

Urkund Analysis Result

Analysed Document: extracto_201852113621.docx (D38237821)
Submitted: 5/3/2018 2:10:00 AM
Submitted By: mcorreap@unemi.edu.ec
Significance: 1 %

Sources included in the report:

<http://upcommons.upc.edu/handle/2099/15461>

Instances where selected sources appear:

1

ÍNDICE GENERAL

DERECHOS DE AUTOR II DERECHOS DE AUTOR III APROBACIÓN DEL TUTOR DE LA INVESTIGACIÓN DOCUMENTAL IV APROBACIÓN DEL TRIBUNAL CALIFICADOR V DEDICATORIA VI AGRADECIMIENTO VII ÍNDICE GENERAL VIII RESUMEN 1 ABSTRACT 2 INTRODUCCIÓN 2 CAPÍTULO I 4 PROBLEMA DE INVESTIGACIÓN 4 Objetivo general 5 Objetivo específicos de la propuesta 5 Justificación 5 CAPÍTULO II 6 MARCO TEÓRICO CONCEPTUAL 6 CAPÍTULO III 11 METODOLOGÍA 11 METODOLOGÍA DESCRIPTIVA 11 METODOLOGÍA DOCUMENTAL 11 CAPÍTULO IV 13 DESARROLLO DEL TEMA 13 CAPÍTULO V 15 CONCLUSIONES 15 REFERENCIAS BIBLIOGRÁFICAS 16

ANÁLISIS DE METODOLOGÍAS ÁGILES ENFOCADO AL MODELO DE REQUERIMIENTO EN EL DESARROLLO DE SOFTWARE

RESUMEN

La presente investigación documental se lleva a cabo debido a la deficiencia de efectividad de desarrollo de software realizado por parte de los desarrolladores hacia los usuarios finales debido a la falta de comunicación y de captación de los requerimientos por parte del desarrollador y de parte del usuario final la falta de argumentación y de especificación del sistema que solicita, la metodología aplicada en la investigación es la descriptiva de manera que se quiere realizar un análisis documental de fuentes confiables (artículos científicos), acerca de las metodologías ágiles aplicadas al modelo de requerimiento para el desarrollo de software, la comunicación, planificación, asignación de responsabilidades y el financiamiento se plantea como factores de gran importancia y consideración al momento de diseñar un determinado software, se ha planteado analizar herramientas tecnológicas de modelado de requerimientos de Open Sources como lo es SIAR (Sistema integral de administración de requerimientos), gestiona los requerimientos funciones de un sistema, o uno de pago como lo es IBM Rational DOORS Next Generation herramienta de administración de requisitos basada en plataforma de colaboración IBM Jazz ofreciendo acceso a las partes del ciclo de vida de un software, esto en base las metodologías ágiles para realización de software de una forma satisfactoria para el usuario final y eficiente por parte del desarrollador.

PALABRAS CLAVE: Comunicación, Distribución de responsabilidades, Planificación, Financiamiento.

ANALYSIS OF AGILE METHODOLOGIES FOCUSED ON THE MODEL OF REQUIREMENT IN THE DEVELOPMENT OF SOFTWARE

ABSTRACT

The present documentary investigation is carried out due to the deficiency of efficiency of development of software realized on the part of the developers towards the final users due to the lacking communication and of capture of the requirements on the part of the developer and on behalf of the final user the lack of argumentation and of specification of the system that it requests, the methodology applied in the investigation is the descriptive one so that there wants to be realized a documentary analysis of reliable sources (scientific articles),

brings over of the agile methodologies applied to the model of requirement for the development of software, the communication, planning, assignment of responsibilities and the financing appears as factors of great importance and consideration to the moment to design a certain software, It has considered to analyze technological tools of shaped of requirements of open sources since it it is A SIAR (integral System of administration of requirements), manages the requirements functions of a system, or one of payment like it is IBM Rational DOORS Next Generation tool of administration of requirements based on platform of collaboration IBM Jazz offering access to the parts of the life cycle of a software, this in bases the agile metodólogas for accomplishment of software of a satisfactory form for the final and efficient user on the part of the developer. KEY WORDS: Communication, Distribution of responsibilities, Planning, Financing.

INTRODUCCIÓN

La ingeniería de software es un conjunto de herramientas que permite realizar procesos de desarrollo, mediante requerimientos, análisis, pruebas, validaciones y modelos para la creación eficiente de un sistema, no es algo rígido que se lleva a cabo, más bien en su perspectiva de desarrollo, permite la adaptabilidad en el sistema, gracias a esto los colaboradores del proyecto pueden realizar las acciones necesarias para ofrecer un software de calidad en el tiempo determinado, capaz de cumplir las expectativas del cliente, satisfaciendo sus necesidades, es decir que el sistema sea intuitivo y permita aprender su utilización en poco tiempo. Cuando se desea desarrollar un software se comienza levantando la información necesaria que permitirá conocer los requerimientos funcionales y no funcionales del sistema, pueden presentarse problemas cuando no se manejan correctamente sus necesidades, generando dificultades al momento de plasmar la información en el desarrollo del software; con las metodologías ágiles se crean sistemas que previenen los cambios de contexto aunque no sucedan y aun cuando no se tenga claro las necesidades y existan cambios no afectara a todo el proyecto sino solo a la parte involucrada. Los casos de uso permiten tener una descripción de la secuencia de interacciones en el sistema el cual mediante su sucesión ordenada de pasos permite obtener una visualización de que tarea se debe desarrollar para cumplir el flujo de eventos a llevar a cabo, siendo las precondiciones que se deben llevar a cabo una vez se comienza el caso de uso del proyecto. Utilizando SIAR permite registrar de manera formalizada los casos de uso permitiendo administrar los atributos del proyecto, medir los alcances, clasificar las prioridades, registrar consultas del sistema, cuando se crea el caso de uso se debe verificar y asegurar su validez, esto consiste en la secuencia de acciones que contengan el caso de uso, se pueden realizar automáticamente trabajando con autómatas finitos siendo un conjunto de estados que se mueven de un estado a otro en respuesta a las entradas del sistema.

La presente investigación documental su estructuración consiste en 5 capítulos: el capítulo 1 contiene el problema de investigación relacionada a la temática de metodología ágil desde el levantamiento de información como también los objetivos tanto general como los específicos y la justificación, el capítulo 2 se presenta el marco teórico conceptual que contiene el desarrollo teórico de la fundamentación de la investigación documental, el capítulo 3 contiene la metodología aplicada en el trabajo documental, el capítulo 4 se expresa el desarrollo del

tema de investigación propiamente dicho y el capítulo 5 contiene de manera concisa las conclusiones correspondiente.

CAPÍTULO I PROBLEMA DE INVESTIGACIÓN

Al desarrollar un sistema requiere un inicio como es el levantamiento de información, que consiste en la explicación del sistema que requiere el usuario, esto presenta un gran inconveniente por parte de los clientes o futuros usuarios hacia los desarrolladores, puesto que no tienen claras las necesidades, lo que implica problemas al momento de plasmar sus ideas en la obtención de un sistema, debido a ciertos factores como problemas de comunicación, escasa distribución de responsabilidades, alteraciones en la planificación, además agregar a esto inconvenientes en el financiamiento de un proyecto de software lo que genera pérdida de tiempo. Es así que para evitar cambios de contextos, en las últimas fases del ciclo de vida de un software, nacen las metodologías ágiles como complemento no perteneciente a las fases del desarrollo del software, la función a cumplir es que al momento de desarrollar software pues se crea la documentación precisa, reduciendo tiempo y apoyando a cada una de las etapas para la obtención del software.

En el caso de la comunicación, puede considerarse un punto crucial en todo proyecto y más aún en un sistema, la relación debe ser directa entre el usuario y desarrollador, cuando no es efectiva esta comunicación genera inestabilidad; sin embargo la finalidad es el trabajo en conjunto y la evaluación de los requerimientos.

Además, la distribución de responsabilidades así como la colaboración del cliente coadyuva a definir los roles o asignaciones de tareas, para generar nodos de responsabilidad en menor escala, con la finalidad de reducir los tiempos en el desarrollo de un software.

Para la planificación de un software, se debe también considerar aspectos como actividades, tiempo, responsables y recursos no solo para identificar el punto inicial sino para comparar lo planificado versus lo ejecutado, e inclusive tomar acciones a tiempo para obtener un producto establecido en un sistema.

Otro factor que puede afectarse es el financiamiento que está directamente relacionado con una adecuada planificación, este financiamiento deberá considerarse en cada una de las etapas del software: planificación, desarrollo, evaluación, medición y control del sistema.

Objetivo general Analizar artículos de las bases de datos científicas y herramientas colaborativas acerca de las metodologías ágiles enfocadas al modelo de requerimiento para el desarrollo del software, aplicando el método descriptivo en el contexto para su respectiva observación y conclusión.

Objetivos específicos • Definir el problema de investigación en base a las metodologías ágiles enfocadas al modelo de requerimientos para el desarrollo de software. • Evaluar las fuentes documentales acerca de las metodologías ágiles. • Concluir los resultados de la evaluación de fuentes bibliográficas de las metodologías ágiles y su relación con los requerimientos para el desarrollo del software.

Justificación

Este trabajo es importante pues se obtendrá un análisis documental para comparar resultados de otros estudios relacionados a la temática propuesta, donde la comunicación, planificación, asignación de responsabilidades y financiamiento aportan a la creación de un sistema el cual es requerido por un determinado usuario. Cabe reconocer que existen sistemas con malas estructuraciones y diseños creando insatisfacción por parte del usuario que requiere el software y también vulnerabilidad hablando propiamente de las seguridades, esto debido a la falta aplicación de las metodologías ágiles como requisito indispensable para el desarrollo del software. Todo esto se da a causa de falta de conocimiento de las nuevas metodologías, inexperiencia en la aplicación de las mismas y por desinterés del desarrollador para fomentar y aplicar las metodologías como proceso en el desarrollo del software. Otro factor causal es la forma lexical en que se expresa el usuario al requerir el diseño de su sistema y la poca aportación en responsabilidad que brinda hacia el desarrollador. Esta investigación documental se realiza con la finalidad de destacar la gran importancia existencial que tiene la obtención en base a conocimiento y aplicación de las metodologías ágiles por parte del desarrollador en los sistemas, la participación directa del usuario en cada uno de los procesos que conlleva a la elaboración correcta de los sistemas para con eso cambiar esa perspectiva ambigua de desarrollar software.

CAPÍTULO II MARCO TEÓRICO CONCEPTUAL

Desarrollado por Nielsen y Norman nace el concepto de usabilidad ágil coexistiendo en la última generación, este es un nuevo paradigma en la producción de elaboración de software, para entender bien sus funciones se debe conocer que es el término ágil, cuando existe un desarrollo que sustituye al modelo tradicional a cascada estamos hablando de que es ágil, subdividiéndose en ciclos de desarrollo de software. (Arcadio, 2011) Las fases de análisis, diseño, implementación, prueba y liberación se desarrollan en pequeños equipos funcionales que se repiten en las subfases continuas, cada fase se denomina terminología ágil la mayoría de los casos es incremental, en Scrum es conocida como sprint, si este no es completo se lo libera y se trata con los siguientes sprint, con el motivo de acercarse cada vez más a las necesidades del cliente, cuando finaliza un sprint los desarrolladores del equipo reevalúan las prioridades del proyecto para determinar que modificación realizar en el siguiente sprint hasta finalizar el trabajo. (Arcadio, 2011) Un método ágil ayuda en la comunicación cara a cara es decir en tiempo real y es más preferido que la documentación del proyecto, debido a que se entiende de manera rápida cómo lo expresa el encargado del desarrollo y si existiera una duda se la disolvería en poco tiempo, un equipo de desarrollo ágil está integrado por cada persona necesaria para terminarlo, se debe incluir por lo menos los clientes y diseñadores para que el objetivo se lleve a cabo en plena satisfacción para el cliente y no solo se deba cumplir el proyecto por el contrato. (Arcadio, 2011) Interactuar con las personas es más importante que los instrumentos y procesos del software es decir que relacionarse con los involucrados en el desarrollo del proyecto es el mejor recurso para su completo entendimiento y creación. Conservar un código simple y avanzado en términos tecnológicos es más importante que tener una gran documentación, es decir que es mejor tener un software operativo de fácil contexto con una pequeña documentación que detalle lo

relevante, porque si nos centramos en tener un documento con demasiados términos y dejamos a un lado el código podríamos realizarlo con difícil entendimiento, siendo estos unos de los principios de las metodologías ágiles de software. (Arcadio, 2011) Se debe destacar que las metodologías ágiles son todas aquellas que permiten revolucionar los antiguos paradigmas de Ingeniería de software mediante la modificación de un grupo de especificaciones y secuencias que suceden durante el desarrollo del sistema, esto se produce intercambiando grandes cantidades de información entre el diseñador y el cliente. (Arcadio, 2011) Las metodologías ágiles se direccionan a superar la usabilidad tradicional de los antiguos paradigmas de software aunque instala nuevas amenazas en el usuario, las sociedades que lo implementan obtienen beneficios sin tanto riesgos, Nielsen detalla que el mayor riesgo que existe en estas metodologías es cuando un método del programa responde mejor a la implementación generando el descuido en el diseño del usuario y la usabilidad. (Arcadio, 2011) Cuando se utiliza el método ágil se debe centrar en el diseño del usuario y la usabilidad de las metodologías de desarrollo explícito del contexto incluyéndose en el proceso de creación del software, es decir que si se toma en consideración el usuario y su uso, damos los mismos recursos que implementemos en la codificación, permitiendo evaluar la usabilidad del software durante las fases de desarrollo, estos test permiten ir a la par con lo que se necesita el cliente reduciendo el riesgo de que se libere al mercado un sistema que no sea utilizado por su complicado funcionamiento. (Patel et al., 2012) Un proyecto que emplea las metodologías ágiles se subdivide en pequeños procesos el cual con constante comunicación con el cliente nos ayuda a adaptarnos mejor al cambio en los requerimientos si fuera necesario, permite la retroalimentación por parte del usuario, ya que este método es flexible y mediante la mejora continua del software por los cambios en los requerimientos constantes proporciona un producto de calidad y adaptabilidad. (Silvia Gabriela, 2012)

Las metodologías ágiles nacen en los años 90 conocidas como desarrollo de software ligero tiempo después nombrada como se las conoce actualmente, estas intentaban reducir los fracasos del software, disminuir los costos, tiempo y funcionalidades en el proyecto, nacen como reacción de disminuir la burocracia que se debe seguir en las metodologías tradicionales, estas imponen un orden y disciplina en el desarrollo de un sistema volviéndose predice en las actividades y eficiente en los resultados. (Cadavid, Martínez, Daniel, & Vélez, 2013) El desarrollo de un software con el metodología ágil permite tener resultados exitosos en trabajo colaborativo, aunque se emplee un gran número de personas en el mismo, debido a que este modelo proporciona comunicación efectiva, adaptabilidad, participación y auto inspecciones en el proyecto, permitiendo minimizar los errores por cambio de requerimientos del cliente a último momento.(Cadavid et al., 2013) En las metodologías ágiles se espera que los requerimientos cambien durante el proceso, es decir que este método nos proporciona herramientas que permitan incrementar los nuevos cambios sin afectar el desarrollo iniciado del software, se caracterizan por la simplicidad, entrega frecuentes, priorizar los requerimientos y comunicación efectiva con los colaboradores del proyecto con el cliente. . (Rivadeneira, Vilanova, Miranda, & Cruz, 2013) Actualmente encontramos sistemas complejos con cambios imprevistos en el contexto, con modificaciones en los requerimientos por parte del cliente, aunque el proyecto de software este empezado, generando resultados difíciles de evaluar por su nueva planificación y desarrollo, debido a que el proyecto tiene una estructura de desarrollo inicial, reestructurar e implementar los nuevos requerimientos es una tarea

difícil porque volver a comenzar de cero el proyecto necesitara más tiempo para que cumpla las objetivos y expectativas requeridas. (Diego Alberto Godoy; Edgardo Belloni; Henry Kotynski; Hector Dos Santos, 2014) En estos casos es necesario implementar las metodologías de desarrollo de software permitiendo crear resultados rápidamente, Scrum es una metodología que centra su atención en las actividades de gerencia basadas en la planificación administrativa con desarrollo incremental, este modelo fue aplicado por Ken Schwaber y Jeff Sutherland para poder realizar software en poco tiempo de manera funcional a los requerimientos. (Diego Alberto Godoy; Edgardo Belloni; Henry Kotynski; Hector Dos Santos, 2014) Las herramientas Dynamics of Agile Software, Modeling Agile Software Maintenance Process, entre otros, son instrumentos de metodologías ágiles que permiten presentar modelos formulados, analizados y evaluados en diferentes escenarios, que permiten tener un conociendo rápido, eficiente y eficaz de que implementar o cambio realizar en el software a desarrollar. (Diego Alberto Godoy; Edgardo Belloni; Henry Kotynski; Hector Dos Santos, 2014) Al utilizar Scrum evitamos la acumulación de documentación de los requerimientos del cliente y solo nos enfocamos en los necesariamente importantes para el proyecto, esto permite que sea accesible en cualquier momento la información siendo esta necesaria en cualquier fase de desarrollo para su consulta o actualización. (Goñi, Ibáñez, Iturrioz, & Vadillo, 2014) La estructura backlogs permite gestionar información instantáneamente comprobando la autenticación del proyecto, cuando se genera un backlogs de un software se crea una lista de requerimientos de casos de uso conocidos como features (características) o user histories (historias de usuario), en cada interacción de los requisitos se incrementan construyendo un backlog sprint conteniendo actividades que se efectuaran en este proceso. (Diego Alberto Godoy; Edgardo Belloni; Henry Kotynski; Hector Dos Santos, 2014) Un backlogs es una herramienta que permite al equipo de desarrollo observar el avance del proyecto mediante el desenvolvimiento individual de los integrantes en el desarrollo del sistema. Los backlogs permiten al equipo identificar los problemas y contribuir con el progreso del software conforme se vaya avanzando el proyecto sin exigir demasiado tiempo en procesos complejos. (Goñi et al., 2014) Cuando realizamos un mantenimiento de baklogs de sprint nos ayuda a utilizar la comunicación directa para poder establecer las actividades a realizar cada uno, repartiéndose las responsabilidades del proyecto, debido a que los integrantes del equipo no toleraran si en un avance se entregue incompleto, si se presentara desventajas o incidentes en el trabajo colaborativo se tomara la responsabilidad grupal para poder resolverlo y enfrentarlo para tomar correctas decisiones que permitan mejorar la dinámica del equipo de software, mediante reuniones de retroalimentación se analiza que problema se tiene para poder mejorarlo y saber que realizar para mejorarlo. (Goñi et al., 2014) Extreme Programming es una metodología responsable en las pruebas de software por lo cual se ha incrementado su uso, con XP (Extreme Programming) los requerimientos de los clientes se representan mediante artefactos que se denominan "Historias de usuario" que permiten representar particularmente las necesidades del usuario, es decir que ya no se registra la información en documentos estandarizados sino que se utilizan herramientas que facilitan su uso mediante las historias de usuario. (dos Santos & Canedo, 2014) Las historias de usuario contienen una lista de cada uno de los criterios de aprobación, el cual permite que se definan las tareas que se realizaran por fases del proyecto, antes que se comience con la implementación de los requerimientos del sistema, obteniendo una validación de las pruebas para obtener las

necesidades de los usuarios para agregarlos en los requerimientos a necesitar en el proyecto. (dos Santos & Canedo, 2014) Entre los principios ágiles debemos destacar que se debe estar preparado para responder a cualquier cambio que se realice en el proyecto, el equipo debe estar autorizado para sugerir modificación que ayuden al desarrollo del software, también se debe favorecer al cliente más haya de solo cumplir el contrato sino ofrecer buenos resultados. (Leiva Mundaca & Villalobos Abarca, 2015) En un contexto con requerimientos cambiantes tener una herramienta como Scrum que permita tener una simulación de gestión del desarrollo del proyecto de software que permita evaluar el impacto de las decisiones a los administradores sobre el desarrollo del proyecto sin poner en riesgo el real y gastar los recursos del mismo. (Barba-Guaman, Quezada-Sarmiento, Calderon-Cordova, & Lopez, 2017)

CAPÍTULO III METODOLOGÍA

METODOLOGÍA DESCRIPTIVA La metodología descriptiva reitera el propósito de describir sucesos o eventualidades dentro de la problemática de investigación cabe destacar que esta metodología es aplicada por el investigador. El cómo se manifiesta y define un determinado fenómeno a estudiar. Este tipo de metodología delimita cualquier tipo de propiedades ya sea de una persona en particular, un grupo, comunidad o cualquier otro tipo de fenómeno que se pueda analizar. Desde la perspectiva científica la palabra describir tiene como significado “medir”, siendo una de sus características principales dentro del contexto metodológico. Miden y evalúan aspectos diversos, dimensiones, fenómenos o componentes de uno sometido a investigación. Los estudios descriptivos se caracterizan por realizar la medición de las variables o conceptos de forma independiente debido a que el objetivo de este tipo de estudio no es indicar las relaciones existentes entre las variables, sino de describir los fenómenos con la mayor precisión posible. Reiterando que no esta metodología no relaciona las variables para poder ser medidas pero lo que sí es posible, es integrar las mediciones de las variables para llegar a determinar de manera general como se manifiesta y como es el fenómeno. Esta metodología es seleccionada en esta trabajo documental debido a las características propias del mismo, ya que el propósito de esta investigación consiste en la recolección de información (teorías, hipótesis, entre otros), con la finalidad de realizar un analizar exhaustivo referente al tema de investigación.

METODOLOGÍA DOCUMENTAL Dentro de los tipos de investigación documental tenemos el diseño de campo (empírica) y el diseño bibliográfico (temática), ambas tienen como fin la resolución de problemas existentes. Pero Su diferencia radica en la forma en que se obtiene los datos para ser analizados. El diseño de campo se realiza mediante la recolección de datos primarios, es decir, debe aplicarse en la investigación alguna de las técnicas existentes de recolección de datos para proceder a su respectivo análisis y concluir en resultado. En cambio el diseño bibliográfico parte de la obtención de datos secundarios en pocas palabras otras investigaciones ya existentes en relación a la temática o problema a resolver. El tipo de metodología documental aplicado en esta investigación es el diseño bibliográfico de modo que este enfoque de investigación está basado en un 100% en artículos científicos proveniente de fuentes (bases de datos) confiables.

Análisis de metodologías ágiles enfocado al modelo de requerimiento en el desarrollo de software

Gráfico 1. Esquema de aplicación de la metodología descriptiva en la investigación documental. CAPÍTULO IV DESARROLLO DEL TEMA

Este trabajo de investigación es una pauta para la descripción de herramientas que brindan una ayuda para la creación y mejora de software en base a los requerimientos. Uno de ellos es un software aplicativo en la plataforma web denominado por sus siglas SIAR que significa (Sistema Integral de Administración de Requerimientos), llamado así por la función principal que el mismo realiza. Se encarga de gestionar cada uno de los requerimientos funcionales de un determinado sistema en particular. El software está basado en los lineamientos del UML denominado así y que significa (Lenguaje de modelado unificado). (Medina et al., 2016) El aplicativo web surge dentro de un proyecto de investigación realizado por el grupo en sus siglas GIDTSI (Grupo de Investigación, Desarrollo y Trasferencia de Sistemas de Información), en el cual el respectivo proyecto fue llamado "Validación de requerimientos a través de modelos conceptuales. Este grupo forma parte y es dependiente del Departamento de Ingeniería de Sistemas de información de la Universidad Tecnológica Nacional en la Facultad Regional Córdoba. (Medina et al., 2016) La facultad se ha planteado brindar una solución a uno de los problemas radicales que tiene la ingeniería de requisitos en base a la elicitación y especificación de los requerimientos en el cual intervienen de forma directa varias etapas del proceso del desarrollo del software, sobrellevando la trazabilidad desde la validación de los requerimientos hasta la implementación del software. (Medina et al., 2016) La realización de un software encargado de la gestión integral de caso de uso, SIAR es un aplicativo web que trabaja con múltiples sistemas, cada sistema posee un alcance distinto y versiones diversas. Entre los alcances cada uno gestiona un determinado grupo de caso de uso, así mismo cada caso de uso posee una secuenciación de pasos de forma ordenada. Los pasos en los casos de uso están conformados por tareas que son descriptivos en base al soporte del sistema que se ha desarrollado. Desde el punto de vista conceptual, un caso de uso es la representación de una secuenciación ordenada entre uno o varios actores y el sistema. (Medina et al., 2016) SIAR, tiene como objetivo encargarse de gestionar los requerimientos, este utiliza un esquema distinto que no únicamente administre los casos de uso por separado sino que además gestione de forma integral el proyecto en sí de desarrollo. Aquí se presentan los otros puntos que tiene SIAR como alcance, configurar el entorno de desarrollo, la interface de usuario, la administración de proyectos de sistemas, administración de casos de uso, sus versiones, las consultas y exportaciones a formato de archivo XML. Siendo estas las principales funcionalidades que el software SIAR proporciona para la iteración e intercambio de datos con otras aplicaciones por ejemplo un autómata infinito. (Medina et al., 2016) La validación de consistencia de casos de uso con simuladores de autómatas finitos, tiene como componentes principales de esta metodología aplicada con los casos de uso ya que son estos los que especifican el comportamiento del sistema. En su definición un caso de uso describe y especifica la secuenciación de las acciones, incluyendo variantes que el sistema puede ejecutar y que produce una resultante de tipo observable de valor para un determinado actor. (Medina et al., 2016) Este software transforma el caso de uso en un autómata finito determinista para su respectiva validación de su cohesión secuencial, siendo así el presente

del análisis de la validación de requerimiento que se propone SIAR como proyecto. Su funcionalidad que se plantea como una alternativa viable ante necesidades como registración normalizada de requerimiento, transformación del caso de uso en máquina de estados y validación de la consistencia secuencial de los cursos de acciones del caso de uso. (Medina et al., 2016) Existen muchas otras herramientas administrativas de requerimientos en el mercado las cuales en su gran mayoría no son open source, se debe adquirir una licencia para poder utilizarlas y muy pocas son quienes brindan un demo para poder conocer la herramienta y familiarizarse. Una de esas herramientas es CodeBeamer Requirements Management, no es open source pero es muy buena. Su función es la de administrar los requisitos ayuda a todo el equipo encargado del desarrollo del software a poder especificar, administrar y rastrear la cobertura de los requisitos del software en todo el proceso de desarrollo, garantizando así la trazabilidad, colaboración y cobertura tanto de código como de pruebas desde el inicio de los requisitos hasta su publicación.(Trail, 2017) CodeBeamer ALM, es una herramienta de netamente de gestión de requisitos todo en uno, a pesar de eso permite y puede adaptarse al entorno de desarrollo existente en el cual se está trabajando. Su API REST, es flexible para pudiendo así integrarse de manera personalizada como MS Office, SAP, JIRA, Enterprise Architect, MATLAB Simulink e IBM Rational DOORS entre otros, a través de ReqIF.(Trail, 2017) Gestiona el ciclo de vida de aplicaciones siendo una solución que ayuda a administrar la complejidad del software y hardware con capacidades avanzadas que pueden tomar gestión de varios ciclos de vida y apoyar y ser soporte de todo el desarrollo de software. Aparte de la robustez, realiza los productos con eficiencia y de forma eficaz esto gracias al código proyector ALM, manteniendo la fiabilidad y la calidad del software de la demanda y los requisitos de gestión mediante el desarrollo de todo el camino a través del código de liberación y sus operaciones de pruebas. Ofrece trazabilidad sin pausas mejorando la transparencia, la colaboración entre los miembros del equipo y características para aumentar la eficiencia de los esfuerzos de desarrollo de la ALM.(Trail, 2017) Permite la captura de airado, guardar y reutilizar sus necesidades en su proyecto y puede derivar sus tareas y pruebas, casos directamente de los requisitos, analizar la cobertura de prueba en los enlaces en tiempo real. La funcionalidad de gestión de riesgos que ayudan que precisamente reducen y mitigan los riesgos. Los métodos de desarrollo ágiles e híbridos son apoyados, mediante el código Reírme que ofrece un aplantilla preconfigurada de seguridad para el desarrollo ágil escalado, aunque es preconfigurado posee flexibilidad personalizable, plantillas con procesos ejecutables son disponibles mediante esta herramienta para la industria automotriz y medica entre otras.(Trail, 2017)

Gráfico 2. Ventana principal de CodeBeamer.

Gráfico 3. Gestión de requisitos en la herramienta CodeBeamer.

Otra herramienta robusta en el tratamiento de los requerimientos del desarrollo del software es IBM Rational DOORS Next generation. Basado en los conceptos de IBM Rational DOORS, es una herramienta de administración de requisitos basada en la plataforma de colaboración IBM Jazz. Ofrece acceso directo a los requisitos para las partes interesadas en todo el ciclo de vida de la ingeniería.(IBM, 2013) Se puede implementar en las instalaciones o en la nube en un entorno de software como servicio (SaaS). Una solución totalmente basada en la web que

incluye definición de requisitos visuales, planificación, gestión de elementos de trabajo e integración de pruebas. Basado en una base de datos comercial con administración de usuarios compartidos, capacidades con licencia y convenciones de interfaz de usuario optimizadas.(IBM, 2013) El mayor desarrollo de problema en un proyecto de software se encuentra en los requisitos seleccionado, al no establecer bien y no llevar control de aquello puede terminar en la mal práctica y resultado de desarrollo de software.(IBM, 2013) La solución de gestión de requisitos de IBM le ayuda a capturar, rastrear, analizar y gestionar el desarrollo de aplicaciones de TI avanzadas y sistemas, la imagen siguiente muestra una parte del funcionamiento de esta herramienta.(IBM, 2013)

Gráfico 5. Gestión de los requerimientos en el software. IBM Rational DOORS Next Generation proporciona una API de extensión de cliente que puede usar para ampliar la funcionalidad de la herramienta utilizando tecnologías que ya conoce, como HTML y JavaScript. Puede crear y alojar un catálogo de extensiones en un servidor para que su equipo pueda compartirlas.(IBM, 2013)

La siguiente imagen muestra una extensión que puede verificar valores de atributos que están relacionados con el estándar ISO 26262 ASIL.(IBM, 2013) Gráfico 6. Extensión de verificación de valores de atributos.

CAPÍTULO V CONCLUSIONES

Hablar de metodologías ágiles es referirse al nuevo concepto de software que llega para poder facilitar el desarrollo de los sistemas de manera que si se presentan cambios imprevistos en los requerimientos en cualquier parte del ciclo de vida, permita al desarrollador modificarlo sin afectar de manera drástica el proyecto, cambiando esa parte sin tener que parar el desarrollo total, si el estado de avance es demasiado avanzado, empezarlo de nuevo causarían problemas, la metodología ágil facilita la elaboración del proyecto, debido que permiten desarrollar software de una manera más eficiente y la óptima posible. Las herramientas que hoy en día existen en el mercado son de gran utilidad dentro de un proyecto de software, aunque estos tienden algunos a ser costosas sus licencias. Existen también unas que otras que son open sources y que también cubren las necesidades primordiales se puede decir del ciclo de vida del software. El Sistema Integral de Administración de Requerimientos (SIAR), como sistema open sources y creado en ambiente web, permite gestionar los requerimientos funcionales mediante el Lenguaje Unificado de Modelado (UML), se puede administrar los atributos de caso de uso, diseñarlos, validar el modelo conceptual, gestionar los alcances del proyecto, su clasificación, registrar consultas y visualizarlas, SIAR nos ofrece un método automatizado de cohesión de validación del caso de uso desde un punto de vista de transacción de estados definidos en la especificación funcional.

IBM Rational DOORS Next generation, es una herramienta de pago que su función es la administración de requisitos este puede implementarse en las instalaciones o en la nube en un entorno de software como servicio (SaaS). Una solución totalmente basada en la web que incluye definición de requisitos visuales, planificación, gestión de elementos de trabajo e integración de pruebas.

REFERENCIAS BIBLIOGRÁFICAS ADDIN Mendeley Bibliography CSL_BIBLIOGRAPHY Arcadio, J. (2011). Usabilidad Ágil y Reingeniería de Sitios WEB: USABAGLEWeb. Barba-Guaman, L., Quezada-Sarmiento, P. A., Calderon-Cordova, C., & Lopez, J. P. O. (2017). Detection of the characters from the license plates by cascade classifiers method. *FTC 2016 - Proceedings of Future Technologies Conference*, (December), 560–566. <https://doi.org/10.1109/FTC.2016.7821662> Cadavid, N., Martínez, F., Daniel, J., & Vélez, M. (2013). Redalyc.Revisión de metodologías ágiles para el desarrollo de software. Diego Alberto Godoy; Edgardo Belloni; Henry Kotynski; Hector Dos Santos. (2014). WICC 2014 XVI Workshop de Investigadores en Ciencias de la Computación WICC 2014 XVI Workshop de Investigadores en Ciencias de la Computación, 485–489. dos Santos, V. S. A., & Canedo, E. D. (2014). Agile methodology in the software development: Case study: Electoral justice of Brazil. *2014 9th Iberian Conference on Information Systems and Technologies (CISTI)*, 1–6. <https://doi.org/10.1109/CISTI.2014.6876981> Goñi, A., Ibáñez, J., Iturrioz, J., & Vadillo, J. Á. (2014).

0: <http://upcommons.upc.edu/handle/2099/15461>

100%

Aprendizaje Basado en Proyectos usando metodologías ágiles para una asignatura básica de Ingeniería del Software, 133–140.

IBM. (2013). Data Sheet - IBM Rational DOORS Next Generation. Retrieved from <http://www-03.ibm.com/software/products/en/ratidoor> Leiva Mundaca, I., & Villalobos Abarca, M. (2015). Método ágil híbrido para desarrollar software en dispositivos móviles. *Ingeniare. Revista Chilena de Ingeniería*, 23(3), 473–488. <https://doi.org/10.4067/S0718-33052015000300016> Medina, O. C., Marciszack, M. M., Groppo, M. A., Claudia, C., Carlos, M. J., & Humberto, M. E. (2016). Software para la gestión de requerimientos del Modelo Conceptual de un sistema de información Resúmen Introducción Líneas de Investigación , Desarrollo e Innovación. *Red de Universidades Con Carreras En Informática (RedUNCI)*, (978-950-698-377-2), 433–437. Patel, A., Seyfi, A., Taghavi, M., Wills, C., Na, L., Latih, R., & Misra, S. (2012). A COMPARATIVE STUDY OF AGILE , COMPONENT-BASED , ASPECT-ORIENTED AND MASHUP SOFTWARE DEVELOPMENT METHODS, 175–189. Rivadeneira, S., Vilanova, G., Miranda, M., & Cruz, D. (2013). El modelado de requerimientos en las metodologías ágiles. *XV Workshop de Investigadores En Ciencias de La Computación*, 383–387. Retrieved from <http://hdl.handle.net/10915/27196> Silvia Gabriela, R. M. (2012). Metodologías ágiles enfocadas al modelado de requerimientos. *Universidad Nacional de La Patagonia Austral*, 5(1), 1–29. Retrieved from <https://dialnet.unirioja.es/descarga/articulo/5123612.pdf> Trail, A. (2017). *Intland Software Releases codeBeamer ALM 9 . 0*.

Hit and source - focused comparison, Side by Side:

Left side: As student entered the text in the submitted document.

Right side: As the text appears in the source.

Instances from: <http://upcommons.upc.edu/handle/2099/15461>

0: <http://upcommons.upc.edu/handle/2099/15461> 100%

Aprendizaje Basado en Proyectos usando metodologías ágiles para una asignatura básica de Ingeniería del Software, 133-140.

0: <http://upcommons.upc.edu/handle/2099/15461> 100%

Aprendizaje Basado en Proyectos usando metodologías ágiles para una asignatura básica de Ingeniería del Software
