

Propuesta metodológica para el desarrollo de software en proyectos de titulación en la especialidad de Ingeniería en Sistemas Computacionales

Methodological proposal for software development in degree projects in the specialty of Computer Systems Engineering

Freddy Adrian Moreira Pinargote.

Universidad Técnica de Manabí, Ecuador.
fmoreira5020@fci.edu.ec

José Eduardo Chancay Giler.

Universidad Técnica de Manabí, Ecuador.
jchancay8954@utm.edu.ec

Carlos Luis Pinargote Navarrete.

Universidad Técnica de Manabí, Ecuador.
clpinargote@utm.edu.ec

Marely del Rosario Cruz Felipe.

Universidad Técnica de Manabí, Ecuador.
marely.cruz@fci.edu.ec

RESUMEN.

El objetivo del presente trabajo es proponer una metodología de desarrollo de software para los trabajos de titulación en la especialidad de Ingeniería en Sistemas Computacionales, mediante el estudio analítico e investigativo. Se aplicaron técnicas de la entrevista a un grupo de desarrolladores de software, además se empleó la técnica de la encuesta, que fue aplicada a estudiantes y docentes de la Facultad de Ciencias Informáticas. La cual permitió recabar ciertas características de las metodologías ya existentes, de forma tal que sean adaptables y fácil de ejecutar para el escenario que se maneja en la carrera y el conocimiento que poseen los estudiantes al emplear una metodología de desarrollo de software en sus proyectos. Como resultado del trabajo, se presenta una metodología híbrida denominada XPRUM, la cual consta de cuatro fases, que integra la metodología XP y Scrum, que permiten conducir el proceso de desarrollo de software en los trabajos de titulación. Promoviendo el uso de la herramienta que nos presenta la metodología Kanban, que permitirá el control de los procesos y sus actividades.

PALABRAS CLAVE.

Desarrollo de software, metodologías ágiles, XP, Scrum, Xprum.



Fecha de recepción: 05-07-2018 Fecha de aceptación: 31-05-2019

Moreira, F. A., Chancay, J. E., Pinargote, C. L. & Cruz, M. R. (2019). Propuesta metodológica para el desarrollo de software en proyectos de titulación en la especialidad de Ingeniería en Sistemas Computacionales

International Journal of Educational Research and Innovation (IJERI), 12, 76-89

ISSN: 2386-4303





ABSTRACT.

The objective of this paper is to propose a software development methodology for the degree works in the specialty of Computer Systems Engineering, through analytical and investigative study. Interview techniques were applied to a group of software developers, and the survey technique was applied, which was applied to students and teachers of the Faculty of Computer Science. Which allowed to collect certain characteristics of the existing methodologies, in such a way that they are adaptable and easy to execute for the scenario that is handled in the career and the knowledge that the students have when using a methodology of development in their projects. As a result of the work, a hybrid methodology called XPRUM is presented, which consists of four phases, which integrates the XP and Scrum methodology, which allows to conduct the software development process in the titration works. Promoting the use of the tool presented by the Kanban Methodology, which allows the control of processes and their activities.

KEY WORDS.

Software development, agile methodologies, XP, Scrum, Xprum

1. Introducción.

En la actualidad se presentan numerosas propuestas metodológicas que inciden en el proceso de desarrollo de software, separadas en dos grandes grupos; las metodologías tradicionales o pesadas y las metodologías ágiles o livianas, estas han demostrado ser efectivas y necesarias en un gran número de proyectos.

1.1. Metodologías existentes.

A continuación, se hacen referencia algunas de ellas.

Entre las metodologías tradicionales podemos citar: Rational Unified Procces (RUP) (Martínez y Martínez, 2014) con Ivar Jacobson a la cabeza hasta su lanzamiento oficial en 1998 como propiedad de IBM; Microsoft Solution Framework (MSF) (Pérez, 2011), que es un compendio de las mejores practicas en cuanto a administración de proyectos, además adapta tanto el modelo cascada y el modelo espiral; TSP "Proceso en Equipo para el Software", es una metodología para dirigir el desarrollo de software además de establecer un entorno donde el trabajo efectivo de equipo sea normal y natural (Cendejas, 2014); Iconix (Amavizca et al., 2014), de Doug Rosenberg y Kendall Scott, que consiste en un lenguaje de modelamiento y un proceso de desarrollo de software práctico. Es un proceso dirigido, como RUP, relativamente pequeño y ligero, como XP.

Entre las metodologías ágiles más destacadas se pueden nombrar: Extreme Programming (XP) (Letelier y Penadés, 2012) de Kent Beck, se caracteriza porque planifica, analiza y diseña en un mismo momento y durante el desarrollo, la cual propone un desarrollo rápido de software; SCRUM (Abrahamsson, Salo, Ronkainen y Warsta, 2017) presentada por Ken Schwaber que también muestra un desarrollo ágil de software, orientada a proyectos en los que cambian rápidamente los requisitos, en donde el desarrollo del proyecto se hace por medio de iteraciones llamadas Sprint; Kanban (Yanina, 2018) simplifica la planificación y la asignación de responsabilidades, en un tablero se representan los procesos del flujo de



Fecha de recepción: 05-07-2018 Fecha de aceptación: 31-05-2019

Moreira, F. A., Chancay, J. E., Pinargote, C. L. & Cruz, M. R. (2019). Propuesta metodológica para el desarrollo de software en proyectos de titulación en la especialidad de Ingeniería en Sistemas Computacionales

International Journal of Educational Research and Innovation (IJERI), 12, 76-89

ISSN: 2386-4303





trabajo, en donde existen tres columnas (Pendiente, En Progreso, Terminado); Dynamic Systems Development Method (DSDM) (Navarro, Fernández y Morales, 2013) utiliza un ciclo de vida iterativo, fragmenta el proyecto en periodos cortos de tiempo y define entregables para cada uno de estos periodos; Agile Unified Process (PUA) (Edeki, 2013) de Scott Ambler que es una versión simplificada del RUP.

Se encontró varios temas referentes sobre la temática de estudio siendo de apoyo a nuestra investigación. Caso similares se encontraron en la web, por ejemplo, una de ellas se da en la Universidad de Holguín de Cuba, donde se realizó una propuesta metodológica para el desarrollo de software educativo, que fueron adaptadas a las características y necesidades de la docencia de dicha Universidad (Mandriaga, Rivero y Leyva, 2016).

A base de eso se compararon y establecieron ciertas características en donde se eligieron puntos relevantes en donde ayudo a tener una base y fortalecer los conocimientos. Estos puntos sirvieron para la realización de la investigación adaptas a nuestro escenario.

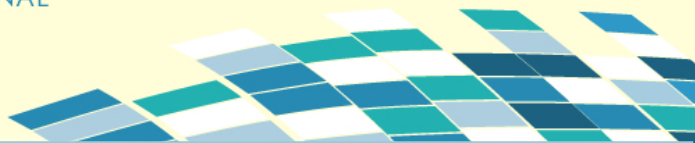
1.2. Metodologías Ágiles y Tradicionales.

Mediante el estudio de un grupo de modelos y metodologías tradicionales de desarrollo de software (Modelo en cascada (Sommerville, 2005), Modelo de desarrollo incremental (Sánchez, Vargas, Reyes y Vidal, 2011), Modelo orientado a la reutilización (Montilva, 2006), Modelo espiral (Tinoco, Rosales, y Salas, 2010), RUP (Martínez y Martínez, 2014)) se logra presentar bajo la investigación ciertas desventajas para el desarrollo de software en los trabajos de tesis, como:

- A veces es difícil que la parte interesada explique todos los requisitos o necesidades antes de comenzar con el proyecto. Por lo tanto, se presentan ciertas dificultades a la hora de implementar los nuevos requisitos durante el proceso de desarrollo de los proyectos.
- Tienen procesos muy controlados y numerosas normas provenientes de estándares seguidos por el entorno de desarrollo por lo que se hace difícil a ciertos cambios.
- El cliente interactúa con el equipo de desarrollo mediante reuniones y además el usuario observa el producto al final, por lo tanto, generan riesgos muy grandes al final del proyecto si no se detecta a tiempo los errores.
- Generan alto costo al momento de poder implementar un cambio durante el proceso ya que manejan proyectos grandes con documentación extensa, utilizando muchos recursos humanos y técnicos.

En general, las metodologías tradicionales buscan imponer disciplina al proceso de desarrollo de software y de esa forma volverlo predecible y eficiente. Para conseguirlo se soportan en un proceso detallado con énfasis en planeación (Navarro, Fernández y Morales, 2013) propio de otras ingenierías. El principal problema de este enfoque es que hay muchas actividades que hacer para seguir la metodología y esto retrasa la etapa de desarrollo.





Las metodologías ágiles tienen dos diferencias fundamentales con las metodologías tradicionales; la primera es que los métodos ágiles son adaptativos, no predictivos. La segunda diferencia es que las metodologías ágiles son orientadas a las personas, no orientadas a los procesos (Navarro, Fernández y Morales, 2013).

El primer hecho es muy importante ya que contrasta con la predictibilidad buscada por las metodologías tradicionales. Con el enfoque de las metodologías ágiles los cambios son eventos esperados que generan valor para el cliente (Patel et al., 2012). Además, las características de los proyectos para los cuales las metodologías ágiles han sido específicamente pensadas, se ajustan a un amplio rango de proyectos de desarrollo; incluyendo con especial énfasis a los equipos de desarrollo pequeños, con plazos reducidos, requisitos ligeros y basados en nuevas tecnologías cambiantes (Letelier y Penadés, 2012). Estas características son semejantes al escenario que se maneja en la especialidad de Ingeniería en Sistemas Computacionales para los trabajos de titulación de la Universidad Técnica de Manabí (UTM). Dentro de los estudios realizados, cabe recalcar que la metodología que se aplica para el desarrollo de software en los proyectos de titulación, al momento es PUA, que es una metodología ágil, que se ha convertido en tendencia y que ha sido adaptada en los últimos años para dichos proyectos, aunque sin tener un estudio técnico sobre las afectaciones al desarrollo del proyecto y la calidad del producto software.

Para el desarrollo de software se acuerdan cuatro principios básicos (Beck et al., 2011), que establecen prioridades y se distinguen de los sistemas tradicionales:

- Individuos e interacciones, por encima de procesos y herramientas, la gente es el principal factor del éxito de un proyecto software, los desarrolladores necesitan adaptarse al trabajo en equipo. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- Software funcionando, por encima de documentación extensiva, la regla a seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental. Si una vez iniciado el proyecto, un nuevo miembro se incorpora al equipo de desarrollo, se considera que los dos elementos que más le van a servir para ponerse al día son: el propio código y la interacción con el equipo.
- La colaboración con el cliente, debe de estar por encima con respecto a la negociación ya sea este como un contrato, con la finalidad que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será el punto de partida del proyecto, asegurando que las necesidades del cliente fueron resueltas obteniendo un gran éxito en todo el proceso del desarrollo del proyecto.
- Respuesta ante el cambio, por encima de seguir un plan, la planificación no debe ser estricta puesto que hay muchas variables en juego, debe ser flexible para poder adaptarse a los cambios que puedan surgir. Una buena estrategia es hacer planificaciones detalladas para unas pocas semanas y planificaciones mucho más abiertas para unos pocos meses.



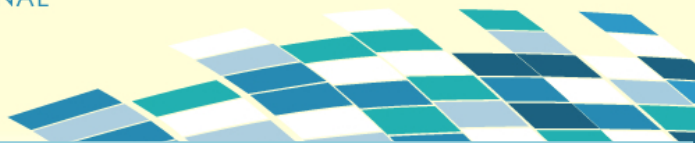
Fecha de recepción: 05-07-2018 Fecha de aceptación: 31-05-2019

Moreira, F. A., Chancay, J. E., Pinargote, C. L. & Cruz, M. R. (2019). Propuesta metodológica para el desarrollo de software en proyectos de titulación en la especialidad de Ingeniería en Sistemas Computacionales

International Journal of Educational Research and Innovation (IJERI), 12, 76-89

ISSN: 2386-4303





1.3. La necesidad.

Dentro de los reglamentos de la Universidad Técnica de Manabí (UTM), se encuentra el reglamento de titulación en el cual hace referencia al proceso de titulación en el nivel de grado para todas las carreras de la UTM; estableciendo que el trabajo de titulación es el resultado investigativo, académico o artístico, en concordancia con el perfil y objetivos de la Carrera, en el cual el estudiante demuestra el manejo integral de los conocimientos adquiridos a lo largo de su formación profesional; mismo que solo podrá ser entregado para sustentación y evaluación cuando haya completado la totalidad de las horas establecidas en el currículo de la carrera. En lo que corresponde a la especialidad de Ingeniería en Sistemas Computacionales, el trabajo de titulación es realizado por lo general no más de dos personas, en donde se tiene un límite mínimo de poder sustentar en seis meses y máximo un año, pudiendo escoger una modalidad para su trabajo de titulación como: propuesta tecnológica, proyecto de investigación, prototipo tecnológico, emprendimiento, dispositivo tecnológico, entre otros. La mayoría de los trabajos de titulación en la carrera son realizados bajo la modalidad de propuesta tecnológica, en la que los estudiantes desarrollan software para la solución de un problema o necesidad social.

La escasa investigación por parte de los estudiantes para poder emplear una metodología de desarrollo de software adecuada para sus respectivos trabajos de titulación y el poco seguimiento que le dan al implementar una metodología, hacen que el software este desarrollado empíricamente y que muchas veces no se logre los resultados esperados, aparte de los inconvenientes que se presentan al no utilizar una metodología de desarrollo de software.

Es por ello que el presente trabajo tiene como objetivo proponer una metodología de desarrollo de software, mediante el estudio analítico e investigativo, recabando ciertas características de las metodologías ya existentes, dirigida para el desarrollo de proyectos de titulación de la especialidad de Ingeniería en Sistemas Computacionales.

2. Materiales y métodos.

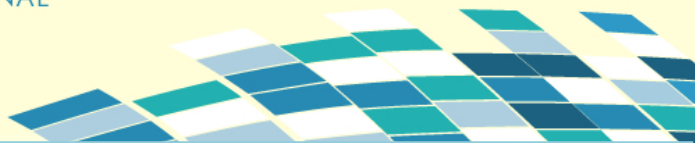
La investigación se llevó a cabo desde octubre de 2017 a febrero de 2018, cuyo fin fue la creación de una propuesta metodológica de desarrollo de software que facilitará en la elaboración de software para los proyectos de titulación en la especialidad de Ingeniería en Sistemas Computacionales. Para el desarrollo de la investigación se emplearon distintos métodos como se nombran a continuación:

Analítico – Sintético, este método permitió potenciar el análisis, la interpretación y la caracterización de los procesos que comprenden el desarrollo de software al emplear metodologías ágiles y llegar a resultados conclusivos y sintéticos.

Descriptivo, ya que se utilizó técnicas de recolección de información, así como, la observación, entrevistas y la encuesta logrando obtener la información con referencia al estudio realizado.

Comparativo, porque permitió establecer mediante la comparación las semejanzas y diferencias existentes entre las distintas metodologías, en cuantos a sus procesos y sus características.





Además, el método Documental-Bibliográfico, el cual se empleó para la revisión de la documentación científica, relacionada con el desarrollo de software. Se tomaron como objetos de estudios los trabajos de titulación (tesis) de los años 2016 - 2017 de la especialidad de Ingeniería en Sistemas Computacionales, para analizar las metodologías de desarrollo de software que fueron utilizadas. Obteniendo 53 trabajos de titulación en diferentes tipos de modalidad, con la finalidad de conocer los tipos de metodologías que los estudiantes aplicaron durante sus trabajos, como resultado encontramos que la metodología más utilizada es Proceso Unificado Ágil (PUA).

Para el estudio y aprendizaje de las metodologías de desarrollo de software se revisaron diversos tipos de materiales como artículos, publicaciones, entre otros, donde se pudo adquirir ciertas características de estas metodologías en las que fueron adaptadas a nuestra propuesta metodológica. Teniendo en cuenta el escenario donde va hacer aplicada esta metodología que impone un conjunto de normas, como el tiempo, en donde un proyecto de tesis normalmente será de seis meses y el número de estudiantes que no pueden ser más de dos, o sea proyectos desarrollados en corto tiempo y que participan pocas personas.

Se emplean además el método de Boehm y Turner para caracterizar un proyecto de software. (Velázquez, Barroso, Castro, María, y Gonzalez, 2012).

3. Resultados del trabajo.

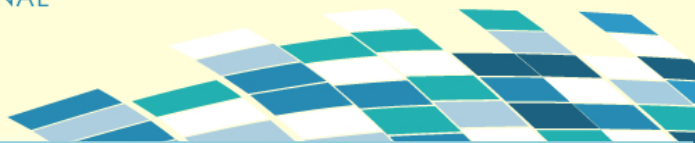
3.1. Estrategia para el desarrollo.

Mediante el método Boehm y Turner, que consta de cinco criterios (Personal, Dinamismo, Cultura, Tiempo, Tamaño y Criticidad) se puede estimar cuan ágil o prescriptivo debía ser el enfoque a utilizar, teniendo como resultado que las metodologías a utilizar en los proyectos de titulación en la especialidad de Ingeniería en Sistemas Computacionales serán de tipo Agiles, ya que dentro del escenario de la presentación de un proyecto de titulación son más adaptables a poder adquirir ciertas característica que nos presentas dichas metodologías.

Dicho lo anterior, para la construcción de la propuesta, se utilizaron cuadros comparativos como herramienta de apoyo, donde se realizó el estudio de estas metodologías y se compararon características de las mismas con el escenario de los proyectos de titulación, específicamente para el desarrollo de software y como podrían adaptarse a ella. Se obtiene características de las metodologías:

- Scrum para la gestión del proyecto de desarrollo de software
- XP para la parte técnica del desarrollo de software y;
- Complementándose con Kanban para controlar el avance del proyecto mediante las tarjetas visuales.





Estas características poseen propiedades que se adaptan al contexto de la Facultad de Ciencias Informáticas para los trabajos de titulación; entiéndase las propiedades académicas del estudiantado universitario de la especialidad de Ingeniería en Sistemas Computacionales, el tiempo de elaboración de un trabajo de titulación, equipo muy reducido de estudiantes para dicho trabajo y requisitos cambiantes a lo largo de la ejecución.

3.2. Análisis de las Entrevistas y Encuestas.

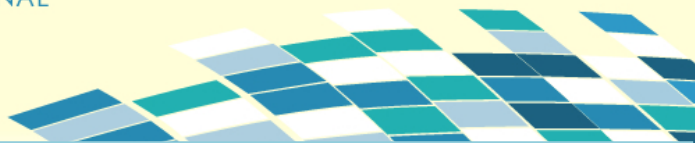
Se realizó una entrevista a cuatro profesionales especializados en el desarrollo de software, que una vez hecho el análisis se detectó que tres de ellos utilizan para el desarrollo de software la metodología Scrum, dado que su empresa se enfoca más en la gestión de proyectos, además siendo esta una de las más utilizada en la actualidad, dos de ellos utilizan una herramienta en la web, basada en la metodología Kanban llamada Trello (Boronat, Montañés, Garcia-Sanoguera, Fenollar y Fombuena, 2017) que les permite gestionar las tareas. Para el último, que aparte de ser desarrollador también cumple la labor de profesor, indica que la metodología XP es muy utilizada para el desarrollo de software y menciona que una característica es la programación en pareja que supone una mejor calidad del código.

Se efectuó una encuesta a 20 docentes de la facultad de Ciencias Informáticas, que una vez analizado cada encuesta se mostraron los siguientes resultados con respecto al tipo de metodología que ellos han aplicado y que recomienda a utilizar en los proyectos de titulación. De acuerdo al escenario que se implantan en la Facultad de Ciencias Informáticas los docentes contestaron con un 90% aplicar las metodologías Ágiles en vez de las metodologías Tradicionales, con respecto a los diferentes metodologías Ágiles que existen los docentes recomienda aplicar en los trabajos de titulación con un 50% la Metodología eXtreme Programming (XP) y con un 30% la metodología Scrum, el 20% restante se derivó en las otras metodologías existentes que están dentro del grupo de las metodologías Ágiles.

Por último en cuanto a la experiencia que posee el docente ellos mencionaron con un 80% que es factible crear una metodología Híbrida que se ajuste a las necesidades de los trabajos de titulación, por motivo que existen escaso control de los procesos de las iteraciones, extensas documentaciones y pocas reuniones entre el equipo y cliente.

Se realizó unas encuestas a 60 estudiantes de la Facultad de Ciencias Informáticas con la finalidad de poder conocer sobre el tipo de metodología que ellos conocen y que han aplicado en los proyectos integradores que se efectúa al final de cada semestre. El 70% contestaron que, si han aplicado en alguna medida una metodología de desarrollo de software en los proyectos integradores, y un 63% mencionaron que usaron las metodologías Ágiles en vez de las Tradicionales. Entre las metodologías Ágiles que han aplicado los estudiantes contestaron con un 60% las metodologías Scrum y en segundo lugar con un 30%, la metodología eXtreme Programming (XP). De acuerdo al análisis de la revisión de las tesis, se llegó a la conclusión que la metodología que se ha estado aplicando es la PUA (Proceso Unificado Ágil), por ello se preguntó al estudiante si conocen y si han aplicado la metodología PUA en algún proyecto de desarrollo de software, el 58%





contestaron que, si la conocen, pero no la han aplicado, un 20% manifestó que si la conocen y que si la han aplicado y el 22% restante mencionaron que no la conocen.

3.3. Metodologías empleadas en los proyectos de titulación en la facultad de ciencias informáticas.

Durante el periodo 2016-2017 se revisaron 53 tesis en las cuales detallan a continuación: Con respecto a su modalidad del proyecto de titulación se encontró que el 77% de las tesis se basan en una Propuesta metodológica, es decir que el otro 23% se derivan en las diferentes modalidades en las cuales encontramos: Trabajo comunitarios, Emprendimiento, Dispositivo Tecnológico, Investigación y Soporte Tecnológico, de acuerdo a los analizado encontramos que un 60% de los proyectos de titulación no aplican una metodología de desarrollo de Software, el otro 40% aplican una metodología que se detalla a continuación: con un 17% aplicaron la metodología PUA (Proceso Unificado Ágil) y con el 23% varios tipos de metodologías tanto para el desarrollo de aplicaciones web y móvil. Se nombran las siguientes: Mobile-D, EORM - Enhanced Object Relationship Methodology, Object Oriented Hypermedia Desing Methodology (OOHDM) y Cascada.

3.4. Propuesta metodológica.

Para esta propuesta, se analizó ciertas características que presenta la metodología eXtreme programming (XP) para llevar a cabo el proceso de desarrollo del proyecto mediante las mejores prácticas que esta ofrece, pretendiendo que el proceso sea efectivo y eficiente. El objetivo es producir rápidamente versiones del sistema que sean operativas, aunque obviamente no cuentan con toda la funcionalidad pretendida para el sistema, pero sí que constituyan un resultado de valor para el negocio.

Para la planificación de los proyectos se analizó la metodología SCRUM que es una forma de gestionar proyectos de software, no es una metodología de diseño, ni de análisis, es una metodología de gestión del trabajo. En la propuesta no se pretende imponer dicha metodología en su totalidad, de la misma se tomó algunas prácticas para su asentamiento en función de que puedan ser adaptables a equipo reducido (dos a cuatro personas) pero que estos, si se rijan por la entrega de prototipos funcionales dentro de plazos cortos y a cambio de reuniones diarias, proponer reuniones semanales (en dependencia de las posibilidades del estudiante).

Adicional a esto se promoverá la utilización del tablero de Kanban o conocido también como tarjetas visuales de la metodología KANBAN, la filosofía de dicha metodología es de poder ejecutar pocas cosas para hacerla de una manera eficaz y muy rápida, evitando así los famosos cuellos de botella. Para esto se puede hacer uso de la herramienta web Trello, el cual también ayudará a controlar mediante la visualización los procesos que se están ejecutando en cada una de las fases que se proponen más adelante, siendo esto opcional para los desarrolladores.

Para la propuesta metodológica, según las necesidades y de acuerdo al escenario que se presenta en la especialidad de Ingeniería en Sistemas Computacionales para los trabajos de titulación, se crea la Metodología XPRUM, con las siguientes fases o etapas por las cuales debe cumplir el proceso de desarrollo de software, entre las cuales se plantean:





- Análisis
- Planeación
- Construcción
- Pruebas y evaluación

Para la mejor comprensión del esquema del modelo de la propuesta de este trabajo, así como la iteración de las fases del mismo se puede visualizar en la Figura 1.

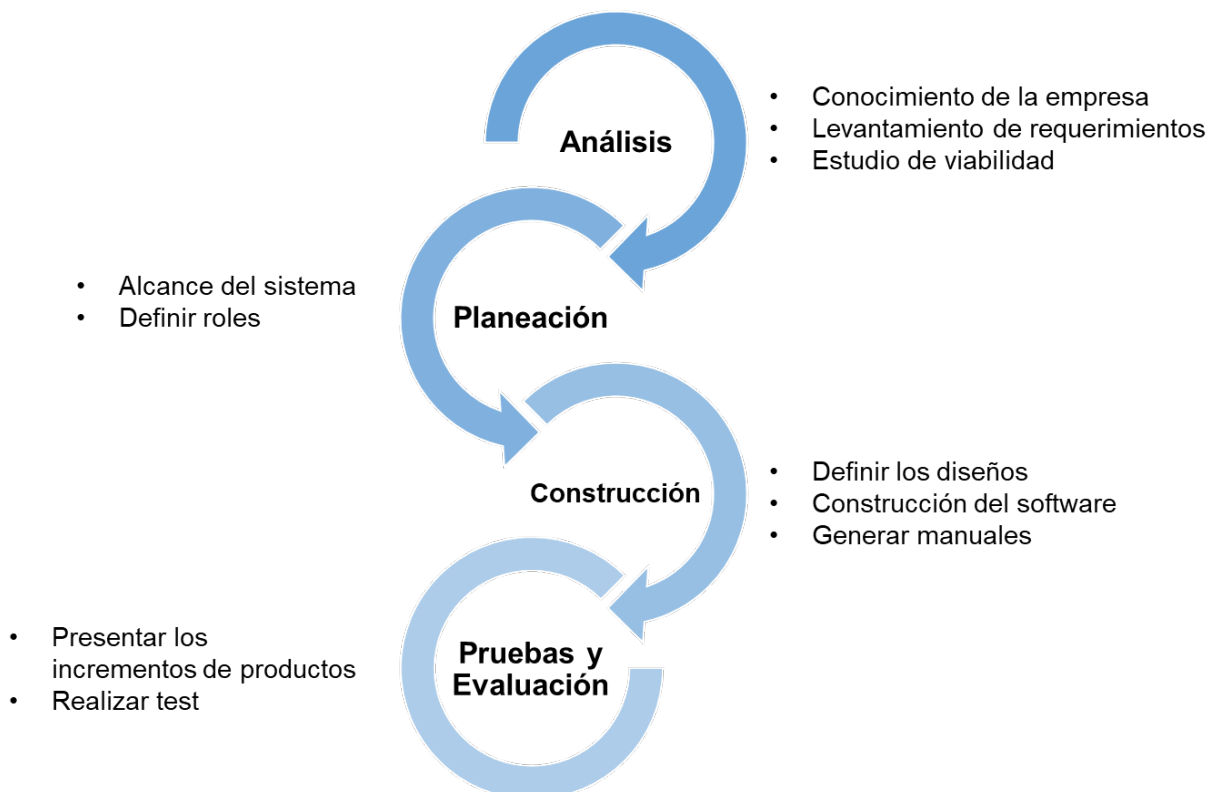


Figura 1. Etapas de la metodología propuesta (Fuente: elaboración propia).

Análisis.

El objetivo principal de esta primera etapa, será de conocer a fondo la situación actual de cómo está la empresa en todos los aspectos es decir en cuanto a su infraestructura de los equipos computacionales (hardware/software), así como también el personal que estén trabajando dentro de cada una de las áreas con sus respectivos cargos o puesto, de igual manera conocer sus necesidades o requerimientos dentro de la aplicación a desarrollar.

Mediante el análisis por lo general, el cliente y el usuario están involucrados, en especial si se necesita cambiar la especificación del sistema, por lo tanto, se necesitará recopilar información adicional, el cual será modelado por el analista.



Planeación.

En la planeación se identifican las funciones del sistema y su vinculación a cada uno de los requerimientos encontrados en la etapa anterior, en donde se debe producir un modelo de clases, en el caso de que sea necesario acompañado del refinamiento del modelo de casos de uso, estos modelos deben ser desarrollados solo a nivel de diagrama UML solo se deben especificar con mayores detalles en caso de que el cliente lo requiera.

Mediante esta etapa de la planeación se conforma el equipo y las herramientas de trabajo las cuales deben de estar enfocadas al fortalecimiento y rendimiento de los procesos, al igual que las políticas de trabajo a la incorporación de nuevas tecnologías en el escenario de los proyectos de titulación, siendo estos importantes para establecer principios de trabajo en equipo como son: comunicación del equipo, integración del docente y/o estudiante en el equipo de trabajo, énfasis en el desarrollo incremental a través de iteraciones cortas mediante un control de las tarjetas visuales. Igualmente se definen aquí las normas colaborativas con el cliente, en este caso la parte interesada, docente y/o estudiante. Para la conformación del grupo de trabajo se plantean los siguientes roles:

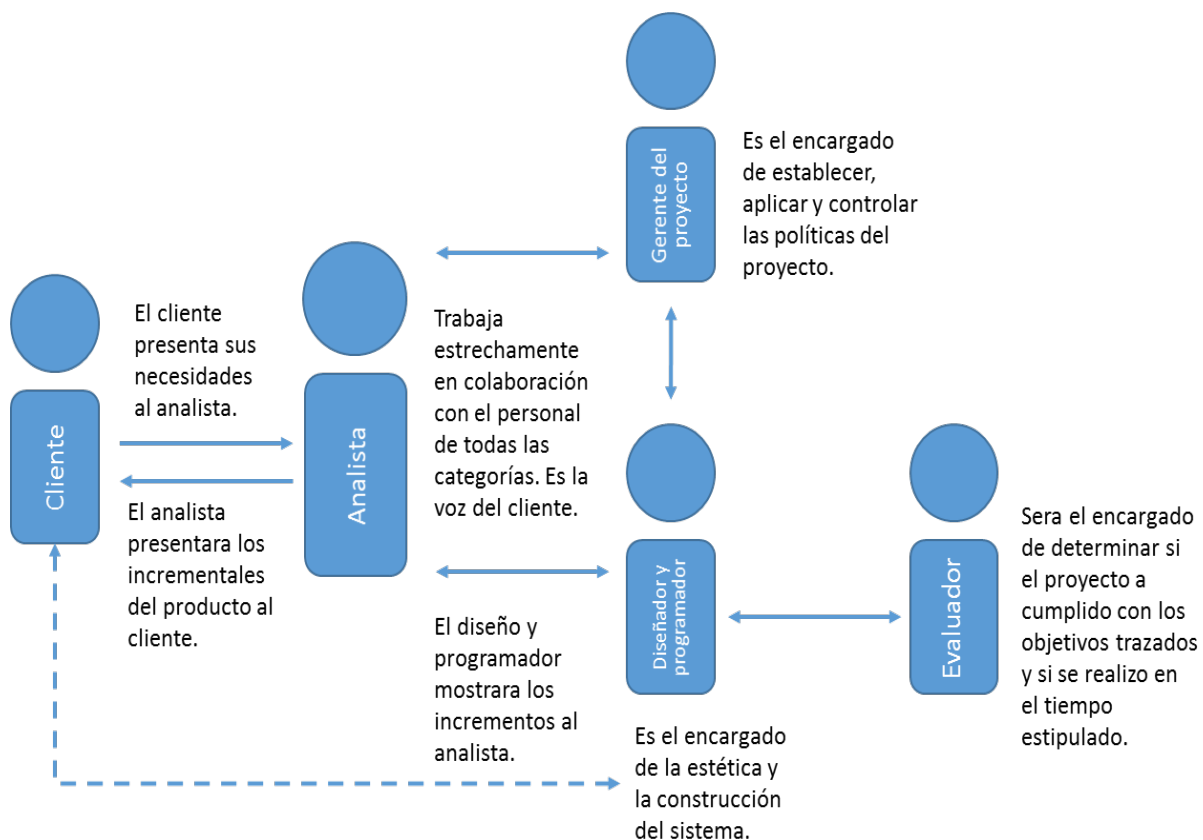


Figura 2. Descripción y relación de los roles (Fuente: elaboración propia).



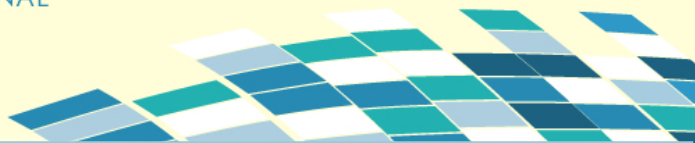
- Gerente de proyecto: Es el encargado de controlar, establecer y aplicar las políticas del proyecto. Así como a reconocer los riesgos que afectan directamente las probabilidades de éxito del proyecto y la constante medición, formal e informal, de dicho riesgo a lo largo del ciclo de vida del proyecto y a modificar sobre la marcha del proceso los aspectos que así lo requieren
- Analista: Es la persona encargada de poder trabajar estrechamente en colaboración con el personal de todas las categorías para averiguar los requisitos y los problemas que surjan en el proyecto existente, además de construir el modelo de análisis del sistema de software. Este identifica los requerimientos que necesita el cliente en el sistema. Construye la arquitectura del sistema junto con sus prototipos y rutas de navegación. Igualmente colaborará en la construcción del programa.
- Diseñador y Programador: Es el encargado de la creación y construcción de las interfaces y de poder programar bajo un lenguaje de programación todo lo solicitado de acuerdo a las necesidades o los requisitos del cliente, junto al analista indagan y priorizan los requisitos a través de la utilización de las tarjetas visuales. Igualmente es el encargado de poder modificar cualquier proceso o aspecto que sea de mejora para el proyecto.
- Cliente: Es la persona que genera las necesidades y presenta el problema actual de la empresa o institución solicitando la creación de un software.
- Evaluador: Es la persona que se encarga de realizar el análisis de un proyecto determinado, que determinara si este cumple o no todos los requisitos que fueron previamente analizados para la construcción del software, informando sí se realizó en el tiempo estipulado a demás evalúa punto a punto cada función y característica que posee. Es una persona ajena al grupo de desarrollo del software.

Durante esta fase se definen los aspectos tecnológicos, metodológicos y organizacionales, en el cual el software se desarrollará. Con respecto a lo tecnológicos se detallan las características de los componentes de hardware y software que son de vital importancia con el objetivo de poder brindar soporte al software, en cuanto al aspecto metodológico se elabora una lista de todas las actividades y estrategia que deben de acompañar al uso del desarrollo de software durante el transcurso de un proyecto de titulación, y por último en el aspecto organizacional se detallan las actividades en relación de la gestión del proceso de desarrollo e implementación del software durante el transcurso de un proyecto de titulación.

Construcción.

Para la fase de construcción, una vez analizados y planificado los requerimientos y las necesidades pasaremos a elaborar o construir los requerimientos funcionales (manipulación de datos, cálculos) y no funcionales (rendimiento, disponibilidad, seguridad) que están dentro del proyecto, durante esta etapa se elaborara un diseño definitivo que satisfaga tanto al grupo de desarrollo como al cliente y sus usuarios.





En esta fase se termina de especificar las historias de usuario que se incluirán en la iteración, las mismas que se diseñan con la arquitectura presentada y se programan todos los componentes definidos y se ejecuta los test de cada uno de estos. Además, para el control se promueve el mecanismo de las tarjetas visuales.

Con estas previas etapas realizadas correctamente y habiendo verificado los diseños se procede al desarrollo e implementación del proyecto realizando dentro de los tiempos acordados el desarrollo de los módulos. Se procederá a traducir los modelos producidos en fases anteriores en código fuente. Además de esto los programadores generaran los manuales, dependiendo de las necesidades y con la información que se consideren relevante, tanto para el usuario, así como para los mismos desarrolladores, estos manuales se deben actualizar, conforme avancen los Sprint.

Pruebas y evaluación.

Esta fase es de vital importancia para todo ciclo de vida del proyecto. Es por esto que esta fase se la fusiono dado que las pruebas serán evaluadas tanto por el equipo de desarrollo como el tutor y revisor del proyecto de titulación, esto a su vez implica que los desarrolladores también deben de tener capacidades de testing. Las pruebas son el proceso de análisis de un sistema, o componentes del sistema, para detectar diferencias entre el comportamiento especificado y el observado. Además, esta fase se centra principalmente en pasar el software desarrollado a un entorno de producción real para que sea aprobado por los usuarios, esto no significa que el software está desarrollado totalmente, sino que contiene funcionalidades útiles para el usuario, es decir, se puede liberar una versión estable con las historias de usuarios incluidas en la planificación del sprint. El objetivo de ésta es aumentar la confiabilidad del sistema.

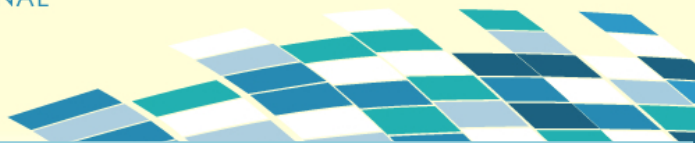
Esta fase no es inherente a un solo proceso del sistema, porque a menudo ocurre que las fallas en una parte del sistema afectan el funcionamiento de otra e igualmente para su resolución se requiera un rediseño de funcionalidades y hasta de necesidades del cliente en vista de la experiencia durante el desarrollo. Aquí también el equipo se apoyará en las tarjetas visuales para la organización y el control de las actividades a realizar en esta etapa.

4. Conclusiones.

Las metodologías ágiles funcionan bien dentro de un contexto específico caracterizado por equipos pequeños de desarrollo, con clientes que pueden tomar decisiones acerca de los requerimientos y su evolución, con requerimientos que cambian con frecuencia (semanal, mensual), con alcance del proyecto o presupuesto variable, con pocas restricciones legales y con pocas restricciones en el proceso de desarrollo. Siendo características casi opuestas a las metodologías tradicionales.

La propuesta de esta metodología fue concebida con la finalidad de adaptarse a los trabajos de titulación que se realizan en la especialidad de Ingeniería en Sistemas Computacionales, tanto en la cantidad de participantes, como en la complejidad y duración del proyecto. Esta también describe un conjunto de fases que permiten guiar el proceso de desarrollo de software, ofreciendo una alternativa de solución al problema descrito en el hecho de que los trabajos de titulación de la especialidad de Ingeniería en Sistemas





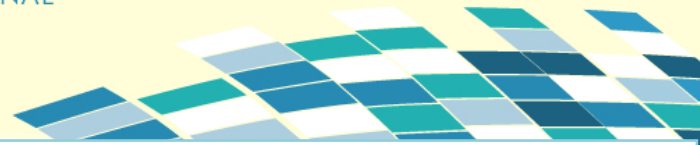
Computacionales, se llevan a cabo mediante una metodología que no se acopla a la realidad de estos trabajos.

Se creó una nueva metodología de desarrollo de software llamada XPRUM, que se orienta a grupos de trabajos reducidos y se construyó a partir de la combinación de algunos procesos de la metodología Scrum, que servirá para la gestión del proyecto y de la metodología XP, para la parte técnica del desarrollo de software, además se integra la metodología Kanban, que permitirá controlar el avance del proyecto mediante las tarjetas visuales.

Referencias bibliográficas.

- Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile software development methods: Review and analysis. *VTT publication 478, Espoo, Finland*, 107.
- Amavizca, L., García, A., Ruíz, Jiménez, E., Duarte, G., & Vázquez, J. (2014). Aplicación de la metodología semi-ágil ICONIX para el desarrollo de software: implementación y publicación de un sitio WEB para una empresa SPIN -OFF en el Sur de Sonora, México, 12, 10. Retrieved from: <http://www.laccei.org/LACCEI2014-Guayaquil/RefereedPapers/RP246.pdf>
- Beck, Kent; Beedle, Mike; Bennekum, Arie van; Cockburn, Alistair; Cunningham, Ward; Fowler, Martin; Grenning, James; Highsmith, Jim; Hunt, Andrew; Jeffries, Ron; Kern, Jon; Marick, Brian; Martin, Robert C.; Mellor, Steve; Schwaber, Ken; Sutherland, Jeff; D. (2011). *Manifiesto por el Desarrollo Ágil de Software*. *Agilemanifesto.Org/Iso/Es*. Retrieved from agilemanifesto.org/iso/es
- Boronat, T., Montañés Muñoz, N., Garcia-Sanoguera, D., Fenollar, O., & Fombuena, V. (2017, July). Utilización de técnicas Kanban para la gestión de tesis doctorales. In *In-Red 2017. III Congreso Nacional de innovación educativa y de docencia en red*. Editorial Universidad Politécnica de Valencia. 110–116.
- Cendejas, J. L. (2014). Implementación del modelo integral colaborativo (MDSIC) como fuente de innovación para el desarrollo ágil de software en las empresas de la zona centro-occidente en México. *Nova Scientia*, 7(13), 268. Retrieved from: <http://www.eumed.net/tesis-doctorales/2014/jlcv/#indice>
- Edeki, C. (2013). Agile unified process. *International Journal of Computer Science*, 1(3), 13–17.
- Letelier, P., & Penadés, M. C. (2012). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).
- Mandriaga, C., Rivero, Y., & Leyva, A. (2016). Propuesta metodológica para desarrollo de software educativo en la Universidad de Holguín / Methodological proposal for educational software development in the University of Holguin. *Ciencias Holguín, Revista Trimestral*, 22(4), 5. Retrieved from <http://www.redalyc.org/pdf/1815/181548029003.pdf>
- Martínez, A., & Martínez, R. (2014). Guía a rational unified process. *Escuela Politécnica Superior de Albacete–Universidad de Castilla la Mancha*.
- Montilva, A. (2006). Desarrollo de Software Basado en Líneas de Productos de Software. *IEEE Computer Society*, 1–34.
- Navarro Cadavid, A., Fernández Martínez, J. D., & Morales Vélez, J. (2013). Revisión de





- metodologías ágiles para el desarrollo de software. *Prospectiva*, 11(2), 30–39. <https://doi.org/10.15665/rp.v11i2.36>
- 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. *Tehnicki Vjesnik / Technical Gazette*, 19(1), 175–189. Retrieved from: http://hrcak.srce.hr/index.php?show=clanak&id_clanak_jezik=117665
 - Pérez, O. A. (2011). Cuatro enfoques metodológicos para el desarrollo de Software RUP–MSF–XP–SCRUM. *INVENTUM*, 6(10), 64-78.
 - Sánchez López, M., Vargas López, M., Reyes Luna, B. A., & Vidal Vásquez, O. L. (2011). Sistema de Información para el Control de Inventarios del Almacén del ITS. Reporte de Proyecto. *Conciencia Tecnológica*, (41), 41–46.
 - Sommerville, I. (2005). *Ingeniería del software*. Pearson Educación.
 - Tinoco Gómez, O., Rosales López, P. P., & Salas Bacalla, J. (2010). Criterios de selección de metodologías de desarrollo de software. *Industrial Data*, 13(2), 70–74.
 - Velázquez, M. B., Barroso, L. C., Castro, E. L., María, A., & Gonzalez, S. (2012). Aplicando el método de Boehm y Turner Applying the Boehm and Turner method. *Serie Científica de la Universidad de las Ciencias Informáticas*, 5(6), 1–12.
 - Yanina, M. (2018). *Conoce las 3 metodologías ágiles más usadas* | *OpenWebinars.net*. Retrieved from: <https://openwebinars.net/blog/conoce-las-3-metodologias-agiles-mas-usadas/>

