elementos de un modelo de casos de uso son:

- Actores:
Primarios interaccionan con el sistema para explotar su funcionalidad
Secundarios soporte del sistema para que los primarios puedan trabajar
Iniciadores no utilizan directamente el sistema, pero desencadenan el trabajo de otro actor
- Caso de uso
- Relación
- Respuesta

Los tipos de casos de uso se dividen en los siguientes:

- Resumidos
- Extensos
- Esenciales

Los casos de uso se colocan en los documentos en dos formas la primera es con una planilla de descripción detallada de casos de uso en la que se describe de forma escrita cada detalle como el actor el curso normal y alterno etc. y la segunda en un diagrama de los mismos este es una descripción gráfica de la interacción y las respuestas que el usuario obtendrá del sistema.

Descripción detallada de casos de uso

Caso de Uso	<< Nombre del CU >>		<< Identificador >>	
Actores	<< Listado de los actores participantes en el CU >> << Podemos indicar quien es el que inicia el CU usando (I) >>			
Tipo	<< Tipo del caso de uso >> << Primario, Secundario u Opcional >> <<Esencial o Real >>			
Referencias	<< Indicamos que requisitos se pueden incluir dentro de este CU >>	<< CU que tienen relación con este >>		
Precondición	<< Condiciones sobre el estado del sistema que tienen que ser ciertas para que se pueda realizar el CU >>			
Postcondición	<< Efectos que de forma inmediata tiene la realización del CU sobre el estado del sistema >>			
Autor	<< Esta línea se podría repetir para mantener una historia de cambios del CU >>	Fecha		Versión

Propósito
<< Descripción general del CU (Suficiente con una línea) >>

Resumen
<< Descripción de alto nivel del flujo normal (básico) del caso de uso (Suficiente con un pequeño párrafo) >>

Figura 8 Plantilla de detalle de casos de uso. Recuperado de Casos de uso Mega M. 2010

Diagramas de casos

Los diagramas de casos de uso muestran relaciones entre la interacción sistema y sus actores, los casos de uso se representan mediante elipses con el nombre, los actores son representados por dibujos con figura humanoide y se indica “Actor” debajo del mismo a continuación un ejemplo en la figura 9. (Miguel V. 2010)

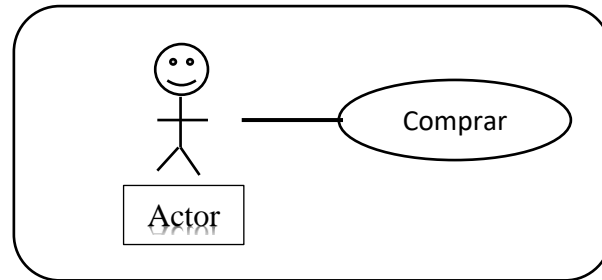


Figura 9 Elementos de un diagrama de casos de uso.
Recuperado de Casos de uso

En este diagrama pueden darse las siguientes relaciones:

Inclusión, el caso de uso A incluye a B es decir que para cumplir A podría intervenir B.

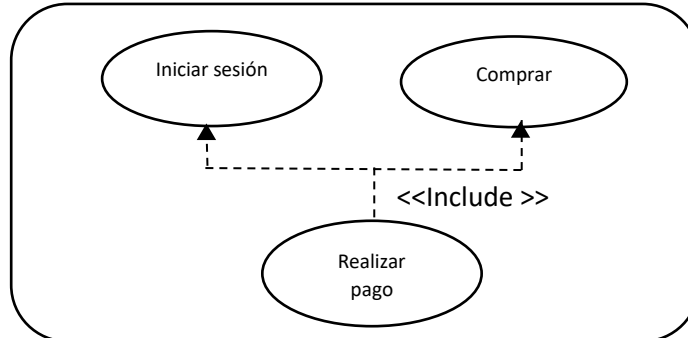


Figura 10 Ejemplo de una relación de inclusión.

Extensión, el caso A podría incorporar el comportamiento especificado en B.

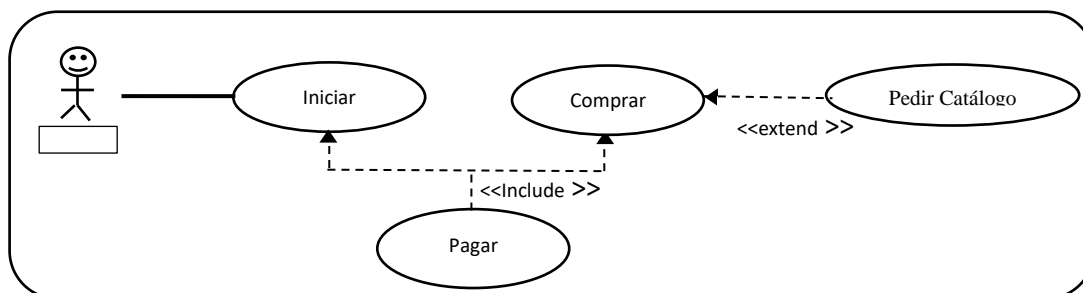


Figura 11 Ejemplo de una relación de extensión.

2.5 Estándares de Calidad

Norma ISO 9001 2015

ISO 9001: 2015 es la norma internacional de Sistemas de Gestión de la Calidad (SGC), incorpora el ciclo PHVA (Planificar, Hacer, Verificar, Actuar) y pensamiento basado en riesgos.

Permite a una organización:

Planificar sus procesos y sus interacciones.

Asegurarse de que sus procesos cuenten con recursos y se gestionen adecuadamente.

Que las oportunidades de mejora se determinen y se actúe en consecuencia.

Esta Norma Internacional se basa en los principios de la gestión de la calidad descritos en la Norma ISO 9000.

Los principios de la gestión de la calidad son:

- Enfoque al cliente
- Liderazgo
- Compromiso de las personas
- Enfoque a procesos
- Mejora
- Toma de decisiones basada en evidencia
- Gestión de relaciones.

Esta norma internacional promueve la adopción de un enfoque a procesos al desarrollar, implementar y mejorar la eficacia de un sistema de gestión de la calidad, para aumentar la satisfacción del cliente mediante el cumplimiento de los requisitos del cliente.

El enfoque a procesos implica la definición y gestión sistemática de los procesos y sus interacciones, con el fin de alcanzar los resultados previstos de acuerdo con la política de la calidad y la dirección estratégica de la organización.

La aplicación del enfoque a procesos en un sistema de gestión de la calidad permite:

- ✓ La comprensión y la coherencia en el cumplimiento de los requisitos
- ✓ La consideración de los procesos en términos de valor agregado
- ✓ El logro del desempeño eficaz del proceso
- ✓ La mejora de los procesos con base en la evaluación de los datos y la información.

Pensamiento Basado en Riesgos

“El pensamiento basado en riesgos ha estado implícito en ediciones anteriores de esta Norma Internacional, incluyendo, por ejemplo, llevar a cabo acciones preventivas para eliminar no conformidades potenciales, analizar cualquier no conformidad que ocurra, y tomar acciones

que sean apropiadas para los efectos de la no conformidad para prevenir su recurrencia” (Organización Internacional de Normalización ISO.2015).

CMMI

Lanzado en 1987 como *Capability Maturity Model*, un proyecto del Instituto de ingeniería de software, centro de investigación de la universidad Carnegie-Mellon. En 1991, se publicó por primera vez el modelo CMMI para software, que está basado en una lista de comprobación de los principales factores de éxito de los proyectos de desarrollo de software realizados a finales de los años setenta y principios de los años ochenta. La proliferación de nuevos modelos dio lugar a confusión, por lo que el gobierno financió un proyecto de dos años en el que participaban más de 200 expertos del mundo industrial y académico a fin de crear un solo marco extensible para la ingeniería de sistemas, la ingeniería de software y el desarrollo de productos. El resultado fue Modelo de Capacidad de Madurez por sus siglas en inglés CMMI.

Es un modelo de mejora de rendimiento de clase mundial, consiste en recolectar las mejores prácticas diseñadas para promover los comportamientos que conducen a un mejor desempeño en cualquier organización, CMMI cuenta con cuatro distintos modelos que ayudan a identificar las capacidades clave para elevar el rendimiento, la calidad y rentabilidad, que se muestran a continuación (Figura. 12).



Figura 12 Clasificación CMMI.

Fuente: Portadas de manuales extraídas de modelo CMMI

Este trabajo estará centrado en el modelo CMMI *for development*, el cual es un conjunto de comportamientos organizativos de méritos demostrados en el marco del desarrollo de software y la ingeniería de sistema.

CMMI-Dev es el modelo de referencia para la mejora de las diferentes áreas de proceso en los proyectos de desarrollo y mantenimiento de software, podemos comprender así que es un conjunto de buenas prácticas que cubre el ciclo de vida del producto, desde su concepción

hasta su entrega y mantenimiento, por lo tanto, es este modelo indicado para empresas que se desenvuelven en la rama de desarrollo y mantenimiento del software (Figura 13).

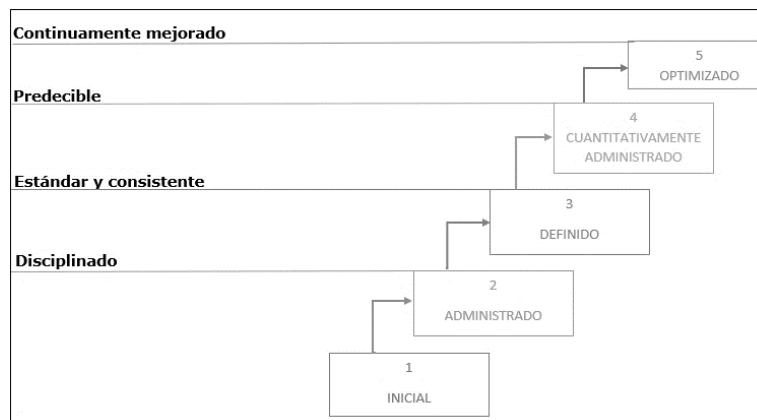


Figura 13 Niveles de Madurez CMMI-Dev.
Fuente: ASE. 2008.

Análisis de cada nivel de madurez de CMMI:

Nivel 1 Inicial

La organización dispone de un proceso guiado o apegado a normas y/o modelos de buenas prácticas y logran objetivos.

Nivel 2 Administrado

No sólo dispone de un proceso o modelo, cada paso es revisado y evaluado para comprobar que se cumplen los requisitos definidos.

Nivel 3 Definido

La organización posee un proceso gestionado y este se ajusta a la política de procesos marcada por la organización.

Nivel 4 Cuantitativamente gestionado

Los procesos se controlan utilizando técnicas cuantitativas (pápel, pruebas etc.)

Nivel 5 Optimizado

Además de cumplir todas las condiciones de los niveles anteriores, de forma sistemática se revisa y modifica o cambia para adaptarlo a los objetivos del negocio (Allsoft software engineering, 2008).

MoProSoft

Modelo de procesos para la industria de software, que contribuye a la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. Desarrollado por la Asociación Mexicana para la Calidad en Ingeniería de Software a través de la Facultad de ciencias de la Universidad Nacional Autónoma de México (UNAM) y a solicitud de la secretaria de Economía para obtener una norma mexicana que resulte apropiada a las características de tamaño de la gran mayoría de empresas mexicanas de desarrollo y mantenimiento de software. Moprosoft es el nombre del modelo en la comunidad universitaria y profesional, y la norma técnica de contenido es la NMX-059/01-NYCE-2005 que fue declarada Norma Mexicana el 15 de agosto de 2005 con la publicación de su declaratoria en el Diario oficial de la Federación.

Un conjunto integrado de las mejores prácticas basadas en los modelos y estándares reconocidos internacionalmente, tales como ISO 9000:2000, CMM-SW, ISO/ IEC 15504, PMBOK, SWEBOK entre otros (Oktaba H., Alquicira C., et al., 2003).

La categoría de Alta Dirección contiene el proceso de Gestión de Negocio; la categoría de Gestión se compone de Gestión de Procesos, Gestión de Proyectos y Gestión de Recursos, a su vez, este último se divide en tres subprocesos: el de Recursos Humanos, el de Bienes, Servicios e Infraestructura y el de Conocimiento de la Organización. Finalmente, la categoría de Operación contiene los procesos de Administración de Proyectos Específicos y de Desarrollo y Mantenimiento de Software.

Al estandarizar los procesos y las prácticas usando Moprosoft, la industria de software podría tener las siguientes ventajas, que ayudarían en su crecimiento:

- Al tener prácticas integradas, que abarcan desde la gestión de negocio hasta el desarrollo y mantenimiento de software, las empresas tendrían mayor control sobre su desempeño en el mercado.
- El costo de la incorporación del nuevo personal podría disminuir si se enfocan la educación y la capacitación a un modelo único.
- Las empresas pequeñas, al seguir procesos similares, podrían asociarse con mayor facilidad para afrontar proyectos de mayor envergadura.
- La exportación de servicios de software de las empresas mexicanas sería más factible. Incluso se podría disminuir la necesidad de la intermediación de las empresas transnacionales, gracias a que Moprosoft considera las prácticas reconocidas internacionalmente (Allsoft software engineering. 2011).

Actividades para el proceso de obtención de requerimientos de CMMI y MoProSoft

En las empresas se emplean los modelos de calidad CMMI y Moprosoft, que dictan las siguientes actividades para el proceso de obtención de requerimientos:

MoProSoft

Fase de Requerimientos

Entrada: Plan de Desarrollo

Distribución de tareas a los miembros del equipo de trabajo según su rol, de acuerdo al plan de desarrollo actual.

Obtención de los requerimientos y documentación o modificación de la Especificación de requerimientos.

Verificación y validación de la Especificación de requerimientos, generando el reporte de Verificación y Validación.

Elaboración o modificación del plan de Pruebas de Sistema.

Verificación del Plan de Pruebas de Sistema y generación del Reporte de Verificación.

Incorporación de la Especificación de Requerimientos y del Plan de Pruebas de Sistema como líneas base a la Configuración de Software.

Elaboración de la Especificación de Requerimientos correspondientes a esta actividad.

Salidas: Configuración de Software, Reporte(s) de Verificación, Reporte de Validación, Reporte de Actividades.

CMMI Dev.

Gestión de Requerimientos CMMI- Dev

Definir los requisitos del cliente:

SG1 Las necesidades, expectativas, restricciones e interfaces de los interesados son recogidas y traducidas a requisitos del cliente.

SP1.1 Obtener las necesidades, las expectativas, las restricciones, y las interfaces de los interesados para todas las fases del ciclo de vida del producto.

SP1.2 Transformar las necesidades, las expectativas, las restricciones y las interfaces de las partes interesadas en requisitos del cliente.

Derivar los requerimientos y componentes del producto:

SG2 Los requisitos del cliente son refinados y elaborados para desarrollar los requisitos del producto y de componentes del producto.

SP2.1 Establecer y mantener los requisitos del producto y de componentes del producto, los cuales están basados en los requisitos del cliente.

SP2.2 Asignar los requisitos para cada componente del producto.

SP2.3 Identificar los requisitos de la interfaz.

Analizar y validar los requisitos definidos:

SG3 Los requisitos son analizados y validados, y una definición de la funcionalidad requerida es desarrollada.

SP3.1 Establecer y mantener los conceptos operativos y los escenarios asociados.

SP3.2 Establecer y mantener una definición de la funcionalidad requerida y los atributos de calidad.

SP3.3 Analizar los requisitos para asegurarse de que son necesarios y suficientes.

SP3.4 Analizar los requisitos para equilibrar las necesidades y las restricciones de los interesados.

SP3.5 Validar los requisitos para asegurar que el producto resultante se ejecutará según lo previsto en el entorno del usuario.

Capítulo 3

Modelo de Recolección de Requerimientos Centrado en el Usuario

Con base en el análisis previo realizado en una empresa que desarrolla proyectos reales y cuyo plan de trabajo emplea practicas establecida en el modelo CMMI y MoProSoft, cuyo proceso prevé un desarrollo detallado de la fase de análisis de requerimientos de proyecto de software, se identificaron áreas de oportunidad de mejora, en las cuales se podría implementar una estrategia que ayude a cumplir cabalmente las necesidades del usuario y brinde a cada proyecto un nivel más alto de calidad.

Se plantea un modelo estratégico que contempla una reducción del impacto de las evoluciones en el entendimiento de requerimientos, y que se ajusta considerando los recursos existentes en la empresa, este modelo puede ser aprendido por los integrantes actuales del equipo de fábrica de software. (Figura 14).

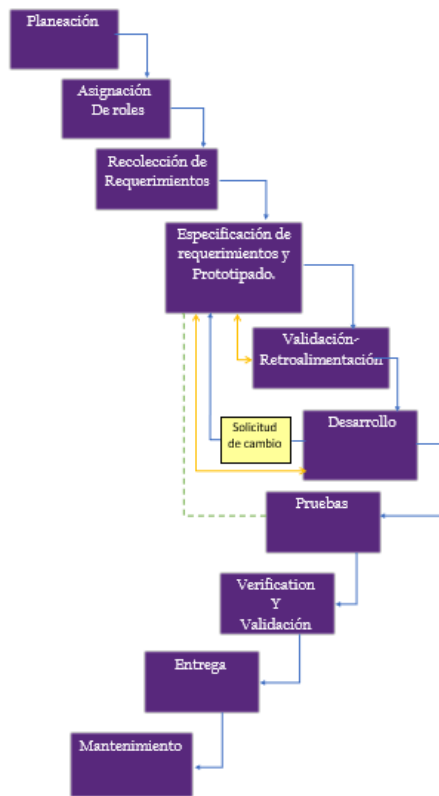


Tabla 1 Simbología.

<i>Flecha</i>	<i>Relación</i>
→	Secuencial
- - -	Dependencia
↔	Retroalimentación

Figura 14 Modelo Recolección de requerimientos centrado en el usuario.

A continuación, en la figura 15, se muestra el modelo propuesto en un diagrama BPMN.

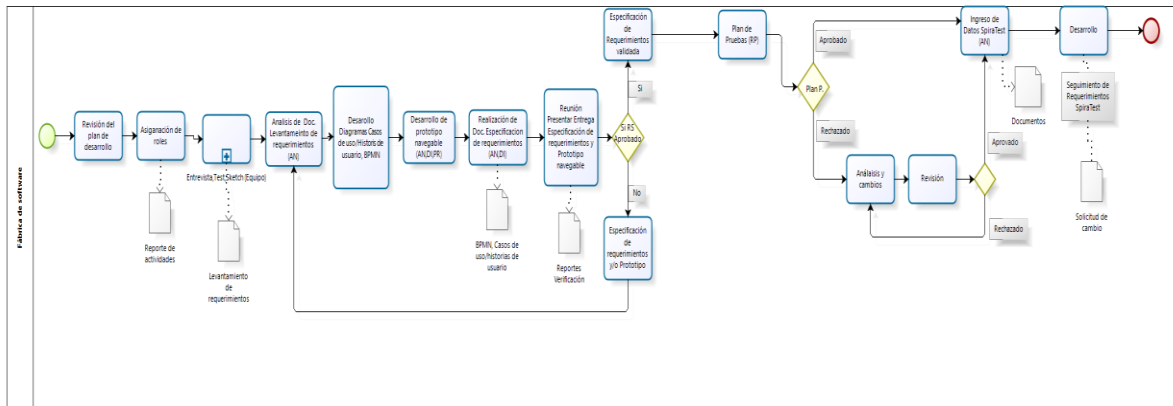


Figura 15 Modelo de recolección de requerimientos centrados en el usuario en diagrama BPMN.

La primera casilla marca una revisión del plan de desarrollo elaborado por el equipo de administración de proyectos (MoProSoft) que brindan información como políticas, estrategias, objetivos y metas a los integrantes equipo de fábrica de software, una vez revisado y comentado este plan, el siguiente punto lleva al equipo de desarrollo a determinar cuáles será el rol o roles de cada integrante, especificar sus tareas y documentos entregables.

En cuanto este especificado el analista de requerimientos, arquitecto de software y responsable de pruebas, el cliente será citado para una sesión de entrevista en la cual se aplicarán los test incluidos en este capítulo y se pedirá al cliente realice su propio sketch de como visualiza en ese momento el producto que requiere, se sugiere la grabación de sonido de la reunión (con autorización de los presentes) para un análisis posterior, el objeto de realizar respaldos de toda esta información es entender no solo el sistema como tal sino también el entorno en el que será utilizado.

Una vez analizada la información el equipo define los requerimientos en diagramas de casos de uso, las iteraciones sistema – usuario y diagrama BPMN en este punto se une el diseñador UI/UX quien aportará conocimientos en la realización de un prototipo desarrollado en *Balsamiq Mockups* que permitirá un recorrido sencillo por el sistema para detectar posibles fallos antes de la codificación y oportunidades de mejora.

Esta información será plasmada en el documento cuyo nombre será “(iniciales del proyecto) Especificación de Requerimientos.doc” y será presentado al cliente para su aprobación, en caso de ser validada el analista de requerimientos y responsable de pruebas desarrollaran “Plan de pruebas” que será aprobado también por el cliente y en caso de ser positivo el AR procede a registrar la lista de requerimientos en *SpiraTest*, para su seguimiento.

Luego de determinado tiempo el equipo realiza una reunión de sincronización_(20 a 30 minutos máximo). Cada miembro del equipo inspecciona el trabajo que se está realizando y

se muestra al usuario el prototipo inicia diseñado y se hacen los cambios y adaptaciones necesarias.

Durante la iteración, el cliente junto con el equipo refinan la lista de requerimientos (para prepararlos para las siguientes iteraciones) y, si es necesario, cambian o re-planifican los objetivos del proyecto para maximizar la utilidad de lo que se desarrolla y el retorno de inversión.

El último día de la iteración se realiza una reunión el equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, re-planificando el proyecto.

Retrospectiva. El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Facilitador se encargará de ir eliminando los obstáculos identificados.

3.1 Gestión del Cambio

En todo desarrollo de software habrá cambios que permiten la evolución del mismo, y dichas transformaciones no se consideran perjudiciales para el producto, pero hay casos en los que estas alteraciones no son de evolución si no de corrección de errores ya sea de desarrollo o más frecuentemente en requerimientos cuyas apariciones de no ser manejadas correctamente podrían transformarse en graves fallas de usabilidad y el desecho del producto; en la Figura 16 Gestión de cambios se muestra el camino a seguir cuando el equipo de fábrica de software enfrenta una solicitud de cambio y el primer paso es analizar esa solicitud de cambio para definir si afecta directamente la funcionalidad o si afectará en cuanto a detalles visuales o iterativos.

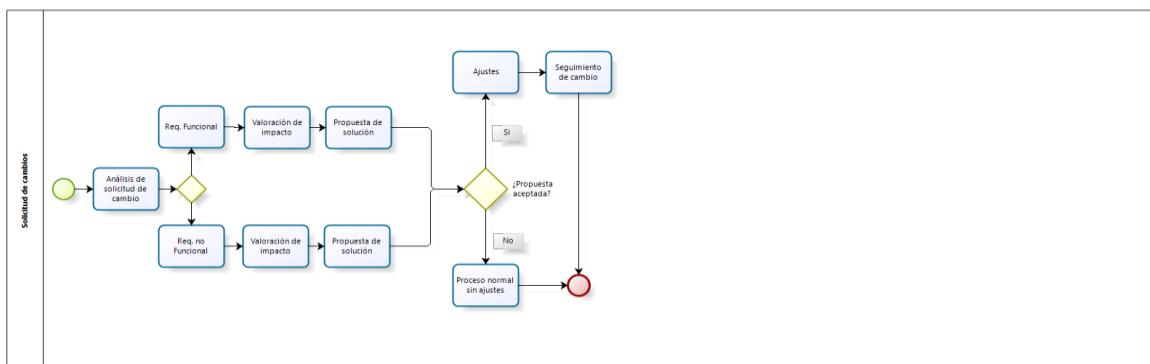


Figura 16 Gestión del cambio.

Lo siguiente es definir el impacto que este cambio genera, a otros módulos, por ejemplo, y con ello desarrollar una propuesta de solución para el cliente quien decidirá después de una detallada explicación de los cambios en todos los ámbitos del sistema, si es viable y/aceptable para el que se efectúen dichos cambios; en caso de acceder el equipo incluirá los ajustes necesarios y se realizara un seguimiento de esos cambios en la aplicación de *SpiraTest*.

Es necesaria una capacitación previa sobre conceptos básicos de Ingeniería de requerimientos, se propone una contribución de varios elementos del equipo de desarrollo de software para llevar a cabo un entendimiento del entorno en el cual será desarrollado un proyecto y por ende el dominio de conocimiento para establecer los requerimientos adecuados.

3.2 Descripción de Procedimiento

3.2.1 Asignación de Roles

De acuerdo a estas tareas y después de la revisión de plan de desarrollo, es importante el cabal entendimiento de los roles y responsabilidades dentro de un equipo de fábrica de Software para una mejor organización no solo de tareas sino también de que documentación.

A continuación, se muestra un diagrama con los roles sugeridos por Moprosoft para el equipo de desarrollo de software (Figura 17).

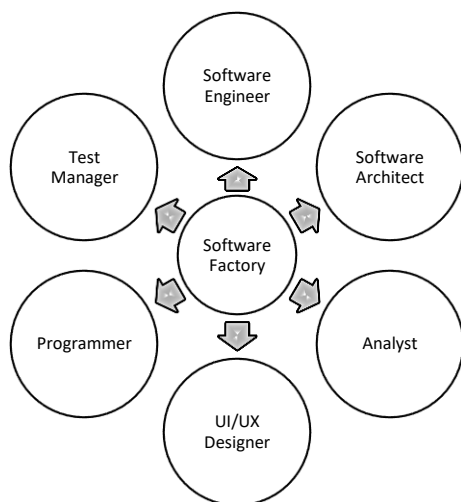


Figura 17 Roles necesarios en fábrica de software.

La adecuada organización dentro de fábrica de software para desarrollo de proyectos. Requiere de estos cinco roles comprendidos y llevados a cabo correctamente.

Ingeniero de Software poseerá conocimientos sobre los principios y las metodologías de desarrollo de software, aplicando el conocimiento científico al diseño, construcción y toda la documentación requerida para desarrollar, operar y brindar mantenimiento al software.

Arquitecto de software poseerá conocimiento sobre la organización y y/o estructura del programa, es decir, esquemas, diagramas, selección de lenguaje (en caso de no ser especificado por el cliente), descripción total para llevar a cabo el problema y en qué forma.

Analista es uno de los vínculos de unión entre el usuario y el área informática, elabora un análisis sobre las funcionalidades para el proyecto, para detectar y entender futuros problemas y oportunidades, requiere de habilidades para articula necesidades que están asociadas al problema principal, colaborar efectivamente con el equipo a través de sesiones de trabajo colaborativo, capacidades para comunicarse verbales y escritas, conocimientos en

negocios y dominios de tecnología, habilidad de absorber y entender rápidamente cualquier información. Idea y proyecta mensajes visuales, contemplando diversas necesidades que varían según el caso: estilísticas, informativas, identificativas, de persuasión, de código, tecnológicas, de producción, de innovación, etc. Creativos del color, las líneas, los detalles, las figuras e ideas nuevas que se combinan con el arte digital.

Diseñador UX/UI Este rol podría ser ejecutado por dos personas distintas ya que el diseñador UX poseerá habilidades en el diseño de la interacción total del sistema es decir la propuesta de navegación y respuestas del sistema basado en mapas mentales, percepciones e incluso emociones que pueda presentar el usuario, mientras que un diseñador UI posee conocimiento sobre diseño, pero referente a la estética de las interfaces.

Programador Desarrolla programas de computación analizando requerimientos, diseñando soluciones lógicas, usando las herramientas computacionales adecuadas, a fin de satisfacer los requerimientos del cliente.

Responsable de pruebas Identifica pruebas para llevar a cabo, y la forma adecuada de aplicarlas, implementa pruebas individuales, prepara y ejecuta las pruebas, registra resultados, realiza un análisis y recuperación de errores de ejecución.

A continuación, se muestra una distribución de roles Figura 18.



Figura 18 Distribución de roles para el desarrollo.

Es de suma importancia que nuestro futuro analista de requerimientos aprenda y entienda que debe ser muy cuidadoso en la manera en que se comunica con sus clientes y/o usuarios finales. En muchas ocasiones la persona con la que interactuaremos no estará familiarizada con el entorno de desarrollo de software y tendrá muchas ideas sobre lo que pretende del sistema en cuestión sin embargo no tendrá una manera precisa de cómo expresar las características que requiere, desde este punto debemos partir para comenzar a realizar un levantamiento de requerimientos de software.

¿Pero cómo entablamos esa comunicación con nuestro cliente? Existe un buen número de técnicas para obtener requerimientos, así que combinando estas múltiples técnicas podríamos adentrarnos más en el universo de ideas para el software en desarrollo.

La actividad más importante es precisamente la entrevista que realizaremos a nuestro cliente la cual debe ser lo más concisa y productiva posible con técnicas que nos permitan comprender mejor las ideas y necesidades a cubrir con el proyecto. Durante esta primera reunión, haremos uso del cuestionario preliminar propuesto en este guía con el fin de concretar algunos puntos que basados en otras experiencias resultan puntos que el equipo cataloga como obvios y resultan en grandes errores en etapas finales del proyecto.

Es preciso adecuar la comunicación con nuestro cliente es decir entablar una adecuada comunicación para el resto del desarrollo de nuestro proyecto, el cuestionario indicara que tipo de lenguaje deberíamos establecer para lograr una mejor comprensión, es decir utilizar una terminología amigable para todo el equipo de trabajo.

3.2.2 Recolección de Requerimientos

Inicialmente es posible concretar una reunión con el cliente directo, aunque sabemos que probablemente él no sea el usuario en cuestión, por esta cuestión es importante no sólo fijar los requerimientos sino también aterrizar ideas, una buena forma de comenzar es conociendo un poco sobre la empresa para la cual se desarrolla el proyecto en cuestión, esto dará una perspectiva más amplia acerca de nuestro problema a resolver o debilidades a atacar. Para realizar esta tarea se describen distintas técnicas necesarias para obtener un mejor conocimiento de los requerimientos del cliente.

Cuestionario Inicial

Lo primero que se debe indagar es el entorno al que pertenecerá el sistema a ser desarrollado, saber quién será el usuario final a quien estará dirigido el proyecto; es claro que en ocasiones la persona a cargo de la interacción con nosotros podría o no, estar habituada al rubro de la tecnología incluso si fuera así sus ideas pueden no ser las bastante claras al inicio.

De manera que este test preliminar podría establecerse de la siguiente manera.

- 1.- ¿Cuál es el departamento de la empresa al que está dirigido este proyecto?
- 2.- ¿Cuál es el objetivo de este proyecto?
- 3.- ¿Cuál es el número aproximado de usuarios?
- 4.- ¿Está familiarizado con algún lenguaje o entorno de programación?
- 5.- ¿Cuál es el sistema operativo más utilizado en la empresa que representa?
- 6.- ¿Cuál es la exigencia técnica (normas, y procedimientos a ser empleados y aplicados) de este proyecto?
- 7.- ¿Requisitos especiales de usuarios para este proyecto?
- 8.- ¿Tiene algún ejemplo o antecedentes que sirva como referencia?
- 9.- ¿Quién será nuestro contacto directo?
- 10.- ¿Cuál será su disponibilidad?
- 11.- ¿Tendríamos acceso a información de la empresa?
- 12.- Describa en sus palabras como visualiza el producto (Se otorgarán algunas hojas quizá lápices o cosas para realizar trazos).

Esta lista de preguntas está basada en el análisis de los problemas obtenidos de experiencias pasadas, en proyectos internos y externos de una empresa.

Debido a que una prueba de *eye-tracking* es altamente costosa no fue posible implementarla en este proyecto, pero sirvió de inspiración para el desarrollo del siguiente test denominado mapeo sobre pantalla blanca que básicamente posee la función de identificar de acuerdo al mapa mental del usuario, la posición y presentación de los distintos elementos, a continuación, se presenta el teste aplicado.

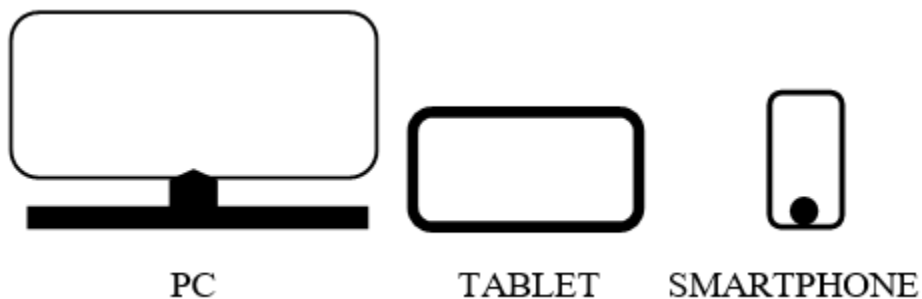
Esta prueba se implementará junto con el test inicial, para establecer la idea del mapa mental apropiado para elegir los artefactos y la posición dentro de la interfaz.

El objetivo de esta prueba es mostrar una visión cercana al desarrollador de cómo podría ser la interacción del usuario con la interfaz.

Inicio

Las ilustraciones que se muestran en la prueba son ejemplos, el resultado del prototipo podría variar.

Por favor, elija el o los dispositivos para trabajar.

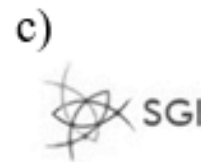
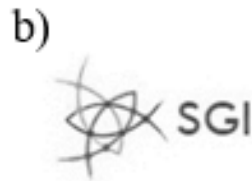


Cada elemento tiene un índice 1, 2, 3,4 etc. y opciones de selección como a), b), escribe el número del elemento y la letra que elijas, en una posición en la pantalla.

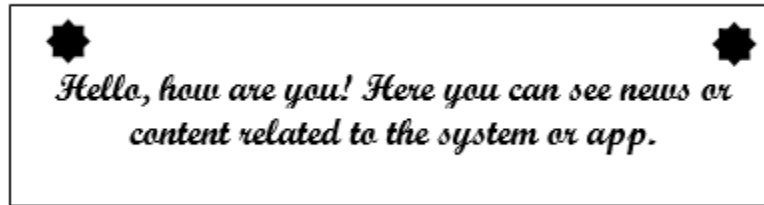


SCREEN

1.- Logotipo de la empresa, elija el tamaño y la posición escríbelo en la pantalla de arriba.



2.- Escribe el número 2 en la posición que elijas en la pantalla.



3.- Escribe el número 3 y la letra de tu elección en el lugar que elijas en la pantalla.

a) Horizontal



b) Vertical



4.- Escribe el número 4 y la letra de tu elección en el lugar que elijas en la pantalla.

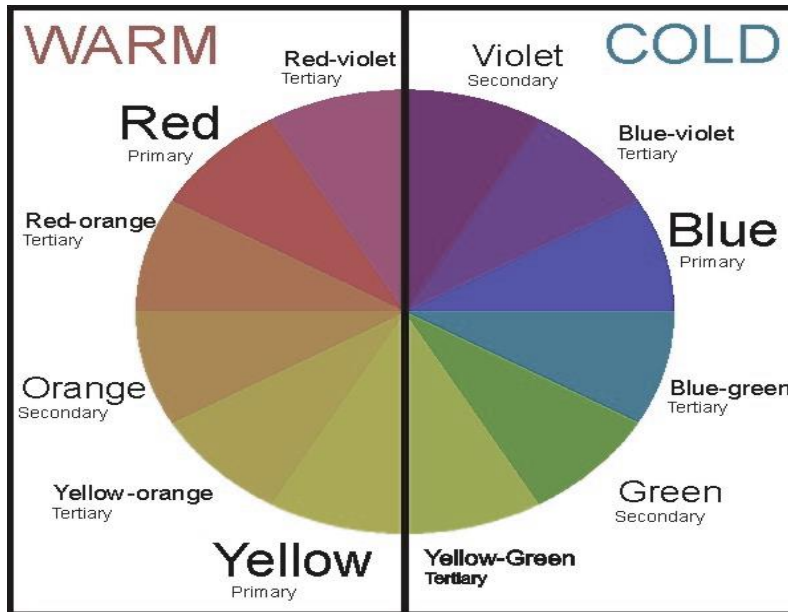
a) Icons



b) Description



5.- ¿Qué colores prefieres?



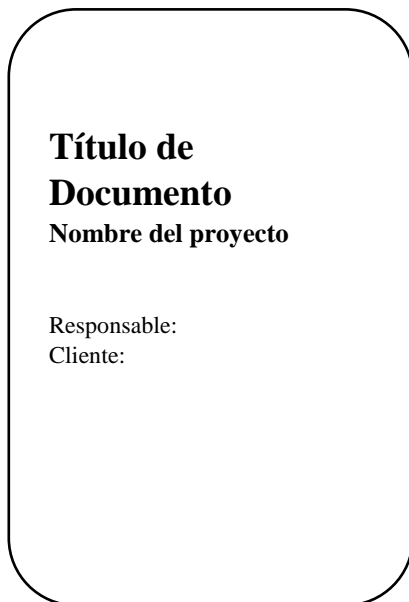
Escribe tus respuestas:

¡Gracias por tu ayuda!

Una vez que estos test estén llenos, es de mucha ayuda que el usuario dibuje una idea sencilla de la interfaz con las acciones que tiene en mente, a este boceto se le conoce como sketch, cuya definición puede darse como un dibujo hecho de manera provisoria, solamente con los elementos esenciales.

3.2.3 Desarrollo de Documentos Entregables y Establecimiento de Requerimientos

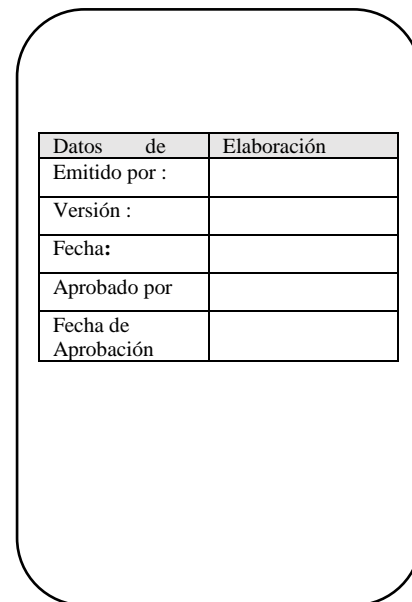
El departamento de fábrica de software realiza distintos entregables internos previos para revisión y validación, en esta primera fase de levantamiento de requerimientos se establece un documento que contendrá una lista preliminar de requerimientos, narrativa, reglas de negocio, Sketch y anotaciones, se adjuntaran grabaciones realizadas en la entrevista con el cliente cuyo contenido se encontrará en una sola plantilla entregable. Con base en el análisis minucioso de este documento se dará pie a redactar el documento entregable “Especificación de Requerimientos” (Figura 19 y 20) o ERS, dentro de este documento se encuentran diagramas que llevan el nombre BPMN (Modelo de proceso empresarial y notación), este será desarrollado con la herramienta *Bizagi Process Modeler*, de igual manera el documento ERS incluirá un diagrama de casos de uso.



Título de Documento
Nombre del proyecto

Responsable:
Cliente:

Figura 19 Portada del documento de requerimientos.



Datos de	Elaboración
Emitido por :	
Versión :	
Fecha:	
Aprobado por	
Fecha de Aprobación	

Figura 20 Información del documento.

Contenido

1. Introducción

- 1.1 Propósito (Documento)
- 1.2 Alcance
- 1.3 Tipo de Validación

2. Presentación del producto

- 2.1 Objetivo (Proyecto)
- 2.2 Alcance
- 2.3 Listado de Actores
- 2.4 Relación sistemas externos
- 2.5 No contempla
- 2.6 Restricciones y supuestos
- 2.7 Modelo de dominio
- 2.8 Reglas de Negocio

3. Descripción de Requerimientos

- 3.1 Listado de la funcionalidad del sistema
- 3.2 Diagramas de casos de uso
- 3.3 Descripción detallada de Casos de uso
- 3.4 Prototipo de navegación

4. Requerimientos no funcionales

- 4.1 Confiabilidad
- 4.2 Eficiencia
- 4.3 Mantenimiento
- 4.4 Portabilidad
- 4.5 Reusabilidad

5. Requerimientos de Interfaz

- 5.1 Interfaces de Usuario
- 5.2 Interfaces de Hardware
- 5.3 Interfaces de software
- 5.4 Interfaces de Comunicación
- 5.5 Restricciones de Diseño

6. Especificaciones Suplementaria

- 6.1 Componentes Comprados
- 6.2 Requerimientos de Licencia
- 6.3 Protocolo de Entrega
- 6.4 Observaciones
- 6.5 Anexos

Listado de la Funcionalidad del Sistema

Esta lista contendrá un número identificador del requerimiento, las iniciales del nombre del proyecto su número de identificación y hará referencia a su función, la prioridad del requerimiento, el nivel de complejidad de programación y el por qué es necesario o la funcionalidad que tendrá el requerimiento ejemplo Figura 21.

N°	Nombre del Caso de Uso	Prioridad	Complejidad	Necesidad

Figura 21 Ejemplo listado de la funcionalidad del sistema.

Diagrama de casos de uso y su descripción detallada

Esta sección del documento es bastante importante, en algunas ocasiones, este diagrama es reemplazado por una tabla de historias de usuario, pero para este trabajo en específico se eligió utilizar diagramas de casos de uso, proveniente del lenguaje UML, para mostrar en

forma muy gráfica de la interacción funcional del usuario con el sistema, en la figura 22 se muestra un ejemplo sencillo de un diagrama de casos de uso.

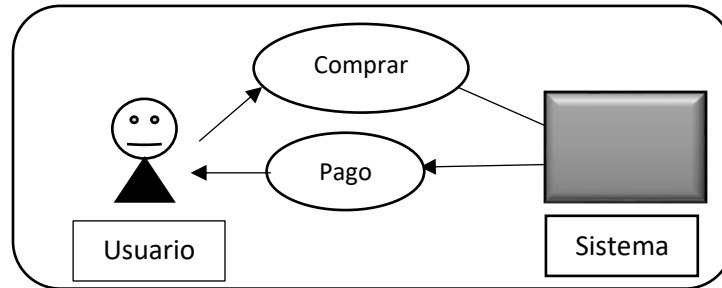


Figura 22 Ejemplo de un diagrama de casos de uso.

Un diagrama de casos de uso es una representación de las posibles acciones que realizará un usuario y la repuesta que dará el sistema a dicha acción. La herramienta propuesta para la creación de los diagramas de casos de uso es *yEd Graph Editor*. Un diagrama de casos de uso, es un potente editor de diagramas, que se puede utilizar de forma rápida y eficaz al momento de generar diagramas. Se pueden crear diagramas de forma manual o importar datos externos para el análisis y tramitar incluso grandes conjuntos de información con sólo pulsar un botón.

Tabla 2 Ejemplo para describir detalladamente un caso de uso.

Nombre del Caso de Uso:	
Actor Principal:	Actor Secundario:
Prioridad: <input type="checkbox"/> Necesario <input type="checkbox"/> Media <input type="checkbox"/> Deseable	
Complejidad: <input type="checkbox"/> Complejo <input type="checkbox"/> Medio <input type="checkbox"/> Simple	
Objetivo:	
Precondiciones:	
Éxito:	
Fracaso:	
Curso Normal	Alternativas
Observaciones:	
Autor:	Fecha Creación:
Autor Ultima Modificación:	Fecha Ultima Modificación:

Documentación desarrollada en SpiraTest

Para completar esta documentación preliminar el analista de requerimientos da inicio al proyecto en la herramienta llamada *SpiraTest* figura 23.

Los requerimientos ya en lista se escriben y guardan en la herramienta, se colocan algunos detalles como la prioridad el nombre y cuando debe ser iniciado, para que posteriormente al llevar un registro de actividad pueda imprimirse una matriz de trazabilidad para revisar el status de cada requerimiento, es posible también registrar las pruebas que se ejecutaran con cada requerimiento para obtener reportes de resultados con las pruebas.

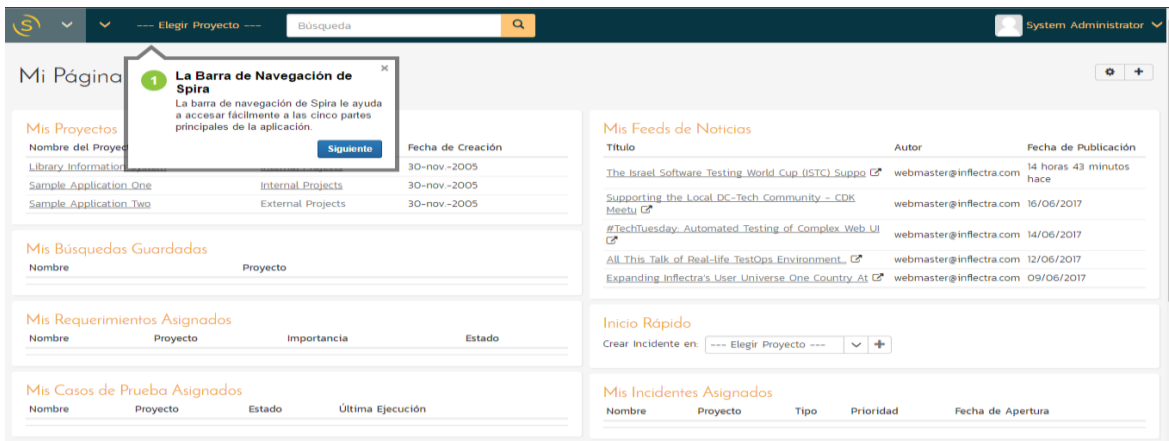


Figura 23 Pantalla inicial SpiraTest.

Recuperado de: Plataforma SpiraTest Requirements management (www.SpiraTest.com) (2017).

Capítulo 4 Sistema Gestor de Incidencias: Caso de Estudio.

4.1 Recolección de requerimientos

Para poner a prueba el modelo propuesto en este trabajo, se estableció un caso de estudio de proyecto interno en una empresa real, que contribuirá a detectar deficiencias o fallas en el proceso propuesto y establecer planes de riesgos para proyectos futuros.

Descripción

Un sistema de administración de incidencias que permita a los líderes de las áreas GPR (Gestión de Procesos) y QA (Aseguramiento de Calidad) llevar un registro de las incidencias que se encuentren en los procesos y auditorías dentro la empresa, permitiendo dos tipos de usuarios:

- ✓ Administradores
- ✓ Usuarios maestros
- ✓ Usuarios primarios

Administradores: Gestionan las bases de datos y poseen control de horas de accesos, así como bajas en el sistema.

Los usuarios Maestros: Contarán con una interfaz de usuario con permisos para seleccionar el tipo de incidencias a registrar:

Módulo de seguimiento de incidencias en el cual podrán cambiar el status, agregar observaciones, corroborar fechas de registros de incidencias con ayuda de un calendario que muestre datos sobre las incidencias activas, próximas a iniciar y fechas de finalización con distintos colores para facilidad de identificación.

El sistema permitirá

- Registrar
- Buscar
- Editar
- Eliminar

El sistema permitirá generar gráficas que muestren datos como:

- Registros de incidencias por Tiempo (Semanas, meses, años)
- Registros de incidencias por Auditorías
- Incidencias en proceso, detenidas, finalizadas
- Exitosas, fallidas.

El usuario Maestro podrá elegir el tipo de gráfica de tres opciones distintas.

Módulo para Generar Reportes, el cual mostrará una serie de opciones para que el usuario elija los datos que llevará su reporte y el formato en que ser impreso, además este deberá ser almacenado para su consulta digital posterior.

Módulo de seguimiento de usuarios cuya función será mostrar una lista de los usuarios registrados, y permitirá generar un reporte que contenga datos sobre qué usuario visitó el portal, en el rango de tiempo que elige el administrador.

Cerrar Sesión.

Los usuarios Primarios:

- ✓ Inicio de sesión
- ✓ Acceso al módulo de seguimiento de incidencias en el cual sólo tendrán permisos de búsqueda.
- ✓ Cerrar Sesión.

Permitir llevar un buen control de datos tanto para el Master user como para el primary user en el Sistema de Gestión de Reuniones.

Alcance:

- ✓ El sistema le debe permitir al administrador tener el control de las reuniones generadas de cada colaborador.
- ✓ El administrador tendrá la posibilidad de generar diferentes tipos de reportes de forma fija y dinámica.
- ✓ El sistema le permitirá al usuario generar reuniones.
- ✓ El sistema le permita al usuario generar reportes fijos.

Los usuarios administradores requieren que el sistema informe no solo todo el seguimiento de las incidencias si no también la atención que se le presta al sistema contando con una sección de informes de visitas. Así como un módulo generador de datos estadísticos referentes a incidencias y un módulo generador de reportes.

Como lo marca el modelo propuesto en cuanto a requerimientos el primer paso fue realizar una entrevista con el cliente en el cual se implementó el cuestionario propuesto, una plática previa sobre el entorno del futuro sistema y la realización de un sketch por parte del cliente, cuyos datos fueron analizados, posteriormente traducidos y clasificados en una lista de requerimientos, para dar inicio al primer acercamiento cliente- sistema desarrollando un prototipo navegable de bajo nivel que no incluía rasgos puntuales de diseño, se puntualizó que el prototipo incluía solo funcionalidad y no elementos puntuales de diseño por lo que la visión final del producto podría y será visualizado de diferente forma con la evolución en las distintas etapas de desarrollo cambiar.

Al igual que otros datos sobre el sistema, aparecen en el documento Especificación de requerimientos (establecido en la norma MoProSoft y CMMI), en la empresa en que se realizó el estudio, desarrollo y pruebas, se manejaba ya un estilo de documento cuyo contenido fue sometido a un análisis, el resultado permitió la modificación de la plantilla

establecida tratando de hacerlo más entendible organizado y retirando información repetitiva, este documento fue validado por los clientes directamente.

Como en todo proyecto de desarrollo de software, es preciso estar al tanto de que habrá cambios más probablemente en la etapa de desarrollo, el objetivo de este modelo no es erradicar por completo dichos cambios si no reducirlos y al igual que el impacto que estos podrían causar. Durante la primera reunión que se tuvo con el cliente, se le pidió que hablara sobre todo el entorno de su trabajo, que hace, como lo desarrolla y para qué fin, para estar totalmente familiarizados con el proceso que lleva el área en la cual debe desenvolverse el sistema. Ya que el llenado del test inicial fue previo a la reunión solo se tocaron algunos puntos referentes al test para aclarar algunas dudas, como horarios y otros puntos.

A continuación, se muestran los test contestados por los usuarios del sistema.

TEST

Por favor contesta las preguntas de la manera más clara posible, la información de este test será utilizada con la finalidad de desarrollar un sistema a fin a sus requerimientos.

1.- ¿Cuál es el departamento de la empresa al que está dirigido este proyecto?

Al área de Gestión de Procesos

2.- ¿Cuál es el objetivo de este proyecto?

El tener la administración, control y monitoreo de las incidencias que se registran en los proyectos y procesos de la organización, para dar soluciones a todas en tiempo y forma.

3.- ¿Cuál es el número aproximado de usuarios?

Aproximadamente 6 usuarios, inicialmente.

4.- ¿Está familiarizado con algún lenguaje o entorno de programación?

No, con ningún lenguaje.

5.- ¿Cuál es el sistema operativo más utilizado en la empresa que representa?

Equipo de cómputo (PC), Smartphone, iPhone e iPad acceso web o aplicación.

6.- ¿Cuál es la exigencia técnica (normas, y procedimientos a ser empleados y aplicados) de este proyecto?

El modelo CMMI y norma NMX-I-059-NYCE-2011.

7.- ¿Requisitos especiales de usuarios para este proyecto?

Que la administración y registro de la información sobre incidencias sea más ágil.

8.- ¿Tiene algún ejemplo o antecedentes que sirva como referencia?

Ninguno.

9.- ¿Quién será nuestro contacto directo?

Disponibilidad total.

10.- ¿Cuál será su disponibilidad?

Disponibilidad total.

11.- ¿Tendríamos acceso a información de la empresa?

La que sea considerada necesaria.

12.- Describa en sus palabras como visualiza el producto (Se otorgarán algunas hojas quizá lápices o cosas para realizar trazos).

Nota: Al terminar el llenado de este cuestionario, favor de guardarlo como un archivo PDF y reenviarlo a Beatriz Fernández Tamayo fernandezbeatriz@gmail.com

Figura 24 Test inicial del usuario de GPR

TEST

Por favor contesta las preguntas de la manera más clara posible, la información de este test será utilizada con la finalidad de desarrollar un sistema a fin a sus requerimientos.

1.- ¿Cuál es el departamento de la empresa al que está dirigido este proyecto?

De manera inicial a las áreas de Gestión de Procesos y al puesto de Aseguramiento de calidad, aunque posteriormente podría escalar a todas las áreas que requieran el registro de incidencias.

2.- ¿Cuál es el objetivo de este proyecto?

Llevar la administración y control de las incidencias generadas en auditorías o en los procesos, a fin de tener un seguimiento adecuado de las incidencias hasta su cierre (corrección).

3.- ¿Cuál es el número aproximado de usuarios?

6 usuarios, inicialmente.

4.- ¿Está familiarizado con algún lenguaje o entorno de programación?

Si.

5.- ¿Sobre qué dispositivos o sistemas operativos y lenguaje de programación contempla que sería más funcional para usted este proyecto?

Aquel que permita realizar un sistema basado en web, especialmente porque se han tenido sistemas de escritorio y no han podido explotarse en el empresa como se planeó inicialmente. Php, HTML5 además de que permita cumplir con requerimientos de usabilidad que le permita ser más fácilmente aceptado por los usuarios.

6.- ¿Cuál es la exigencia técnica (normas, y procedimientos a ser empleados y aplicados) de este proyecto?

El planteamiento y desarrollo del sistema deberá estar alineado a las normas Microsoft y CMMI-Dev. 2.

7.- ¿Requisitos especiales de usuarios para este proyecto?

Cumplir con los requerimientos definidos en las distintas entrevistas.

8.- ¿Tiene algún ejemplo o antecedentes que sirva como referencia?

De un sistema similar no.

9.- ¿Quién será nuestro contacto directo?

Tiempo de oficina en CIDT.

10.- ¿Cuál será su disponibilidad?

QA y GPR.

11.- ¿Tendríamos acceso a información de la empresa?

La que sea considerada necesaria.

12.- Describa en sus palabras como visualiza el producto (Se otorgarán algunas hojas quizá lápices o cosas para realizar trazos).

- Visualizo un sistema dividido por módulos, de lo general a lo particular, que permita la administración de las incidencias, es decir, que permita realizar las características básicas d un CRUD:

Alta de incidencias

Modificación de incidencias

Eliminación de incidencias

Consulta de incidencias

Además de disponer de un “calendario” o mecanismo que permita darle seguimiento a cada incidencia (por área) hasta su cierre.

Contar con un mecanismo de comunicación de las incidencias por área.

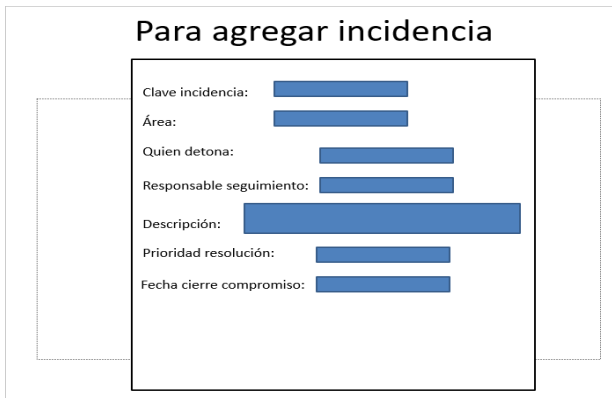
Nota: Al terminar el llenado de este cuestionario, favor de guardarlo como un archivo PDF y reenviarlo a Beatriz Fernández Tamayo fernandezbeatriz@gmail.com

Figura 25 Test inicial del usuario de QA.

Sketch

Una narrativa o sketch realizado completamente por nuestro cliente que será revisada durante la reunión, incluso el cuestionario podría ser enviado previamente al cliente para reducir el tiempo, puede ser complementado con una lluvia de ideas entre los participantes, ya que es una de las maneras más comunes de obtener conceptos de forma informal, es muy útil en muchas situaciones donde se requiere creatividad y un pensamiento cognitivo.

Se sugirió a los participantes del ejercicio que realizaran un sketch sencillo para establecer una imagen de la interfaz, a continuación, los ejemplos:



Para agregar incidencia

Clave incidencia:

Área:

Quien detona:

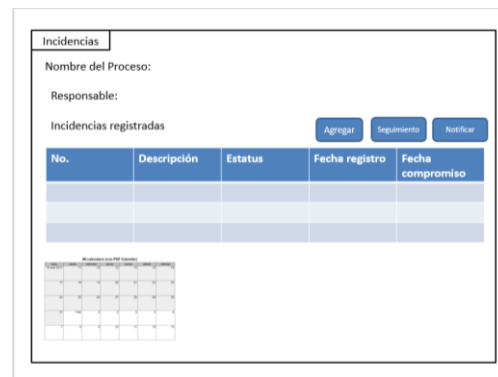
Responsable seguimiento:

Descripción:

Prioridad resolución:

Fecha cierre compromiso:

Figura 26 Sketch realizado por los usuarios pantalla de registro de incidencias.



Incidencias

Nombre del Proceso:

Responsable:

Incidencias registradas

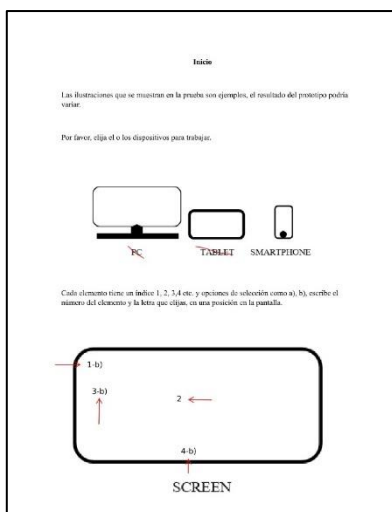
No.	Descripción	Estatus	Fecha registro	Fecha compromiso

Figura 27 Sketch realizado por los usuarios seguimiento de incidencias.

Durante el día fue enviado a los posibles usuarios el test de mapeo sobre pantalla blanca por correo, a continuación, los resultados.

A continuación, los resultados.

Prueba 1



Inicio

Las instrucciones que se muestran en la prueba son ejemplos, el resultado del prototipo podría variar.

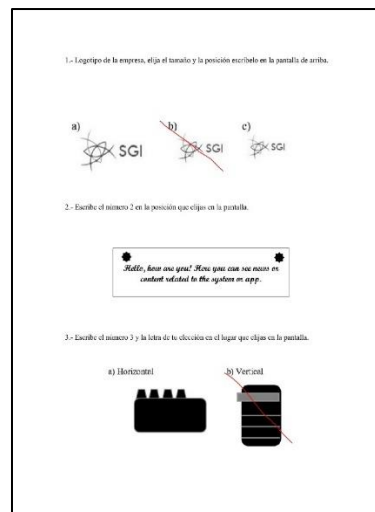
Por favor, elija el o los dispositivos para trabajar.

PC TABLET SMARTPHONE




Cada elemento tiene un índice 1, 2, 3, 4 etc. y opciones de selección como a), b), escriba el número del elemento y la letra que elija, en una posición en la pantalla.

1-b) 2 ← 3-b) ↑

SCREEN





1.- Logotipo de la empresa, elija el tamaño y la posición escribido en la pantalla de arriba.

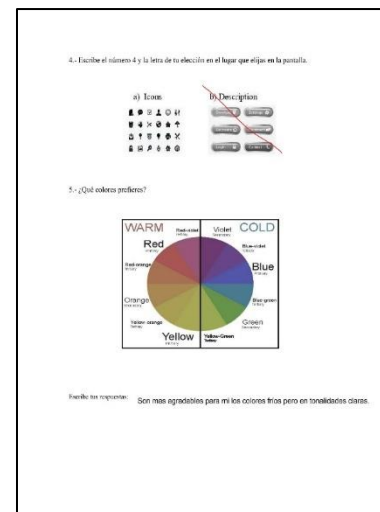
a)  b)  c) 

2.- Escriba el número 2 en la posición que elija en la pantalla.



Hello, how are you? How you can see news in content related to the system in app.

3.- Escriba el número 3 y la letra de la elección en el lugar que elija en la pantalla.

a) Horizontal  b) Vertical 



4.- Escriba el número 4 y la letra de la elección en el lugar que elija en la pantalla.

a) Línea  b) Descripción 

5.- ¿Qué colores prefiere?

WARM Red Yellow COLD Violet Blue

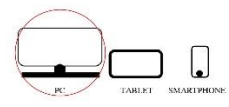
Preferir los colores: Son más agradados para mí los colores fríos pero en tonalidades claras.

Prueba 2

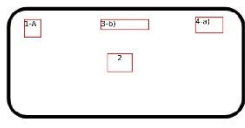
Inicio

Las ilustraciones que se muestran en la prueba son ejemplos, el resultado del prototipo podría variar.

Por favor, elija el o los dispositivos para trabajar:

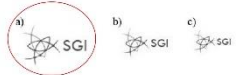


Cada elemento tiene un índice 1, 2, 3, 4 etc., y opciones de selección como a), b), escriba el número del elemento y la letra que elija, en una posición en la pantalla:

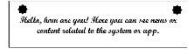


SCREEN

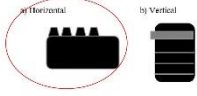
1.- Logotipo de la empresa, elija el tamaño y la posición escribiendo en la pantalla de arriba:




2.- Escriba el número 2 en la posición que elija en la pantalla:




3.- Escriba el número 3 y la letra de su elección en el lugar que elija en la pantalla:



4.- Escriba el número 4 y la letra de su elección en el lugar que elija en la pantalla:



5.- ¿Qué colores prefiere?



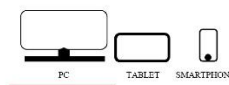
Por favor, responda: Prefiero menor cantidad de colores.

Prueba 3

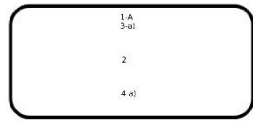
Inicio

Las Ilustraciones que se muestran en la prueba son ejemplos, el resultado del prototipo podría variar.

Por favor, elija el o los dispositivos para trabajar:

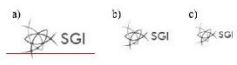


Cada elemento tiene un índice 1, 2, 3, 4 etc., y opciones de selección como a), b), escriba el número del elemento y la letra que elija, en una posición en la pantalla:

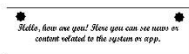


SCREEN

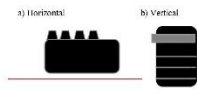
1.- Logotipo de la empresa, elija el tamaño y la posición, escribiendo en la pantalla de arriba:



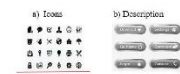
2.- Escriba el número 2 en la posición que elija en la pantalla:



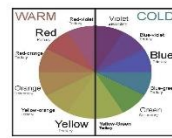
3.- Escriba el número 3 y la letra de su elección en el lugar que elija en la pantalla:



4.- Escriba el número 4 y la letra de su elección en el lugar que elija en la pantalla:



5.- ¿Qué colores prefiere?



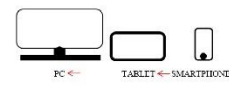
Escriba las respuestas: Creo que los colores que tiene el logo estarían bien, pero que luzca unificados.

Prueba 4

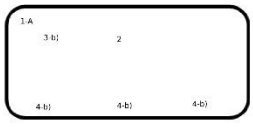
Inicio

Las Ilustraciones que se muestran en la prueba son ejemplos, el resultado del prototipo podría variar.

Por favor, elija el o los dispositivos para trabajar:

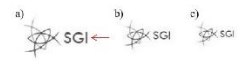


Cada elemento tiene un índice 1, 2, 3, 4 etc., y opciones de selección como a), b), escriba el número del elemento, el tamaño y la letra que elija, en una posición en la pantalla:

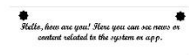


SCREEN

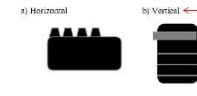
1.- Logotipo de la empresa, elija el tamaño y la posición escribiendo en la pantalla de arriba:




2.- Escriba el número 2 en la posición que elija en la pantalla:



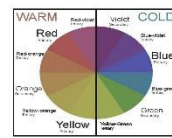
3.- Escriba el número 3 y la letra de su elección en el lugar que elija en la pantalla:



4.- Escriba el número 4 y la letra de su elección en el lugar que elija en la pantalla:

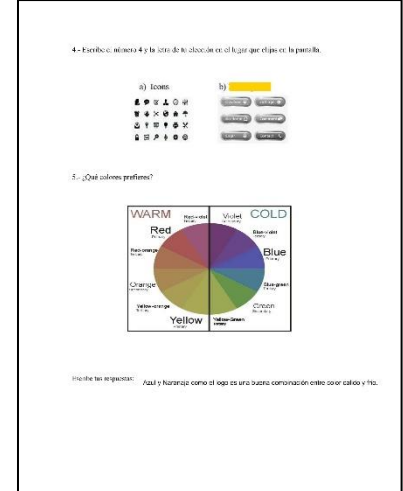
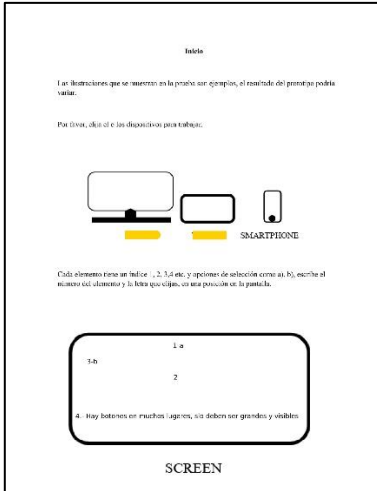


5.- ¿Qué colores prefiere?



Por favor, responda: Colores algunos como verde y amarillos, pero como que en tonalidad de azul ya que el logo de la empresa es así.

Prueba 5



Las pruebas arrojaron resultados muy puntuales en cuanto a preferencias de diseño y contribuyeron al equipo a desarrollar el diseño con más rapidez, cabe resaltar que al ejecutar estas pruebas con 5 personas contribuyo a no tener resultados empatados, es decir para cada pregunta se obtuvo una respuesta de inclinación clara.

Tabla 3 Resultados de las pruebas de mapeo sobre pantalla blanca.

Prueba		1	2	3	4	5
1	PC Tablet	B Esquina Superior Izquierda	Centro	B Centro Izquierda	B Centro Inferior	Colores Fríos
2	PC	A Esquina superior Izquierda	Centro	A Centro superior	A Esquina superior derecha	Pocos colores
3	PC	A Centro Superior	Centro	A Centro	A Centro Inferior	Azul y Naranja
4	PC Tablet	A Esquina Superior Derecha	Centro	B Centro Izquierda	B Centro Inferior	Azul y Naranja
5	PC Tablet	A Centro Superior	Centro	B Esquina Superior Izquierda	B Centro Inferior	Azul y Naranja

Como resultado se obtuvo que el sistema gestor de incidencias será lanzado para PC y Tablet, con imágenes de tamaño considerable, contenidos y actualizaciones irán al centro de la pantalla el menú de navegación esta posicionado en la parte superior de la pantalla debajo del banner con diseño por pestañas, los botones deben ser descriptivos y se encontraran

debajo de cada sección dependiendo de su función, por ultimo la paleta de colores seleccionada deberá combinara un color cálido y uno frio, como esta diseñado el logo de la empresa para unificar este módulo con los ya desarrollados.

4.2 Desarrollo de Documento de Establecimiento de Requerimientos

Lista de Funcionalidad de SAI

A continuación, se muestra la lista de funcionalidad obtenida para el Sistema Gestor de Incidencias.

Tabla 4 Listado de funcionalidad.

N°	Nombre del Caso de Uso	Prioridad	Complejidad	Necesidad
1	AI_001_Inicio_Sesión	Esencial	Simple	El sistema debe contar con un módulo de inicio de sesión, el cual debe realizar la validación del usuario que acceso; el usuario debe existir en la base de datos de los empleados registrados con status activos.
2	AI_002_Recuperar_Contraseña	Útil	Medio	El sistema debe tener un módulo de recuperación de contraseña que permita al usuario la modificación de contraseña en caso de extravió u olvido para re ingresar al sistema
3	AI_003_Crear_Cuenta	Esencial	Simple	El sistema debe contar con un módulo de creación de cuenta que permita a usuarios invitados ingresar a ver los informes de incidencias de acuerdo a su área, y permita tener un control de visitas a los administradores.
4	AI_004_Registro_Incidencia	Esencial	Medio	El sistema debe contener un módulo de registro de incidencia en que se inserten datos sobre la incidencia y esta sea guardada.
5	AI_005_Consulta_Incidencia	Esencial	Medio	El sistema debe contener poder contener módulo de consulta,

				que permita la edición de las mismas
6	AI_006_Editar	Esencial	Medio	El sistema debe permitir editar incidencias si el administrador lo considera prudente
7	AI_007_Eliminar	Útil	Medio	El sistema debe permitir eliminar incidencias si el usuario así lo desea
8	AI_008_Seguimiento_Incidencia	Esencial	Medio	El sistema debe contar con modulo que muestre todas las incidencias registradas y algunos datos (estatus, fecha, tipo).
9	AI_009_Generar_Gráficos	Esencial	Complejo	El sistema debe contener un módulo de generación de gráficas estadísticas que permitan visualizar el avance sobre las incidencias.
10	AI_010_Generar_Reportes	Esencial	Complejo	El sistema debe contener un módulo de generación de reportes, en el cual los administradores podrán elegir los datos que se manejaran en el reporte y será impreso en PDF.
11	AI_011_Cosultar_Usuarios	Esencial	Complejo	El sistema debe permitir a ls administradores consultar una lista de todos los usuarios registrados en el mismo.
12	AI_012_informe_visitas	Esencial	Medio	El sistema debe permitir al usuario generar informes de las visitas en rango de fechas que el determine.
13	AI_013_Cerrar_Sesión	Esencial	Medio	Cierre de sesión de Admins y Usuarios para seguridad de información
5	AI_005_Consulta_Incidencia	Esencial	Medio	El sistema debe contener poder contener módulo de consulta, que permita la edición de las mismas
6	AI_006_Editar	Esencial	Medio	El sistema debe permitir editar incidencias si el administrador lo considera prudente

Ejemplo de un check para detallar un requerimiento, esto contribuye a que el equipo piense en todos los caminos posibles que podría tomar un requerimiento.

Tabla 5 Descripción detallada de casos de uso de SAI.

Nombre del Caso de Uso: AI_001_Inicio_Sesión	
Actor Principal: GPR	Actor Secundario:
Prioridad: <input checked="" type="checkbox"/> Necesario <input type="checkbox"/> Media <input type="checkbox"/> Deseable	
Complejidad: <input checked="" type="checkbox"/> Complejo <input type="checkbox"/> Medio <input type="checkbox"/> Simple	
Objetivo: Ingreso al sistema	
Precondiciones: Iniciar sesión	
Éxito: Iniciar correctamente Sesión y entra al sistema	
Fracaso: No se puede entrar al sistema	
Curso Normal	Alternativo
Este caso de uso inicia cuando el usuario decide iniciar sesión en el sistema.	Los datos ingresados no son validos El usuario no está registrado
Si está registrado ingresara a pantalla principal	El ingreso falla y el sistema manda mensaje de error.
Si es Administrador puede registrar incidencias Editarlas, Eliminarlas.	Los datos ingresados no son validos El usuario no está registrado
Si es un usuario normal podrá consultar las incidencias y realizar reportes y gráficos	
Este caso de uso inicia cuando el usuario decide iniciar sesión en el sistema.	
Observaciones:	
Autor: Analista	Fecha Creación: 20/08/17

Autor Ultima Modificación:

Fecha Ultima Modificación: 02/09/17

Con ayuda de los datos de la lista de funcionalidad y descripción detallada de requerimientos el equipo realizó un diagrama de casos de uso que se muestra a continuación:

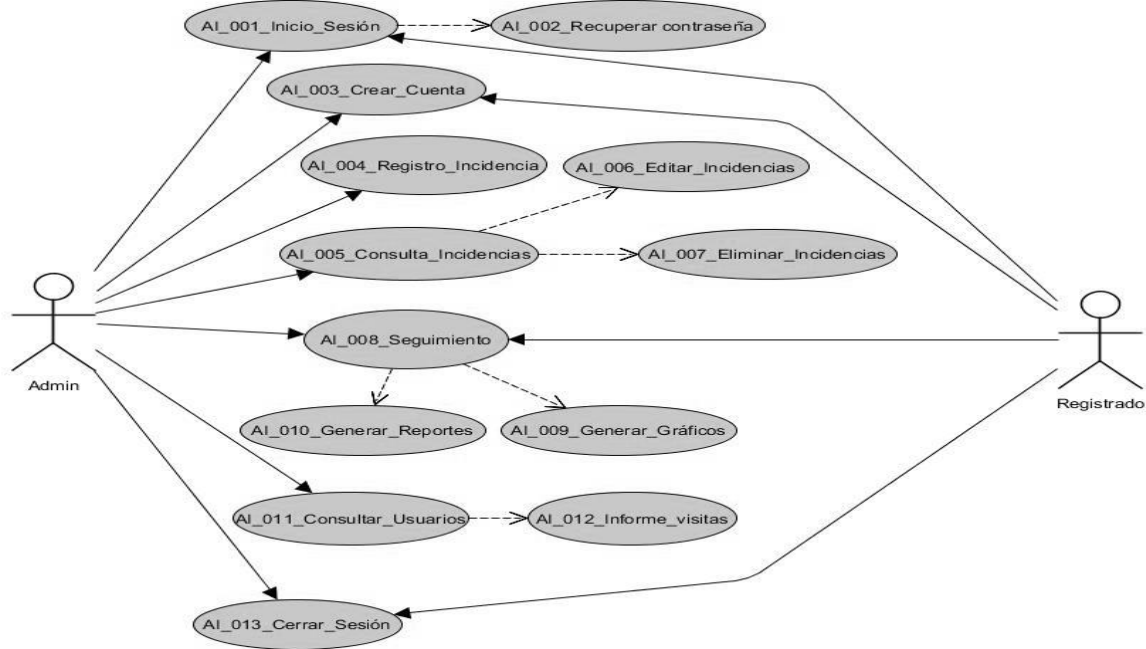


Figura 28 Diagrama de casos de uso de SAI.

Así como el Diagrama BPMN para explicar mejor el proceso del sistema:

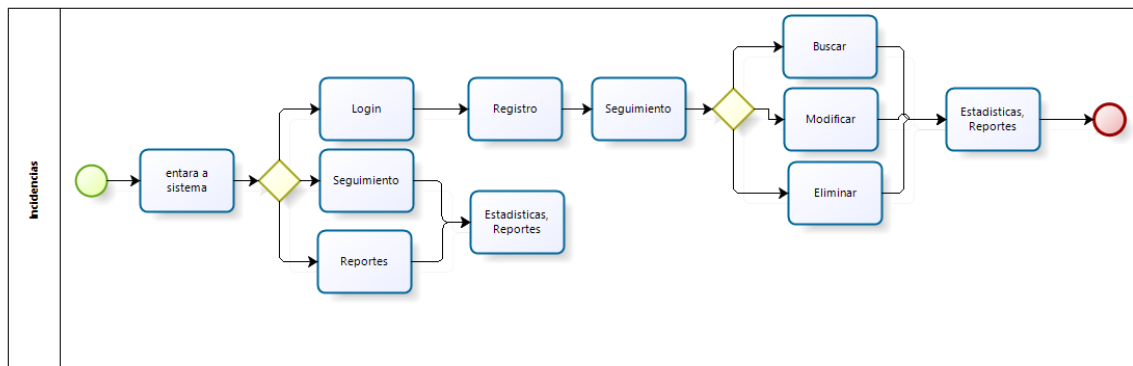


Figura 29 Diagrama BPMN de SAI.

Con ayuda de la lista de funcionalidad el analista puede ingresar los datos en la herramienta sugerida *SpiraTest*, para obtener reportes como matriz de trazabilidad y reportes de requerimientos, a continuación, un ejemplo de un documento obtenido en dicha herramienta.

Req #	Name	Type	Status	Release	Real Release	Requirements Traceability
9	RQ37	GPR_Ingresar a sistema	Use Case	Accepted	20/10/2017	20/10/2017
10	RQ46	GPR_Registro de incidencia	Use Case	Accepted	20/10/2017	21/10/2017
11	RQ47	GPR_Especificaciones incidencias	Use Case	Under Review	20/10/2017	20/10/2017
12	RQ48	GPR_Actualizaciones	Use Case	Under Review	20/10/2017	20/10/2017
13	RQ49	GPR_Consultas	Use Case	Accepted	20/10/2017	20/10/2017
14	RQ50	GPR_Eliminar	Use Case	Under Review	20/10/2017	22/10/2017
15	RQ51	GPR_Especificaciones_Reportes	Use Case	Accepted	20/10/2017	22/10/2017
16	RQ52	GPR_Generar_Reportes	Use Case	Accepted	20/10/2017	20/10/2017
17	RQ53	QA_Ingresar a sistema	Use Case	Accepted	20/10/2017	20/10/2017
18	RQ54	QA_Registro de incidencia	Use Case	Accepted	20/10/2017	20/10/2017
19	RQ55	QA_Especificaciones incidencias	Use Case	Under Review	20/10/2017	20/10/2017
20	RQ56	QA_Actualizaciones	Use Case	Under Review	20/10/2017	20/10/2017
21	RQ57	QA_Consultas	Use Case	Accepted	20/10/2017	20/10/2017
22	RQ58	QA_Eliminar	Use Case	Under Review	20/10/2017	20/10/2017
23	RQ59	QA_Especificaciones_Reportes	Use Case	Accepted	20/10/2017	20/10/2017
24	RQ60	QA_Generar_Reportes	Use Case	Accepted	20/10/2017	20/10/2017

Figura 30 Matriz de trazabilidad de requerimientos básica con datos exportados desde SpiraTest.

Req #	Name	Description	Type	Priority	Status	Author	Owner	Creation Date	Test Coverage
7	37	AI_001_Inicio_Sesión	Use Case	2 - High	Planned	System Administrator		16/10/2017	0Covering,0Failed,0Passed,0Blocked,0Cau
8	38	AI_002_Recuperar_Contraseña	Need	2 - High	Planned	System Administrator		16/10/2017	0Covering,0Failed,0Passed,0Blocked,0Cau
9	39	AI_003_Crear_Cuenta	Use Case	2 - High	Requested	System Administrator		16/10/2017	0Covering,0Failed,0Passed,0Blocked,0Cau
10	40	AI_004_Registro_Incidencia	Use Case	2 - High	Requested	System Administrator		16/10/2017	0Covering,0Failed,0Passed,0Blocked,0Cau
11	41	AI_005_Consulta_Incidencia	Use Case	2 - High	Requested	System Administrator		16/10/2017	0Covering,0Failed,0Passed,0Blocked,0Cau
12	42	AI_006_Editar	Use Case	2 - High	Requested	System Administrator		16/10/2017	0Covering,0Failed,0Passed,0Blocked,0Cau

Figura 31 Lista de requerimientos generada en la plataforma SpiraTest.

Una vez ingresados los requerimientos la plataforma genera de manera automática un resumen detallado de requerimientos (especificación de requerimientos), Plan de requerimientos (Levantamiento de requerimientos), Resumen de requerimientos (lista de

requerimientos) y Matriz de trazabilidad. De igual forma *SpiraTest* permite el desarrollo y seguimiento de planes de prueba para el desarrollo del proyecto.

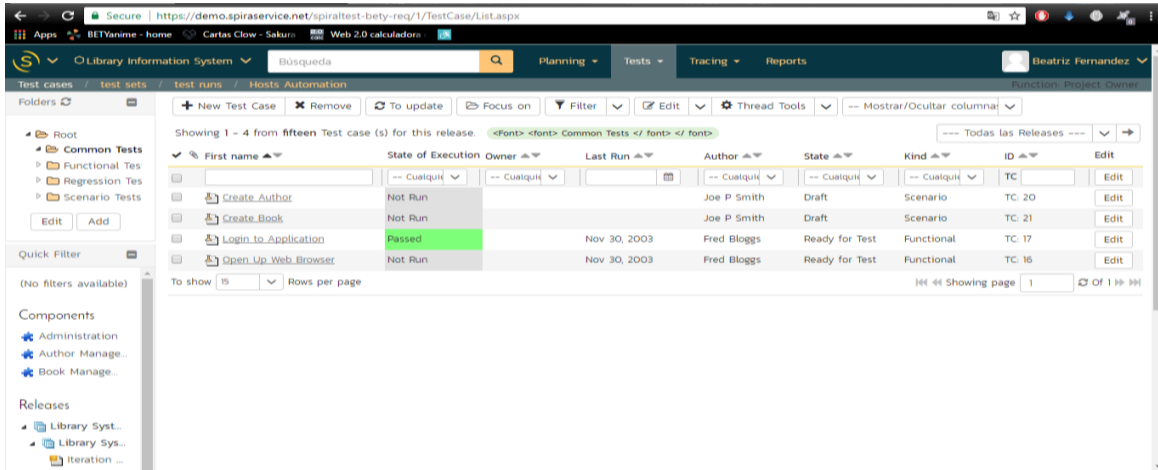


Figura 32 Pantalla de seguimiento de requerimientos de SpiraTest.

Reporte sobre requerimientos generado con la herramienta con cambios generados por el avance y correcciones del proyecto *SpiraTest*.

Requirements Detailed Report

This report displays all of the requirements defined for the current project in the order they appear in the requirements list. The requirement's details and coverage status are displayed, along with sub-tables containing the list of covering test cases, linked incidents/requirements, attached documents, associated tasks, linked artifacts and the change history

Project5:Reportes de Incidencias

Sistema de control de incidencias, arrojados por departamento y por auditorias en procesos.

RQ:37-GPR_Ingresar a sistema

Type:	Use Case	Component:	
Priority:	2 - High	Status:	Accepted
Author:	System Administrator	Creation Date:	14-Aug-2017
Coverage:	0Covering,0Failed,0Passed0B	Last Modified:	28-Aug-2017
Owner:		Estimate:	1.0
Release #:		Est. Effort:	8.00
Task Est. Effort:		Task Actual Effort:	
Task Remaining Effort:		Task Projected Effort:	

Requirement Change History:

Change Date	Field Name	Old Value	New Value	Changed By
28-Aug-2017	Requirement Name	Login	GPR_Ingresar a sistema	System Administrator
28-Aug-2017	Status	Requested	Accepted	System Administrator
14-Aug-2017	Type	Feature	Use Case	System Administrator
14-Aug-2017	Importance		2 - High	System Administrator
14-Aug-2017	Requirement Name	Nuevo Requisito	Login	System Administrator

Incluye el nombre del requerimiento

Información sobre cambios, quien los realiza y cuando.

Figura 33 Reporte detallado de requerimientos.

Una vez establecidos y aprobados los requerimientos en una reunión, el equipo dio paso a la creación del prototipado, este fue realizado bajo los resultados de los test y descripciones de los usuarios.

4.3 Prototipado

Una vez aprobadas, la lista de requerimientos, BPMN y Casos de uso procede realizar una última prueba, un prototipo navegable simple que permita una interacción inicial para el cliente, desarrollado en Balsamiq Mockups. En las figuras 32 a 36 se muestran las pantallas de prototipo.



Figura 34 Prototipo de pantalla de registro.

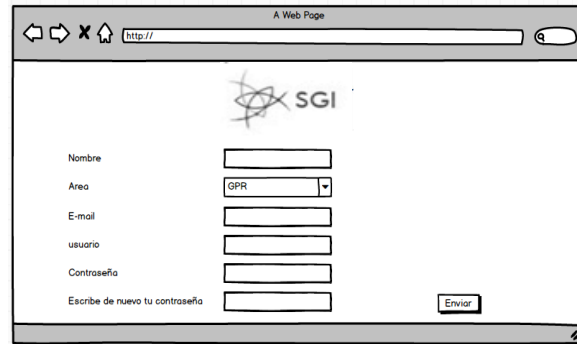


Figura 35 Prototipo de pantalla de registro de incidencia.

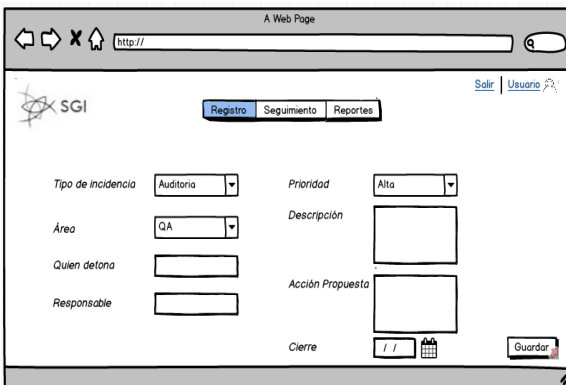


Figura 36 Prototipo de pantalla Registro de incidencia

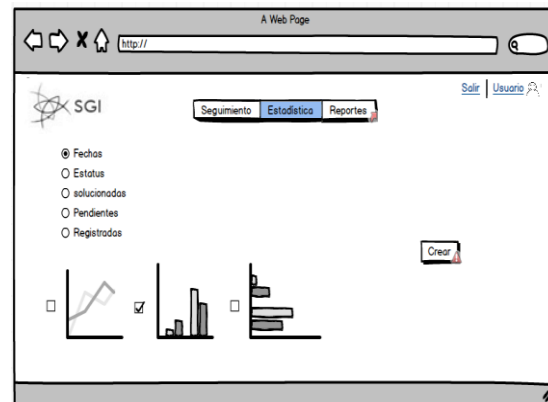


Figura 37 Prototipo de pantalla de reportes.

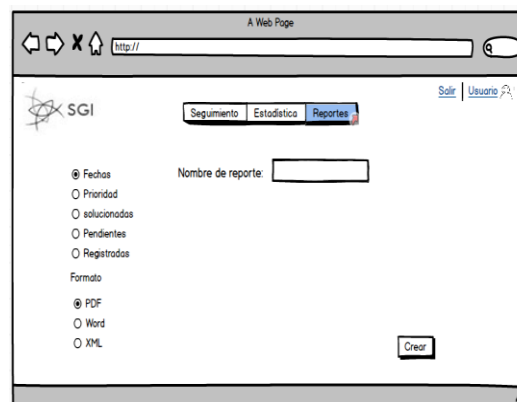


Figura 38 Prototipo de pantalla de búsqueda.

El diseño inicial fue sencillo ya que los datos brindados por los usuarios marcaron la preferencia por colores y artefactos de navegación.

Una vez que el documento de requerimientos fue aceptado, se realizó un reporte de validación para asegurar que se cumplió con la mayoría o en su totalidad los requisitos señalados.

Resultados de la Validación de la Especificación de Requisitos

Tabla 6 Registro recuperado del reporte de validación de la empresa.

Reporte de Validación 1 de Desarrollo y Mantenimiento de Software.			
Especificación de Requisitos			
Participantes:			
Nombre		Puesto	
Rosa Isaura Mejorada Sosa, Carolina Roció Sánchez Pérez.		Cliente	
Beatriz Fernández Tamayo		CIDT	
Elemento a validar	Si	No	Observaciones
Introducción	x		
Propósito	x		
Alcance	x		
Definiciones, Acrónimos y Abreviaturas		x	
Audiencia		x	
Referencias		x	
Presentación del producto	x		
Propósito del sistema	x		
Restricciones y Supuestos	x		
Descripción General	x		
Listado de la funcionalidad del Sistema	x		
Diagramas de Caso de Uso	x		
Listado de Actores	x		
Perspectiva del producto	x		
Modelo de Dominio	x		

Descripción detallada de requerimientos	x		
Requerimientos Funcionales	x		
Reglas y Funciones de Negocio	x		
Requerimientos No Funcionales	x		
Requerimientos de Interfaz	x		
Interfaces de Usuario	x		
Interfaces de Hardware	x		Solo fue mostrado el prototipo en escritorio nos gustaría ver como se mostraba en el otro dispositivo.
Interfaces de software	x		
Interfaces de comunicación	x		
Especificaciones Suplementarias		x	
Restricciones de diseño	x		
Elementos impactados	x		
Requerimientos de licencia		N/A	

Capítulo 5 Resultados

En el periodo de piloteo del modelo de estrategias de mejora para administración de requerimientos centrado en el usuario, se analizaron las fortalezas y fallas del ya mencionado modelo, el caso de estudio en la empresa:

Como en todo proyecto de desarrollo de software, es preciso estar al tanto de que habrá evoluciones probablemente en la etapa de desarrollo, el objetivo de este modelo no fue erradicar por completo dichos cambios si no reducirlos al igual que el impacto que estos hubieran causado.

Específicamente en este proyecto los requerimientos están muy puntuales por parte de los clientes y resultado ser un proyecto sencillo, no se hallaron cambios considerables durante el desarrollo, las normativas de calidad CMMI y Moprosoft puntualizan también documentos llamados “Matriz de trazabilidad” y “Plan de pruebas”, documentos realizados con la ayuda de la herramienta SpiraTest free versión en su forma más básica sin embargo los documentos pueden ser modificados por el analista para agregar datos requeridos.

Tabla 7 Observaciones de resultados.

<i>Actividad</i>	<i>Instrumento</i>	<i>Tiempo estimado</i>	<i>Tiempo real</i>	<i>Resultados</i>	<i>Observaciones</i>
Entrevista	Lluvia de ideas. Grabación de la reunión	1 hr. (Reunión)	30 min. (Reunión)	Obtención previa de datos referentes al entorno del proyecto	Proceso de elicitación que satisface la necesidad del equipo de conocimiento sobre el usuario y sus necesidades principales
Levantamiento de requerimientos	Test Elaboración de Sketch Test Pantalla blanca	1 hr. (Reunión)	1hr. (Reunión)	Recolección de requerimientos funcionales y de diseño en una sola reunión permitiendo la intervención del cliente más a fondo.	Aunque estas técnicas podrían ser al principio tediosas contribuyen mucho a un mejor entendimiento y temprano diseño facilitando el trabajo de análisis.

Tabla 5 Observaciones de resultados.

<i>Actividad</i>	<i>Instrumento</i>	<i>Tiempo estimado</i>	<i>Tiempo real</i>	<i>Resultados</i>	<i>Observaciones</i>
Análisis	Traducción de datos en requerimientos. Definición de viabilidad, alcances y limitaciones	2 días	3 días	Análisis de la información obtenida durante la reunión para establecer ideas y plan de actividades.	Tiempo que se brinda al equipo para traducir todos los datos recabados durante la reunión anterior para traducirlos en listas de requerimientos con ayuda de herramientas de modelado de sistema.
Desarrollo de prototipo	Balsamiq Mockups Método de pantalla blanca	3 días	3 días	Desarrollo de un prototipo de bajo nivel de diseño que permita la navegación o recorrido para el cliente.	La propuesta de un prototipo de bajo nivel de diseño pero que permita la navegación del usuario o cliente, genera que tenga más confianza en el proyecto, se especifica al cliente que el diseño visual puede variar si es su gusto y dependiendo de la evolución del desarrollo, pero la iteración será lo más fiel al prototipo posible.
Especificación de requerimientos	Plantillas de función casos de uso. Diagrama BPMN (Bizaggi).	4 días	5 días	El documento fue desarrollado con la ayuda de las herramientas propuestas Bizaggi, yED Graph, plantilla Funcionalidad.	El documento establecido en MBN sufrió una serie de modificaciones sencillas que pretenden facilitar un poco el proceso de llenado del mismo

Tabla 5 Observaciones de resultados.

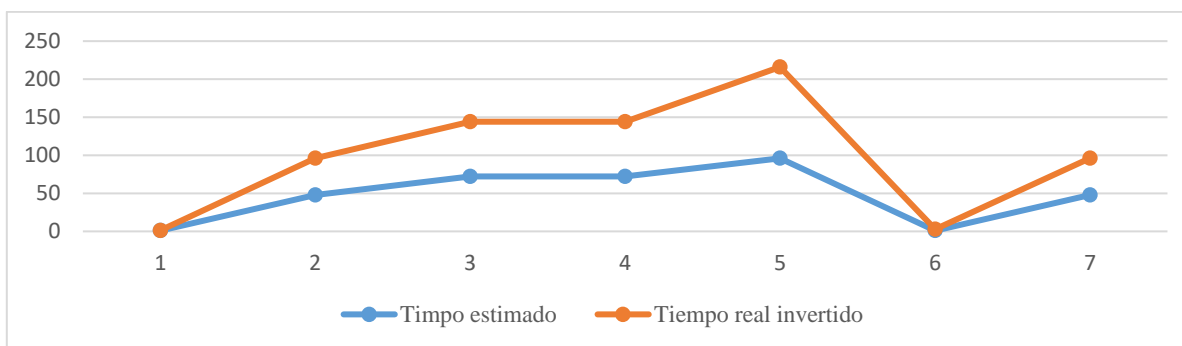
<i>Actividad</i>	<i>Instrumento</i>	<i>Tiempo estimado</i>	<i>Tiempo real</i>	<i>Resultados</i>	<i>Observaciones</i>
Validación	Reunión para revisión de requerimientos. Reporte de validación Firmas, Correo Etc.	1 hora (Reunión)	2 horas	Durante la reunión fue posible puntualizar muy rápidamente los cambios necesarios y fue más sencillo para el cliente y el equipo comprender el resultado deseado de cada módulo.	La tendencia de todos los modelos nuevos de desarrollo de software es incluir al cliente en cada paso que el equipo de por esta razón es precisa su aprobación desde los requerimientos para continuar a la fase de desarrollo, sin su aprobación escrita no hay avance a otra fase.
Verificación					Es precisa la supervisión del equipo para monitorear que se esté realizando lo que se dijo y como se dijo, en caso de haber cambios en el modelo especificar los porque para su posterior uso
Plan de Pruebas	Especificación de requerimientos	2 días	2 días	Elaborado en base con la lista de requerimientos aprobada por el revisor y el cliente.	Para atender los objetivos de la calidad en un desarrollo de sistemas, encargándose de definir aspectos como módulos o funcionalidades.
Otros documentos y seguimiento	<i>SpiraTest</i>	NA	NA	Otros documentos como la lista de requerimientos, matriz de trazabilidad son generados automáticamente en SpiraTest en su forma básica los documentos pueden ser modificados por el analista.	spiraTest facilita el seguimiento de requerimientos con sus cambios incidencias y pruebas y genera automáticamente algunos de los documentos como matriz de trazabilidad (simple), lista de requerimientos, pruebas etc.

Durante la reunión fue posible puntualizar muy rápidamente los cambios necesarios y fue más sencillo para el cliente y el equipo comprender el resultado deseado de cada módulo.

A continuación, se muestran los datos obtenidos durante el piloteo del modelo especificación de requerimientos centrado al usuario, sobre el proyecto “Sistema Gestor de incidencias”.

Tabla 8 Comparación de tiempo invertido con el modelo propuesto vs otros modelos en proyectos pasados.

<i>Actividades</i>	<i>Tiempo estimado</i>	<i>Tiempo real Invertido</i>
1.- Entrevista	1 hr.	30 min.
2.-Elicitacion de requerimientos	1 hr.	1 hr.
3.- Análisis	2 días	3 días
4.- Desarrollo de prototipo	3 días	3 días
5.- Especificación de requerimientos	3 días	3 días
6.- Validación	4 días	5 días
7.- Verificación	1 hr.	2 horas.
8.- Plan de Pruebas	2 días	2 días

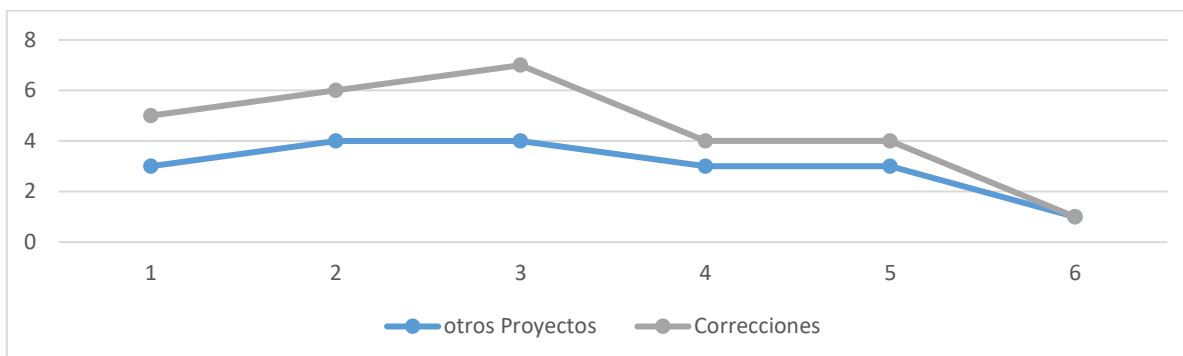


Gráfica 1 Comparación de tiempo invertido con el modelo propuesto vs otros modelos en proyectos pasados

Si bien no fue posible reducir el tiempo, lo gramos no exceder demasiado el tiempo establecido en proyectos anteriores, pero con mejores resultados.

Tabla 9 Relación de correcciones requeridas por actividad.

<i>Actividades</i>	<i>Experiencias de proyectos pasados</i>	<i>Piloteo de modelo Especificación de requerimientos centrado en el usuario</i>
1.- Levantamiento de requerimientos.	3	2
2.- Análisis	4	2
3.- Desarrollo de prototipo	4	3
4.- Especificación de requerimientos	3	1
5.- Validación	3	1
6.- Verificación	1	0



Gráfica 2 Demostrativa de decremento de correcciones por actividad.

Se encontró que la tasa de correcciones se redujo de forma positiva y el proyecto tuvo mejor fluidez con respecto de los proyectos pasados.

Capítulo 6

Conclusiones y Trabajos Futuros

Durante las pruebas del modelo y estrategias propuestas en este trabajo surgieron algunos aspectos a tratar y analizar, por ejemplo:

Sobre la organización y asignación de roles, resultó ser un tanto difícil debido a que el equipo designado para el piloteo del proyecto fue pequeño; obligando al equipo a emplear un poco más de tiempo en algunas tareas, cabe mencionar que las herramientas utilizadas para especificar los requerimientos son en su mayoría de licencia libre, exceptuando a *SpiraTest management requirements*, por lo que la utilización o exclusión de la misma se deja a criterio del ocupante del modelo del modelo ERCU (Especificación de requerimientos centrado en el usuario), de no utilizar *spiraTest* podría verse reflejado un aumento de tiempo en cuanto a llenado de documentación.

Después del análisis de información, los productos finales serán diseñados con ayuda de la información recabada con los *test* propuestos en el Capítulo 3 sección 3.2, estos inicialmente están pensados para ponerse a prueba con un número de personas mayor a 5 para que puedan arrojar datos estadísticos que pueden ser más útiles, es decir de 5 pruebas tres marcaron una misma opción, y así sucesivamente dando una idea de un mapa mental de la mayoría de los usuarios finales, en el caso del proyecto “sistema gestor de incidencias” que se describe en el capítulo 4, sólo eran necesarios dos “Súper-Usuarios” cuyos departamentos eran QA (Calidad) y GPR (Gestión de Procesos), esto represento un problema al momento de pedir que fuera trazado un sketch o boceto del recorrido e interfaz del sistema teniendo un resultado muy visual (QA) y el otro muy lógico (GPR), para lo cual se solicitó que fuera el encargado de GPR el primero en poner a prueba el prototipo navegable en solitario sin exponerlo previamente al diseño propuesto por el encargado de QA, para asegurar que el sistema fuera intuitivo. No hubo mayor complicación para el usuario GPR al usar el prototipo por lo tanto no fueron necesarios los replanteos en ese momento.

Como una observación importante en el proceso elicitación de requerimientos, se detectó una significativa diferencia de conocimiento informático entre ambos usuarios del sistema, lo que provocó un ligero aislamiento de uno de ellos, esto fue solucionado empleando técnicas como *test* de mapeo sobre pantalla blanca que observarse en el Capítulo 4 y una sesión (reunión) tomada en cuenta como un test de usabilidad, se notó un interés creciente en dicho usuario a expresar las ideas y necesidades que poseía sobre el sistema, esto hizo más amano el ambiente en las reuniones empleando un poco más lenguaje que fuera más accesible para ambos usuarios.

En el proceso de llenado de documentos se detectaron algunas deficiencias en cuanto a la agrupación de elemento fue analizada y como resultado se estableció una nueva plantilla Capítulo 3 sección 3.2.3 figuras 19 y 20.

En el proceso de análisis y toma de requerimientos enfocados al usuario se espera también brindar mayor confianza a nuestro cliente, interactuando de manera acoplada a él y su conocimiento en la materia, contribuyendo a un mejor entendimiento, resolución a problemas y necesidades del cliente.

De manera especial podemos concluir que:

- 1.- Incluir una etapa de elicitación en el proceso Capítulo 3 Figura 14 “Modelo Recolección de requerimientos centrado en el usuario” satisface la necesidad del equipo de conocimiento sobre el usuario y sus necesidades principales, las técnicas propuestas a emplearse pueden resultar ligeramente tediosas al inicio sin embargo proveen información a los analistas, diseñadores y desarrolladores sobre el verdadero objetivo de cada sistema o proyecto.
- 2.- Un prototipo permitirá la navegación del usuario final, lo que genera más confianza en el proyecto, es preciso especificar que el diseño visual puede variar si es su gusto y dependiendo de la evolución del desarrollo, pero la iteración
- 3.- La tendencia de todos los modelos nuevos de desarrollo de software es incluir al cliente en cada paso que el equipo dé, por esta razón se precisa una aprobación desde los requerimientos para continuar a la fase de desarrollo, sin una aprobación escrita no hay avance a otra fase.
- 4.- Con el desarrollo de software enfocado en la experiencia de usuario las pequeñas y mediana empresas rescatan, los intereses iniciales y primordiales de todo proyecto realizando sistemas más intuitivos aumentando los niveles de usabilidad.
- 5.- De acuerdo a los resultados obtenidos Capítulo 5 el modelo propuesto podría generar un aumento de inversión de tiempo que tendrá que ser estima desde la planificación del proyecto, en este caso no hubo evoluciones mayores y no fue necesaria petición de cambios, al ser un proyecto de dimensiones reducidas el prototipo fue sencillo de realizar con ayuda de *Balsamiq mockups*, no se garantiza que una total eliminación de evoluciones.
- 6.- Las correcciones especificadas en el Capítulo 5 tabla 7 son referentes tanto a errores de llenado de artefactos como evoluciones del sistema en sí.

Conclusión Principal

Un modelo estratégico de mejoras de recolección de requerimientos centrado en el concepto de experticia de usuario, brinda una mejor comunicación, desarrollo de documentos y mayor calidad y usabilidad al resultado final.

Trabajos Futuros

Este proyecto tiene la capacidad de mejorar adaptándolo a diferentes proyectos con distintos niveles de alcance, así como desarrollando o utilizando una plataforma semejante a *SpiraTest* para la automatización de los requerimientos y la inclusión de guías más amplias que contengan más información sobre los roles requeridos para el desarrollo de este modelo.

Otra sugerencia es adaptar más la tecnología de escenarios virtuales en caso de que sea utilizado para proyectos que contemplan un resultado más tangible o incluso el desarrollo de videojuegos que es la rama del desarrollo en que más se destaca el enfoque de experiencia de usuario.

Para este caso de estudio en especial el número de participantes para obtener los requerimientos fue limitado, pero se recomienda que en el futuro se busque tener una población más grande, para agrupar más modelos mentales y ampliar el marco de información.

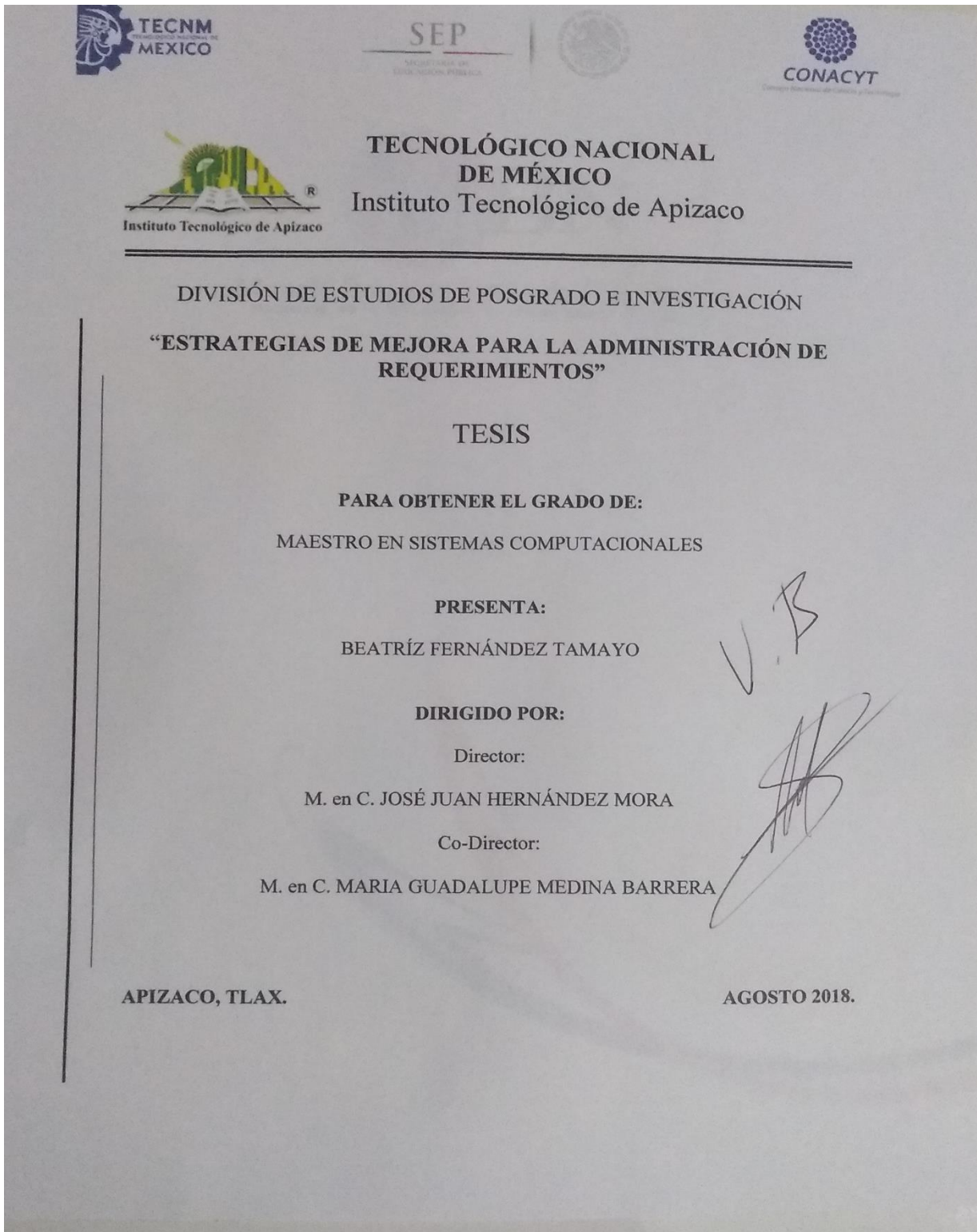
Referencias

- Alma E. Martínez Licon. (2008). *Calidad en el levantamiento de requerimientos*. Depto. de Ing. Eléctrica. Área de Computación y Sistemas.
- Allsoft software engineering. (2011). *El modelo CMMI (for Development)*. Monterrey, N.L. México.
- Bauer F. (1973). *Modulación, avance en ingeniería de software*. Springer-Verlag.pp.128-182.
- Boehm B. y Basili V. (2011). *Software defect redaction Top 10 list*.
- Chrissis M. B., Konrad M., Shrum S. (2009). *CMMI Guía para la integración de procesos y la mejora de productos*. Madrid. Lead-Appraiser.
- Geogy M., Dharani D. (2016). *A Scrutiny of the Software Requirement Engineering process. India*. Procedia Technology ELSEVIER.
- Gracia Bandrés, M.A., Gracia Murugarren, J., Romero San Martín, D. (2015). *TecsMedia: Metodologías de diseño centradas en usuarios*. Departamento de innovación, investigación y Universidad.
- IEEE-TSE. (1991). *Transactions on Software Engineering*.
- Institute of Electrical and Electronics Engineers. (1993). *IEEE Practicas recomendadas para la especificación de requerimientos de software*. New York: Software Engineering Standards Committee of IEEE Computer Society.
- Ignacio Eguía Salinas. (2010). *Filosofía Lean aplicada a la Ingeniería del Software*. Escuela Técnica Superior de Ingenieros. Universidad de Sevilla.
- Instituto Nacional de Tecnologías de la Comunicación. (2018). *¿Por qué fracasan los proyectos de software?*
- Kriesi C., Blindheim J., Bjelland O., Steinert M. (2016). *Creating Dynamic Requirements Through Iteratively Prototyping Critical*. Procedia CIRP ELSEVIER
- Mattmanna I., Gramlich S., Kloberdanza H. (2015). *The Inscrutable Jungle of Quality Criteria - How to Formulate*. Alemania. ELSEVIER.
- Miguel Vega. (2010). *Casos de uso*. Universidad de Granada
- Montero Yuseff Hassan. (2016). *No solo usabilidad Modelos mentales: nosolousabilidad.com*
Recuperado de http://www.nosolousabilidad.com/manual/2_3.htm.
- Leandro Alegsa. (2016). *Definición de Requerimientos (desarrollo de software)*. Recuperado de <http://www.alegsa.com.ar/Dic/requerimientos.php>
- LNCS. (2008). *Guía Práctica De Gestión De Requisitos*. Instituto nacional de la comunicación.
- Organización Internacional de Normalización. *ISO 9001 Sistemas de gestión de la calidad- Requisitos*. (2015). Ginebra, Suiza. Secretaria Central de ISO.
- Oktaba H., Alquicira C., Su A., Martínez A., Quintanilla G., Ruvalcaba M., López F. Rivera M.E., Orozco M.J., Fernandez Y., Flores M.A. (2003). *Modelo de Procesos para la Industria de Software MoProSoft*.

- Peter Senge (2016) *Modelos Mentales*. Escuela Internacional de Coaching. Garnica
- Pressman R., Ph.D. (2010). *Ingeniería del software, un enfoque práctico* 7 edición. University of Connecticut.
- Project Management Institute. (2014). *Pulso de la profesión de PMI: gestión de requisitos: una competencia central para el éxito de proyectos y programas*.
- Safwat A. y Senousy M. (2015). *Addressing Challenges of Ultra Large-Scale System on Requirements Engineering*. ICCMIT.
- Sommerville I., (2011). *Ingeniería de Software* 9 edición. México. Pearson educación.
- Sommerville I., (2016). *Requerimientos de Ingeniería: un tutorial*. IEEE.
- Rodríguez, C. T. (2017). *Impacto de los requerimientos en la calidad de software*. Bogotá-Colombia. TIA.
- Toni Granollers i Saltiveri. (2004). *Modelo de proceso de la ingeniería de la usabilidad y de la accesibilidad*. Universitat de Leida.
- Yusef H., Francisco J., Martín F. y Ghzala I. (2009). *Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información*. Recuperado de <http://www.hipertext.net> núm. 2.
- Hassan-Montero, Y.; Ortega-Santamaría, S. (2009). *Informe APEI sobre Usabilidad*. Gijón: Asociación Profesional de Especialistas en Información, 2009, 73pp. ISBN: 978-84-692-3782-3.
- Hassan Montero, Yusef; Herrero Solana, Víctor (2007). *Eye-Tracking en Interacción Persona-Ordenador*. <http://www.nosolousabilidad.com/articulos/eye-tracking.htm>.
- Walter Sanchez. (2011). *La usabilidad en Ingeniería de Software: definición y características*. Ing-novación.

Anexos





MBN Miracle Business Network S.A de C.V.
Innovación en la tecnología, su mayor aliado en negocios

CARTA DE ACEPTACIÓN

Tlaxcala Tlax a 15 de mayo del 2017

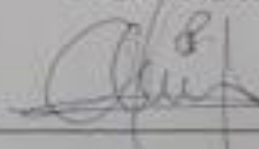
MTRO. FELIPE PASCUAL ROSARIO AGUIRRE
DIRECTOR DEL INSTITUTO TECNOLÓGICO DE APIZACO
PRESENTE

Por medio del presente le comunico que el C. Beatriz Fernández Tamayo alumno del Instituto Tecnológico de Apizaco de la Maestría en Sistemas Computacionales, ha sido aceptado para realizar sus Estancias profesionales bajo el proyecto "ESTRATEGIAS DE MEJORA EN LA ADMINISTRACIÓN DE REQUERIMIENTOS" en nuestra empresa MIRACLE BUSINESS NETWORK S.A. DE C.V. específicamente en el área de Desarrollo y Mantenimiento de Software en el periodo que comprende del 01 de febrero del 2017 al 17 de Agosto del 2017.

El C. Beatriz Fernández Tamayo estará bajo la responsabilidad de la Mtra. Carolina Roció Sánchez Pérez Responsable de Procesos de esta empresa, con el firme objetivo de poner en práctica sus conocimientos y desarrollar habilidades que engrandezcan su perfil profesional y faciliten su posicionamiento dentro del ámbito laboral.

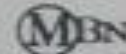
Es para nosotros un placer trabajar con alumnos de excelencia, con deseos de superación y actitud positiva ante los retos globales que hacen de cada esfuerzo un paso más hacia el éxito.

ATENTAMENTE



Lic. Ariana Ramírez Moreno

**Responsable de Recursos Humanos
y Ambiente de Trabajo**



Miracle Business Network, S.A de C.V.
Innovación en la tecnología, su mayor aliado en negocios

18 MAY 2017

R.F.C. MBN060316R09
Calle 3T No. 215 B La Loma Xicoténcatl
Tlaxcala, Tlax. C.P. 90070 Tel. 01 246 416 45 06

MBN Miracle Business Network S.A de C.V.
Hacemos de la tecnología, su mejor aliado de negocio

Tlaxcala, Tlax., 01 Marzo de 2018

Asunto: Constancia de Satisfacción.

Mtro. Felipe Pascual Rosario Aguirre
DIRECTOR DEL INSTITUTO TECNOLÓGICO DE APIZACO
Con atención a:
AT'N: Dr. José Federico Casco Vázquez
JEFE DEL DEPARTAMENTO DE DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN
P R E S E N T E:

Sirva la presente para enviarle un cordial saludo y notificarle que posterior a la recepción del proyecto de tesis de la **Ing. Beatriz Fernández Tamayo**, alumna de la Maestría en Sistemas Computacionales, con número de matrícula **M10370699**, de la institución que usted destacadamente dirige, se incluyó con el proyecto que lleva como título: **"ESTRATEGIAS DE MEJORA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS DE PROYECTOS DE SOFTWARE"**.

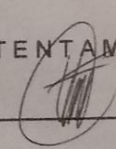
Siendo este desarrollado bajo la dirección del M. en C. José Juan Hernández Mora catedrático de la citada maestría.

En virtud de que sea cubierto satisfactoriamente los objetivos establecidos del citado proyecto.

Tenemos a bien dar constancia de que dicho proyecto de tesis cubre y satisface las expectativas planteadas al inicio de este proyecto.

Sin más por el momento, se extiende la presente carta de satisfacción para los efectos correspondientes.

ATENTAMENTE



Lic. Vidalia Flores Mojica

Responsable de Recursos Humanos y Ambiente de Trabajo.


Miracle Business Network, S.A de C.V.
Hacemos de la tecnología, su mejor aliado de negocio

01 MAR 2018

R.F.C.MBN060316R19

Calle 37 No. 216 B La Loma Xicohténcatl
Tlaxcala, Tlax. C.P. 90070 Tel. 01 246 416 45 00

Calle 37 # 216-B
La Loma Xicohténcatl, Tlaxcala, C.P. 90062

www.mbn.com.mx
Tel: 01 (246) 41 64508

Strategies of Improvement in Requirements Administration

Ing. Beatriz Fernández Tamayo¹, M. en C. Jose Juan Hernández Mora², M. en C. Carolina Roció Sánchez³,
M. en C. María Guadalupe Medina Barrera⁴

¹Master in computational systems, Instituto tecnologico of Apizaco,
Apizaco, Tlaxcala, México
fernandezbeatriz1908[at]gmail.com

²Department of systems and computing, Instituto tecnologico of Apizaco,
Apizaco, Tlaxcala, México
jjhmora[at]itapizaco.edu.mx

³Innovation and talent development center MBN-Miracle Business Network S.A.,
Tlaxcala, México
sanchez.carolina[at]mbn.com.mx

⁴Department of systems and computing, Instituto tecnologico of Apizaco,
Apizaco, Tlaxcala, México
medinabm0184[at]itapizaco.edu.mx

Abstract: *This article presents a model of developing and defining a series of strategies, for improvement in process of collection, analysis and specification of software project requirements, using agile develop methodologies like SCRUM reforced whit ideology of design UX and mind maps, exploiting artefacts designed standards low of quality CMMI and MoProsoft, this model support work team, for to prevent fails and changes what don't contribute evolution of software product.*

Keywords: Design UX, Improvement, Requirements, Software engineering

1. Introduction

IEEE define software engineering as application of a systematic, disciplined and quantifiable approach to software development, operation and maintenance (IEEE, 2008). The concept of software engineering it arose on 1986, after a conference in Garmischi (Germany) what have like objective solve the problems of software crisis.

The software engineering encompasses:

- ✓ Development software process (specification, implementation and design, etc.).
- ✓ Methodologies for software development (RUP, patterns, framework).
- ✓ Tools of development software.

In the software process we find:

Specifications: users and engineers define the software to produce and restrictions in function.

Development: phase of design and encode of software.

Validation: the software has must be tested for ensure quality.

Evolution: the software must be scalable for to accept changes future (TELLO, 2012).

One big part of the software crisis it's precisely the phase of requirements specification, in what found the problems like lack of know of environment of software work or sometimes exclude details about functions and features specific for user,

these failures result in costly losses of time, money and work, for the company, when we cope with bad practices of collection and validation of requirements.

In the development of software products its fundamental for little and big companies the evolution of requirements specification process of a new software products, using parameters more recent, for example include client in all phases of project, for achieve one system totally usable and intuitive.

This work searches the training of software factory team for good development of software products, focused in ideology UX (user experience), centralize all in the user perception.

2. Collection of requirements, whit focused in User experience

UX (user experience) it's a design philosophy whose objective is creation of products what resolve needs specifics of users, ensure higher satisfaction and best using possible, with minimum effort, it's a process in what using multidisciplinary techniques and all decisions it's based in needs, objectives, expectative and motivations of users.

All process of UX has the next structure:

- Thorough knowledge of users, using qualitative research or quantitative.
- Design a product that meets your needs and fits your capabilities, expectations and motivations.

Volume 7 Issue 5, May 2018

www.ijsr.net

Licensed Under Creative Commons Attribution CC BY

International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2017): 7.296

- Try the designed, with a usability test. (Clarenc, 2011)

Several tools of creativity they are useful to team of software factory for to land ideas, and after translate in business opportunities, one of this tools it's SCRUM what in easy terms it's set of practices good, for collaborative job of team with the end of get best results possible of the project, while in this context a prototype, one representation limited of a product, whose using resides in learning and exploration of functionality of initial idea, allowing evolution and correction of mistakes in early stages of project.

With base in analysis this models, tools and activities for process of requirements collection and specification that are established in several company's, we propose a new strategic methodology.

What contemplate reduction of mistakes in requirements knowledge, and that is adjusted considering the existing resources in the company, in order not to generate additional costs, this model can be learned by the current members of the software factory team. (Figure 1).

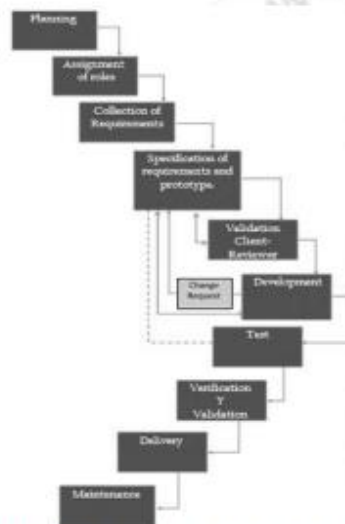


Figure 1 Requirements collection model with UX approach

Were:

Table 1: Symbology

Arrow	Relation
→	Sequence
- - - - -	Dependency
↔	Feedback

The first box indicates a review of development plan elaborated of administration of projects department that provides information such as policies, strategies, objectives and goals to the members of the development team, once reviewed and commented on this plan, the next point leads to Software Factory to determine what will be the role or roles

of each member, specify their tasks and deliverable documents.

As soon as this is specified, the requirements analyst, software architect, responsible for testing and the client will be cited for an interview session in which the tests will be applied and the client will be asked to make his own sketch of how he visualizes at that moment product that is required, sound recording of the meeting is suggested (with authorization of those present) for further analysis, the purpose of making backups of all this information is to understand not only the system as such but also the environment in which it will be used.

2.1 Assignment and description of roles

One of first activities of factory software area its the assignment of roles, for stablish responsibilities and chores.

After of revision and according to this chore, is important understand of each role and responsibilities inside team of SF for best organisation of development.

Then a diagram with the roles suggested by Moprosoft for SF is shown (Figure 2).

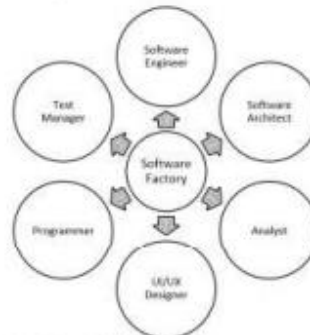


Figure 2 Roles

The adequate organization within the software factory for project development. it requires these five Roles understood and carried out correctly:

Software engineer knowledge's about principles and methodologies of software development, applying the scientific understanding to the logic design, construction and all documentation require for develop, operate and to give maintenance.

Software architect will possess knowledge about the organization and / or structure of the program, that is, diagrams, diagrams, language selection (if not specified by the client), total description to carry out the problem and in what form.

Analyst is one of the links between the user and the computer area, develops an analysis on the functionalities for the project, to detect and understand future problems, requires

Volume 7 Issue 5, May 2018

www.ijsr.net

Licensed Under Creative Commons Attribution CC BY

International Journal of Science and Research (IJSR)

ISSN (Online): 2319-7064

Index Copernicus Value (2016): 79.57 | Impact Factor (2017): 7.296

skills to detect opportunities for improvement that are associated with the main problem, collaborate effectively with the team through work sessions, verbal and written skills, business knowledge and technology domains, ability to absorb and understand information quickly. Idea and project visual messages, contemplating diverse needs that vary according to the case: stylistic, informative, identifying, persuasive, code, technological, production, innovation, etc.

UX / UI Designer this role could be executed by two different people since the UX designer will possess skills in the design of the total system interaction the proposal of navigation and system responses based on mental maps, perceptions and even emotions that can present the user, while a UI designer has knowledge about design, but referring to the aesthetics of the interfaces.

Programmer develops computer programs analyzing requirements, designing logical solutions, using the appropriate computational tools, in order to satisfy the client's requirements.

Test Manager Identifies tests to be carried out, and the appropriate way to apply them, implements individual tests, prepares and executes tests, records results, performs an analysis and recovery of execution errors.

Figure 3 shows a distribution indicating the participation of the different roles for each phase.



Figure 3 Roles Distribution

It's very important that the analyst learn to understand to each user and help him establish ideas.

2.3 Material

The challenge for analyst it's get useful information, in the shortest possible time, for start the development of the customer-focused system, when start information collection, needs more than knowledge in communication and software developer, will using didactics materials as test, pencil, paper, colors, etc. that help translate ideas from imagination a written narrative on paper, during meeting the requirements team, will inquire various aspects of the work environment and user characteristics, for start design not only the logic, also interface of system, this will give a broader perspective about our problem to be solved or detect opportunities for improvement.

To perform this task, different techniques necessary to obtain a better knowledge of the client's requirements are described. Using the following tools:

- ✓ initial test
- ✓ white screen test
- ✓ Sketch
- ✓ Prototype

What gives us as a result an extensive knowledge base for the analysis and definition of the base requirements of our project.

2.3.1 Initial test

As mentioned previously it's very important have information about the users, its say, question what your job is? or What are the processes you perform in your work? for which we using the next preliminary test:

- ✓ What is the sector of the company to which this project is directed?
- ✓ What is the objective this project?
- ✓ What is the approximate number of users?
- ✓ Are you familiar with any programming language or environment?
- ✓ What is the operating system most used in the company?
- ✓ What is the technical requirement (standards, and procedures to be used and applied) of this project?
- ✓ Requirements special of users for project?
- ✓ Do you have an example that serves as a reference?
- ✓ Who will be our direct contact?
- ✓ What will be its availability?
- ✓ We would have access to company information?
- ✓ Describe in your words the project (Some sheets are provided, maybe pencils or elements to make strokes).

One narrative or sketch made by our interviewee that will be reviewed during a subsequent meeting, the test should previously send for reduce the time invested. this can have complemented whit an ideas storm between team, one of the most common ways to get concepts informally is very useful in many situations where creativity and cognitive thinking are required.

This list of questions is based on the analysis of the problems obtained from past experiences, in internal and external projects of a company.

2.3.2 White screen test

This test will implement together with the initial test, for establishing idea of appropriate mind map for choose the artefacts and position inside the interface.

The objective of this test is show a vision close to the developer of how it could be interaction of user with the interface.

Volume 7 Issue 5, May 2018

www.ijsr.net

Licensed Under Creative Commons Attribution CC BY

Paper ID: ART20182905

DOI: 10.21275/ART20182905

1735

International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2017): 7.296

Start

The illustrations that show in the test are examples; the result of prototype could to vary.

Please choose the dispositive or (s) for to work.



PC TABLET SMARTPHONE

Each element has an index 1,2,3,4 etc. and options of selection as a), b), write the number the element and letter that chose, in a position in the screen.

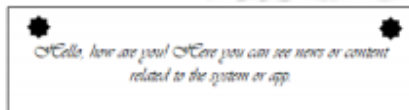


SCREEN

1. Company's logo, choose size and write position in the screen of up.



2. Write number 2 in the position that choose in the screen.



3. Write number 3 and letter of your election in the position that choose in the screen.

a) Horizontal b) Vertical



4. Write number 4 and letter of your election in the position that choose in the screen.

a) Icons b) Description



5. What colors do you prefer?



Write here your answer: _____

Thanks for your help!

2.3.3 Develop of deliverables documents and specification of requirements

In factory or development of software, makes several documents for collection information, the one it's "Collection of requirements" where to save a preliminary list of characteristics necessary of system, narrative, sketch, business rules and notes, in this portfolio to enclose the archives of audio and any other element of utility, whit meticulous analysis of all information, make the official requirements list, BPMN diagram, case of use diagrams and prototype that find in the document "Requirement Specification" that will verity and valid, the name of archives will are complemented whit name of project and portfolio will are called " Requirements".

3. Results

In test time of model "requirements based in user experience", we find several strengths and weaknesses they can improve in each project depending on the condition of this.

The model was proved in the project develop "Incidences Administration System".

All software development project has modifications but this not to be a indicative of failure or bad management of requirements, these changes to be considered as evolutions of system and the important is reduce them or reduce of impact this these can have in develop of project.

Volume 7 Issue 5, May 2018

www.ijsr.net

Licensed Under Creative Commons Attribution CC BY

In this project, specifically the requirements are very specific, in the requirements collection it was easy for team analysis of requirement understands very well the functions you need, whit help of a prototype realised whit Balsamiq Mock-ups software.

The documents necessary for CMMI and MoProsoft verification were realised more speed whit help SpiraTest platform.

The results of tests of Requirements collection model with UX approach, they threw the following data:
 In table 1 they have increment of time invested in relationship whit others projects.

Table 2: Estimated time vs. Real time inverted in hours

Estimated Time	Real Time Inverted
1	0.3
48	48
72	72
72	72
96	120
1	2
48	48

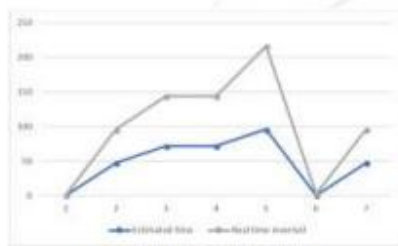


Figure 4 Time inverted in hours

Table 3 show corrections of project "Incidences Administration System". vs past projects, a reduction of corrections was obtained during the development of the project.

Table 3: Number corrections of project Administration requirements system vs past projects.

Activities	Others Projects	Corrections
Requirements collection	3	2
Analysis	4	2
Development prototype	4	3
Specification of requirements	3	1
Validation	3	1
Verification	1	0

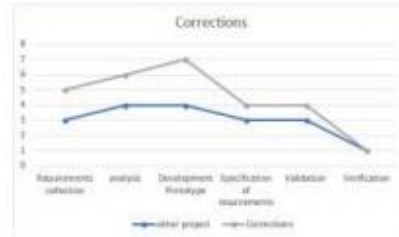


Figure 5 Time invested

4. Conclusion

Recognize interaction requirements as equal important that functional requirements and take in consideration, of minds maps and user experience provide more usability all systems of software and logically more quality.

During the meetings it was possible find the changes necessary and was easier for user and team understand what the desired result was in each module.

A low-level design prototype allowed user navigation, which generated more confidence in the project, it is necessary to specify that the visual design may vary if it is of the user's taste but there will be a time limited for said changes, but the iteration will be as faithful to the prototype as possible.

The tendency of all new software development models is to include the client in each step that the team takes, for this reason an approval is required from the requirements to continue to the development phase, without a written approval there is no progress to another phase.

References

- [1] Clarenc, C. A. (2011). Notions of cyberculture and journalism.
- [2] IEEE. (2008). Standart software engineering. IEEE 830-1993. IEEE.
- [3] TELLO, D. E. (2012). Basic concepts of Software engineering. Technology and Innovation.

Author Profile



Ing. Beatriz Fernández Tamayo obtained the degree of Engineer in Information Technology and Communications from the Technological Institute of Apizaco. Student of the Masters in Computational Systems of the Technological Institute of Apizaco. Areas of interest in research: Software Engineering, Application Software Development Methodologies.



M. in C. Jose Juan Hernández Mora is a professor of the Master's Degree in Computational Systems of the Technological Institute of Apizaco. Area of Affiliation: Systems and Computing, is a Computer Engineer from the Autonomous University of Tlaxcala and Master in Computer Science from the National Center for Technological Research

International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2017): 7.296

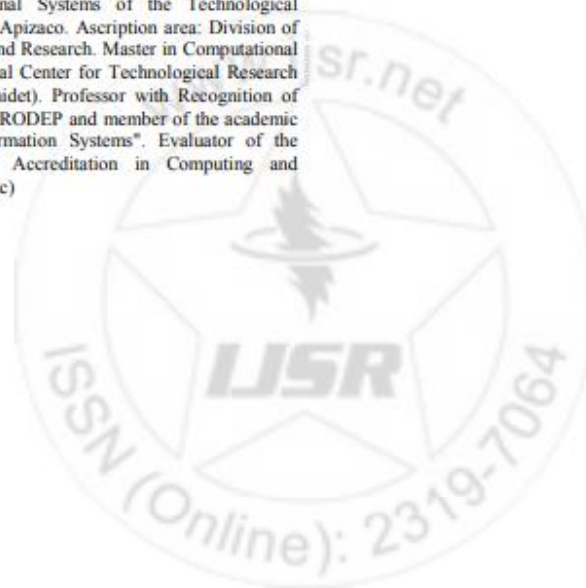
and Development. Areas of interest in research: Software Engineering, Application Software Development Methodologies, Digital Image Processing, Neural Networks.



M. in C. Carolina Rocío Sánchez Pérez is a full-time associate professor of Computer Engineering at the Universidad Autonomy de Tlaxcala and academic coordinator of the Computer Engineering Education Program. Master in Computer Science from the National Institute of Astrophysics, Optics and Electronics (INAOE). He has the Recognition of Desirable Profile PRODEP and is a member of the Academic Body "Distributed Systems" of the UATx. He has participated in bonding projects funded by CONACYT.



M. in C. María Guadalupe Medina Barrera is a full-time professor of the Master's Degree in Computational Systems of the Technological Institute of Apizaco. Ascription area: Division of Postgraduate Studies and Research. Master in Computational Systems by the National Center for Technological Research and Development (cenidet). Professor with Recognition of the Desirable Profile PRODEP and member of the academic body PRODEP "Information Systems". Evaluator of the National Council of Accreditation in Computing and Computing A.C. (conaic)



Volume 7 Issue 5, May 2018

www.ijsr.net

[Licensed Under Creative Commons Attribution CC BY](https://creativecommons.org/licenses/by/4.0/)

Paper ID: ART20182905

DOI: 10.21275/ART20182905

1738