

Metodología para el desarrollo del sistema Web para la gestión de los programas de maestría del Instituto “Pedro Kourí”

Development Methodology of the Web System for the Management of Master’s Programs at “Pedro Kourí” institute

Mario Barrientos Rodríguez ¹	0000-0002-8518-3875
Grisel Zacca González ²	0000-0003-4670-5092
Marta Castro Peraza ³	0000-0002-3469-9345
Delmis Álvarez Gainza ⁴	0000-0002-0851-4167
María Vidal Ledo ⁵	0000-0002-0293-5999
Damián Valdés Santiago ⁶	0000-0001-9138-9792

¹Servicios de Pagos RED S.A. La Habana, Cuba. Correo electrónico: mario@redsa.cu (autor para la correspondencia)

²Centro Nacional de Información de Ciencias Médicas, Infomed. La Habana, Cuba. Correo electrónico.

³Universidad de Ciencias Médicas de La Habana. Instituto de Medicina Tropical “Pedro Kourí”. La Habana, Cuba.

⁴Instituto de Medicina Tropical “Pedro Kourí”. La Habana. Cuba.

⁵Escuela Nacional de Salud Pública. La Habana, Cuba.

⁶Facultad de Matemática y Computación. Universidad de La Habana. Cuba.

RESUMEN

Para llevar a cabo proyectos de desarrollo de software y como resultado final obtener un producto de alta calidad, con un bajo costo, un mínimo de esfuerzo y su entrega en tiempo, resultan fundamentales las actividades ejecutadas durante cada fase del proceso de construcción del software, en el que la identificación y elección de la metodología apropiada juegan un papel significativo para el éxito del proyecto. El objetivo del trabajo es proponer una metodología de desarrollo de software en ambiente web, para la gestión de los programas de maestrías del Instituto “Pedro Kourí”. Para llevar a cabo esta tarea se realizó una revisión sobre las metodologías de desarrollo de software existentes, los principios y fundamentos de las mejores prácticas para las metodologías ágiles y las tradicionales más relevantes en la



actualidad, se estableció una comparación entre las metodologías seleccionadas y se propuso la metodología RUP (Rational Unified Process por sus siglas en inglés) para el desarrollo de los procesos de ingeniería de software del sistema web para la gestión de los programas de maestrías del Instituto “Pedro Kouri”.

Palabras clave: ingeniería de software; metodologías de desarrollo de software; metodologías ágiles; metodologías tradicionales; programas de maestrías; instituto “Pedro Kouri”.

Abstract

To carry out software development projects and as a final result obtain a high-quality product, with a low cost, a minimum of effort and its delivery on time, the activities carried out during each phase of the software construction process are fundamental, in which the identification and choice of the appropriate methodology play a significant role for the success of the project. The objective of the work is to propose a methodology for software development in a web environment, for the management of the master's programs of the “Pedro Kouri” Institute. To carry out this task, a review is carried out on the existing software development methodologies, the principles and foundations of best practices for agile methodologies and the most relevant traditional ones today, a comparison is established between the selected methodologies and the Rational Unified Process (RUP) methodology is proposed for the development of the software engineering processes of the web system for the management of the master's programs of the “Pedro Kouri” Institute.

Keywords: software engineering; software development methodologies; agile methodologies; traditional methodologies; master's programs; “Pedro Kouri” institute.

Recibido: 16/03/2022

Aprobado: 03/10/2022

Introducción

El avance experimentado por la Industria del desarrollo de software ha propiciado un cambio en las actividades del día a día de la sociedad; hoy está presente en casi todas las áreas de interés de los seres humanos: ingeniería, bancos, industria, educación, medicina, ámbito científico, gobierno y en las empresas en general.⁽¹⁾

Desarrollar un software es una tarea compleja y la probabilidad de no satisfacer las expectativas de los clientes y usuarios finales es muy elevada, si no se utiliza por parte del equipo de desarrollo una metodología para su construcción desde la fase inicial.

La selección de la metodología de desarrollo adecuada es indispensable para el éxito de todo



proyecto de desarrollo de software, ya que facilita la realización de todas las tareas necesarias de acuerdo con las características del proyecto y su entorno. ⁽²⁾

En la actualidad existen dos tipos de metodologías de desarrollo de software. Las metodologías tradicionales, primeras en aparecer, garantizan un exquisito proceso de desarrollo del software, donde el seguimiento y la planificación predictiva, herramientas, documentación extensiva y negociación contractual reciben merecida importancia. Luego, como reacción a la filosofía utilizada en las metodologías tradicionales, surgen las metodologías ágiles, basadas en entregas frecuentes de versiones de software funcionales, con mayor relevancia a la planificación adaptativa, colaboración con el cliente y respuestas ante los cambios inherentes al desarrollo de software. ⁽³⁻⁶⁾

A pesar de la existencia de ambos tipos de metodologías de desarrollo, aún persisten problemas en el desarrollo de software relacionados con la calidad y el cumplimiento del tiempo y el presupuesto planificado. Según los resultados del estudio Standish Group 2015 Chaos Report, el 29% de los proyectos ejecutados en el mismo año terminaron con éxito, el 52% fueron renegociados y el 19% fracasaron en su totalidad. ⁽⁷⁾

Uno de los principales problemas relacionados con el proceso de desarrollo de software son los métodos utilizados en su construcción; el débil nivel de rendimiento y eficiencia del grupo de desarrollo, desajuste de los plazos y costes previstos en la planificación, entre otros, constituyen los principales factores de fracaso de ese tipo de proyectos.

La elección de la metodología para el desarrollo de software sigue siendo un aspecto a tratar de manera adecuada, sobre todo en el ámbito de las metodologías tradicionales y de las metodologías ágiles no existe un criterio unificado. ⁽⁸⁾ Según refieren Crêspo Boaventura y otros, la selección de la metodología a utilizar en los proyectos de software actuales, debe ser una evaluación de alternativas técnicas, costos, beneficios, condiciones y riesgos asociados. Una selección inadecuada puede implicar una pérdida de tiempo debido a la ejecución de algunas tareas irrelevantes. ⁽⁹⁾

A partir de esta problemática, este estudio tiene como objetivo proponer una metodología de desarrollo de software, para la gestión de los programas de maestrías del Instituto “Pedro Kourí” (IPK), a través de una comparación entre dos de las metodologías más utilizadas en la actualidad (una tradicional y una ágil).

Métodos

En la presente investigación se realizó un análisis bibliográfico de las dos principales corrientes metodológicas identificadas: desarrollo tradicional frente a las ágiles, sus semejanzas y sus diferencias; sus características que se han ido modificando de acuerdo a las necesidades y los



aspectos relevantes que se han ido desarrollando en el tiempo de la sociedad del conocimiento. Se evidencian cambios pequeños en algunos casos y giros importantes en otros, de ahí la importancia de su estudio.

Resultados

La revisión bibliográfica permitió describir los cambios presentados por las diferentes metodologías y las consideraciones de los principales autores en la temática, para lo cual se plantea el recorrido a fin de evaluar el estado de las mismas en el ambiente del desarrollo de software.

En primer lugar, es fundamental conceptualizar los términos metodología y desarrollo. La palabra metodología está formada por tres vocablos griegos: metá ("más allá"), odós ("camino") y logos ("estudio"); por lo que teniendo en cuenta lo anterior, metodología es definida como el estudio de los métodos para luego determinar cuál es el más adecuado.⁽⁸⁾

Específicamente, el vocablo metodología se define como un "conjunto de métodos coherentes y relacionados por unos principios comunes", mientras que el concepto de desarrollo, es vinculado a la acción de desarrollar o a las consecuencias de este accionar, por lo que resulta necesario, rastrear el significado del verbo desarrollar: se trata de incrementar, agrandar, extender, ampliar o aumentar alguna característica de algo físico (concreto) o intelectual (abstracto). Por lo anterior, se concluye que metodología de desarrollo es: "el estudio y determinación de cuál es el método más adecuado para dar incremento a algo" en este caso al software".⁽⁸⁾

Metodologías de desarrollo tradicional

Las metodologías tradicionales se focalizan en documentación, planificación y procesos.

Según lo sustentado por Crêspo Boaventura y otros, las principales metodologías tradicionales, Rational Unified Process (RUP) y Microsoft Solutions Framework (MSF), centran su atención en llevar una documentación exhaustiva de todo el proyecto.⁽³⁾

Metodología RUP

RUP, es un proceso de ingeniería de software que suministra un enfoque para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta y de mayor calidad para satisfacer las necesidades de los usuarios, que tienen un cumplimiento al final, dentro de un límite de tiempo y presupuesto previsible.^{(3-6),(9-11)}



Las principales características de esta metodología son:

- Forma disciplinada de asignar tareas y responsabilidades.
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo interactivo.
- Administración de requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambio.
- Modelado visual del software.
- Verificación de la calidad del software.

En la Metodología RUP se distinguen 4 fases:

Fase de inicio

Durante esta fase las iteraciones se centran con mayor énfasis en las actividades de modelación de la empresa y en sus requerimientos.

Fase de elaboración

Durante esta fase, el equipo se centra en la realización de los casos de uso, este lleva una serie de requerimientos y pasos; el análisis y el diseño van acumulando las actividades, para empezar una parte de implementación que va a ser orientada a la base de la construcción de todas las especificaciones de la arquitectura del diseño.

Fase de construcción

Durante la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones, se define su análisis y después el diseño y se procede a su implantación y sus respectivas pruebas.

Fase de transición

Durante esta fase se busca garantizar que el producto este bien preparado para su entrega al usuario.

A continuación en la figura 1 se presentan las fases y etapas de la metodología RUP, según los autores Rivas, Corona, Gutiérrez y Hernández.⁽⁸⁾





Fig. 1- Fases y etapas de Rational Unified Process (RUP). Tomado de (Rivas, Corona, Gutiérrez y Hernández, 2015)

Metodologías de desarrollo ágil

De acuerdo con Molina Ríos y otros, son una serie de técnicas para la gestión de proyectos que han surgido como contraposición a los métodos clásicos de gestión.⁽¹²⁾

Sus principios se agrupan en cuatro valores:

- Los individuos y su interacción, por encima de los procesos y las herramientas.
- El software que funciona, frente a la documentación exhaustiva.
- La colaboración con el cliente, por encima de la negociación contractual.
- La respuesta al cambio, por encima del seguimiento de un plan.

Muchas personas asocian las metodologías ágiles con falta de documentación o control sobre el proyecto, pero esto es falso. Lo que se desea es minimizar el impacto de las tareas que no son totalmente imprescindibles para conseguir el objetivo del proyecto en el menor tiempo posible. Se pretende aumentar la eficiencia de las personas involucradas en el proyecto y, como resultado de ello, minimizar el coste.

Entre los principales métodos ágiles tenemos el XP (eXtreme Programming), Scrum, Iconix, Cristal Methods y Agile Unified Process (AUP).^{(11), (13-15)}



Metodología XP

XP es una metodología liviana de desarrollo de software. Es un conjunto de prácticas y reglas para desarrollar software. Está basada en diferentes ideas de cómo enfrentar ambientes cambiantes.⁽¹¹⁾

En vez de planificar, analizar y diseñar para el futuro, XP hace todo esto un poco cada vez, a través de todo el proceso de desarrollo.

Objetivos

La metodología XP tiene dos objetivos primordiales para el correcto desarrollo del proyecto.⁽¹¹⁾

- La satisfacción de cliente: Entendida como dar al cliente lo que necesita y cuando lo necesita, respondiendo rápidamente a las necesidades de este. Uno de los factores importantes en todo proyecto de software es que el sistema software logre el objetivo para el cual fue diseñado y que el equipo de trabajo logre el objetivo para el cual fue contratado, de ahí que el incumplimiento de esto termine con un producto incompleto y un cliente insatisfecho.
- Potenciar al máximo el trabajo en grupo: Todos están involucrados y comprometidos con el desarrollo del software, tanto los jefes como los desarrolladores y los clientes, no hay agentes individuales o aislados al proyecto

Características

Planificación incremental: Se identifican los requerimientos en Historias de Usuario, que son negociadas progresivamente con el cliente.

Entregas pequeñas: Se desarrolla primero la más mínima parte útil que le provea funcionalidad al sistema, y gradualmente se efectúan incrementos que aumentan la funcionalidad de la primera entrega, cada ciclo concluye con una entrega del sistema, tal y como se muestra en la figura 2, ciclo de entrega en XP.



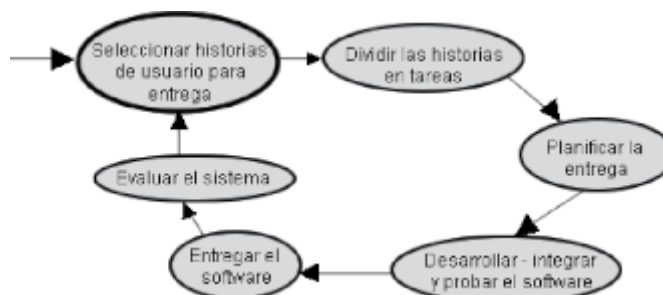


Fig. 2- Ciclo de entrega de Extreme Programming (XP).

Tomado de (Pérez OA, 2011)

Al igual que en RUP, en XP el ciclo de vida (Fig. 3) termina cuando no hay más ciclos de entrega y el sistema ha cumplido el objetivo para el cual fue diseñado, de no ser así, se deberá continuar con el ciclo especificado en la figura 2 hasta que la funcionalidad del sistema sea la deseada.

Diseño sencillo: Solo se ejecuta el diseño necesario para satisfacer los requerimientos actuales, es decir, no se abordan requerimientos futuros.

Desarrollo previamente aprobado: Una de las características relevantes y propias de XP es que primero se efectúan las pruebas y luego se da la codificación, esto con la finalidad de asegurar la satisfacción del requerimiento.

Limpieza del código o refactorización: Consiste en simplificar y optimizar el programa sin perder funcionalidad, es decir, alterar su estructura interna sin afectar su comportamiento externo.

Programación en parejas: Es otra de las características de esta metodología, que propone que los desarrolladores trabajen en parejas en una terminal, comprobando cada uno el trabajo del otro y auxiliándose en la búsqueda de mejores soluciones, garantizando de esta forma un trabajo más eficiente y de mayor calidad.

Propiedad colectiva: El conocimiento y la información deben ser de todos, por lo tanto, no se desarrollan islas de conocimiento, todos los programadores tienen acceso a todo el código y cualquiera puede sugerir y realizar mejoras.

Integración continua: Al terminar una tarea, se integra al sistema entero y se realizan pruebas de unidad a todo el sistema, ésta práctica garantiza una aplicación más funcional en cada iteración y su funcionamiento con los demás módulos del sistema.

Ritmo sostenible: No es aceptable trabajar durante grandes cantidades de horas ya que se considera que puede reducir la calidad del código y la productividad del equipo a mediano plazo, se sugieren 40 horas semanales.

Cliente presente: Se debe tener un representante (cliente o usuario final) a tiempo completo, ya que en XP éste forma parte del equipo de desarrollo, con la responsabilidad de formular los requerimientos para el desarrollo del sistema.



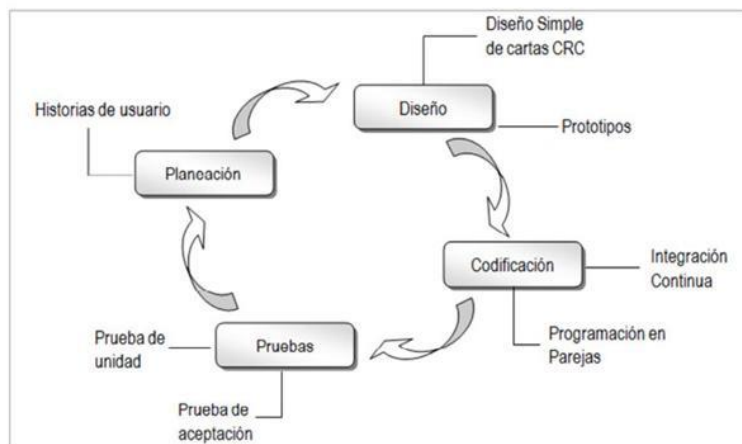


Fig. 3- Ciclo vida de Extreme Programming (XP),

Adaptado de (Pérez OA, 2011).

Principios del Método XP

Este método se basa en cuatro principios bien definidos:

Simplicidad: La simplicidad consiste en desarrollar sólo el sistema que realmente se necesita. Implica resolver en cada momento sólo las necesidades actuales.

Feedback: Una metodología basada en el desarrollo iterativo de pequeñas partes, con entregas y pruebas frecuentes y continuas, proporciona un flujo de retro- información valioso para detectar los problemas o desviaciones y facilitar su corrección.

Decisión: Implica saber tomar decisiones difíciles. Reparar un error cuando se detecta. Mejorar el código tras el feedback y las sucesivas iteraciones.

Comunicación: XP hace casi imposible la falta de comunicación, ya que pone en comunicación directa y continua a clientes y desarrolladores.

Metodologías tradicionales y metodologías ágiles

Las metodologías tradicionales garantizan en los grupos de desarrollo una disciplina de trabajo fundamentada en la documentación sobre el proceso de desarrollo de software, específicamente en la planificación global y total de todo el trabajo a realizar, de manera que una vez todo detallado, comience el ciclo de desarrollo de software; mientras que las metodologías ágiles muchas veces obvian la documentación y se centran en el trabajo, buscando el equilibrio entre proceso/esfuerzo.⁽¹⁶⁾

Surgidas recientemente, las metodologías ágiles constituyen una opción a las metodologías tradicionales, ya que efectúan resistencia a la excesiva documentación de aquéllas. Brooks, en su mítico libro *The Mythical Man Month*, detalla las primeras ideas que se plantean en las



metodologías ágiles; las que, en su gran mayoría, responden al sentido común.⁽¹⁷⁾

La Tabla 1 muestra una comparación entre los aspectos relevantes de las metodologías de desarrollo tradicionales y las metodologías ágiles.⁽¹⁸⁾

Tabla 1. Metodologías tradicionales vs metodologías ágiles (Cadavid et al., 2013).

Metodologías Tradicionales	Metodologías Ágiles
Predictivos	Adaptativos
Orientado a procesos	Orientado a personas
Proceso rígido	Proceso flexible
Se concibe como un solo proyecto	Un proyecto es subdividido en varios proyectos más pequeños
Poca comunicación con el cliente	Comunicación constante con el cliente
Entrega de software al finalizar el desarrollo	Entregas constantes de software
Documentación extensa	Poca documentación

Otros autores concuerdan con la comparación realizada por Cadavid et al, pero le agregan más elementos a esta comparación, los que se relacionan en la Tabla 2.^{(18),(19)}

Tabla 2- Comparación entre Metodologías tradicionales y metodologías ágiles. Tomada de (Canós, Letelier, Penadés, 2014).

Metodologías Tradicionales	Metodologías Ágiles
Se basan en normas provenientes de estándares seguidos por el entorno de desarrollo	Se basan en heurísticas provenientes de prácticas de producción de código
Cierta resistencia a los cambios	Preparados para cambios durante el proyecto
Impuestas externamente	Impuestas internamente por el equipo
Proceso muy controlado, numerosas normas	Proceso menos controlado, con pocos principios
Contrato prefijado	Contrato flexible e incluso inexistente
Cliente interactúa con el equipo de desarrollo mediante reuniones	El cliente es parte del desarrollo
Grupos grandes	Grupos pequeños (<10)
Más artefactos	Pocos artefactos
La arquitectura del software es esencial	Menor énfasis en la arquitectura del software



Selección de metodologías

Este aspecto no ha recibido un tratamiento adecuado, fundamentalmente en el ámbito de las metodologías tradicionales, y en el caso de las ágiles no se ha llegado a un consenso de criterios. Por ello, este artículo se orientó, de acuerdo a la información existente hasta la fecha y a la experiencia personal de los autores, a la formulación de un procedimiento para la selección de una metodología por criterios de conocimientos.

Selección de metodología, por criterios de conocimientos

En función del equipo de desarrollo, se consideran una serie de criterios relacionados con los conocimientos que tengan los integrantes del equipo sobre las metodologías de desarrollo de software a evaluar. Estos criterios se relacionan a continuación:

- Grado de conocimiento
- Soporte orientado a objetos
- Adaptable a cambios
- Basado en casos de uso
- Posee documentación adecuada
- Facilita la integración entre las etapas de desarrollo
- Relación con el Unified Modeling Language (UML)
- Permite desarrollo software sobre cualquier tecnología

Los pesos para cada criterio se establecen en función de los conocimientos que tengan los integrantes del equipo de desarrollo. Por ejemplo, Ríos y Wilson, en su tesis de grado, proponen las siguientes tablas de pesos: ⁽²⁰⁾

- 20% para el grado de conocimiento
- 15% para adaptable a cambios y posee documentación adecuada
- 10% para el resto de criterios

Para determinar la metodología a usar, se realizó una adaptación del cuadro resumen elaborado por Ríos y Wilson para las metodologías RUP y XP. ⁽²⁰⁾

En la evaluación realizada por estos autores, la metodología RUP recibió el mayor puntaje.



Tabla 3- Evaluación de las Metodologías RUP y XP (Adaptado de Ríos y Wilson, 2008).

Criterio	%	RUP	XP	Total
Grado de conocimiento	20	15	10	25
Soporte orientado a objetos	10	10	10	20
Adaptable a cambios	15	10	15	25
Basado en casos de uso	10	10	5	15
Posee documentación adecuada	15	15	10	25
Facilita la integración entre las etapas de desarrollo	10	10	10	20
Relación con UML	10	10	10	20
Permite el desarrollo de software sobre cualquier tecnología	10	10	10	20
Total	100	90	80	

Las diferencias entre corrientes metodológicas (Tablas 1 y 2) obedece a que estas metodologías pueden ser implementadas en diferentes contextos, con diferencias en los requerimientos, en los niveles de riesgo que pueda tener cada proyecto, en los tipos de clientes y en el nivel de calidad, entre otros muchos aspectos, lo cual hace que cada enfoque metodológico sea viable para los negocios con características similares o para un determinado contexto de aplicación.

Situación en el IPK, en cuanto a las Maestrías y desarrollos web utilizados

El IPK es una Unidad de Ciencia y Técnica, que desarrolla la política de trabajo en el campo de la Medicina Tropical en el país. Es la institución de más alto nivel científico en investigación, atención médica y en la formación y actualización docente en Medicina Tropical, dentro del Sistema de Salud Cubano.⁽²¹⁾

La integración asistencial-investigativa-docente del IPK, es de primordial importancia para el Centro de Posgrado IPK, de la Universidad de Ciencias Médicas de La Habana (UCHM), con aseguramiento de los resultados en el campo de la actividad docente, en las diversas modalidades del posgrado, además de organizar la superación profesional, con una serie de cursos de actualización y complementación, imprescindibles para mantener al día a profesionales y técnicos en las diferentes actividades de la asistencia hospitalaria, el diagnóstico microbiológico y parasitológico, los aspectos de la vigilancia epidemiológica, la entomología y el control de vectores.⁽²²⁾

Toda la información relacionada con los programas de Maestrías del IPK se genera en el departamento de Docencia, como órgano responsable de los distintos procesos docentes, en el que la gestión de la información se realiza fundamentalmente en un grupo de bases de datos en Access desarrolladas en el propio departamento y en la UCMH y una pequeña minoría de forma manual en ficheros Word y Excel. En ocasiones los especialistas responsables de los



programas de maestrías han confrontado dificultades para obtener con inmediatez una información en específico, aunque el dato esté registrado, se dificulta ofrecer rápidamente la información solicitada, dado el cúmulo de actividades que se realizan en este departamento.

Al momento de realizar los informes correspondientes a los programas de maestrías para los organismos superiores, los especialistas del departamento docente deben revisar un gran número de documentos en sus ordenadores, o demandar información a los demás profesores de la institución, lo que retrasa o dificulta la entrega de la información solicitada. Además, no existe un sistema de almacenamiento y conservación de la documentación existente que facilite su búsqueda y recuperación, lo que ocasiona falta de noción precisa de lo que se tiene, generación duplicada de información, gastos excesivos de tiempo y dificultad para llevar el seguimiento de la información generada por la institución.

Todas las consideraciones planteadas con anterioridad justificaron el desarrollo de una herramienta informática, en ambiente web, de acceso restringido para profesores involucrados, que permitiera la gestión automatizada de la información relacionada con los programas de maestrías del IPK, facilitando la búsqueda y recuperación de la documentación existente, disminuyendo la generación duplicada de información, los gastos excesivos de tiempo y las dificultades para el seguimiento de la información de la institución.

Idoneidad de la metodología propuesta para dar respuesta al objetivo

Las metodologías tradicionales, por ejemplo RUP, son bastante sistemáticas en su proceso, lo que implica altos niveles de dedicación a la planificación y documentación para poder conseguir el desarrollo deseado, sin embargo, han evolucionado y se han transformado en metodologías enfocadas a procesos ágiles, tales como RUP Ágil, ya que el mercado y la industria del software precisa la obtención de resultados en poco tiempo y dispuestos al cambio constante.

RUP es adaptable, según el tipo de proyecto y las características del mismo, haciéndose énfasis en aquellos flujos de trabajo durante la vida del software que reporten más importancia y sean indispensables. Contiene artefactos diseñados en cada una de las fases, que describen detalladamente las características del software desde que se realiza el análisis del problema hasta la entrega final del producto. RUP constituye una metodología de desarrollo que utiliza algunas de las mejores prácticas en desarrollo de software, ajustable perfectamente a proyectos complejos, a grandes equipos de trabajo y cuenta con gran nivel de aceptación entre desarrolladores de software.

Se propone RUP como metodología de desarrollo para el sistema en ambiente web para la gestión de los programas de maestrías del Instituto “Pedro Kouri”, por ser la más completa y abarcadora, pues como refiere la bibliografía analizada, las restantes son casos particulares de esta. Además, XP igual que el resto de los métodos de desarrollo existentes, presenta algunas debilidades, que constituyen riesgos considerables, como dificultades a la hora de una buena obtención de los requisitos para el sistema.



Conclusiones

La revisión de la bibliografía existente sobre las metodologías de desarrollo constituye una evidencia científica que sustenta la afirmación de que no existe una metodología universal para hacer frente a cualquier proyecto de desarrollo de software; cada una de las metodologías puede ser aplicada de acuerdo con el contexto: recurso humano, documentación necesaria, tiempo y disponibilidad del usuario, entre otros.

Todas las metodologías tienen sus limitantes, del mismo modo que las metodologías ágiles son más apropiadas para pequeños y medianos proyectos, no son las más adecuadas para sistemas que demanden de interacciones complejas con otros sistemas, debido a que dichos sistemas requieren de un nivel de precisión muy alto; aunque no todas las metodologías ágiles están basadas en el desarrollo y entrega incremental, comparten los principios del manifiesto ágil para el desarrollo de software.

No es conveniente utilizar una metodología ágil para el desarrollo de un sistema crítico que requiera un análisis detallado de todos los requerimientos para comprender su complejidad e implicaciones, debido a la complejidad y la extrema precisión que pueda tener la captura de requerimientos, proceso en el cual las metodologías ágiles como XP ofrecen excesiva flexibilidad.

La metodología a utilizar debe ajustarse al proceso de desarrollo concreto a realizar, de acuerdo a las condiciones existentes y el entorno organizacional en el momento de su aplicación, lo que no descarta que en el futuro la organización pueda asimilar otro tipo de metodología para este tipo de proyecto.

La metodología RUP, es la propuesta para el desarrollo del sistema en ambiente web para la gestión de los programas de maestrías del Instituto “Pedro Kourí”

Referencias

1. Chavarría AE, Oré SB, Pastor C. Aseguramiento de la Calidad en el Proceso de Desarrollo de Software utilizando CMMI, TSP y PSP. [Internet]. RISTI - Revista Ibérica de Sistemas e Tecnologías de Informação, 2016. [Consultado: 2021 julio 18] (20), 62–77. Disponible en: <https://dx.doi.org/10.17013/risti.20.62-77>
2. Huanca LM, Oré SB. Factores que Afectan la Precisión de la Estimación del Esfuerzo en Proyectos de Software Usando Puntos de Caso de Uso. [Internet]. RISTI - Revista Ibérica de



- Sistemas e Tecnologias de Informação, 2017 [Consultado: 2021 julio 10] (21), 18–32. Disponible en: <https://dx.doi.org/10.17013/risti.21.pi-pf>
3. Crêspo Boaventura J, Peña Herrera E, Verdecia Vicet P, Fustiel Alvarez Y. Elección entre una metodología ágil y tradicional basado en técnicas de soft computing. [Internet] Revista Cubana de Ciencias Informáticas. 2016 [Consultado: 2021 octubre 10]; 10 (Especial UCIENCIA): 145-14. Disponible en: <http://scielo.sld.cu/pdf/rcci/v10s1/rcci11517.pdf>
4. García-Nieto M, Martínez-Trinidad T, Santos-Posadas H, Lugo-Espinosa O, Acosta-Mireles M, Arjona-Suarez E, et al. Aplicación del proceso unificado en el desarrollo de un software que estima el inventario y el crecimiento-rendimiento maderable en plantaciones de eucalipto. [Internet]. MADERAS Y BOSQUES. 2017 [Consultado: 2021 octubre 20]; 23(1): 163-16. Disponible en: <https://dx.doi.org/10.21829/myb.2017.2311557>
5. Quezada-Sarmiento PA. Implementación de una solución web y móvil para la gestión vehicular basada en Arquitectura de Aspectos y metodologías ágiles: Un enfoque educativo de la teoría a la práctica. [Internet]. RISTI Revista Ibérica de Sistemas e Tecnologías de Informação, 2017 [Consultado: 2021 septiembre 18]; (25): 98-14. Disponible en: <https://dx.doi.org/10.17013/risti.25.98-111>
6. Vera Paredes DA, Córdova Martínez LC, López Bermúdez RM, Pacheco Mendoza SR. Análisis de la metodología RUP en el desarrollo de software académico mediante la herramienta DJANGO. [Internet]. RECIMUNDO 2019 [Consultado: 2021 agosto 18]; 3(2): 964-16. Disponible en: [https://doi.org/10.26820/recimundo/3.\(2\).abril.2019.964-979](https://doi.org/10.26820/recimundo/3.(2).abril.2019.964-979)
7. Hastie S, Wojewoda S. Standish Group 2015 Chaos Report. [Internet]. CULTURE & METHODS, 2015 [Consultado: 2021 septiembre 01]. Disponible en: <https://www.infoq.com/articles/standish-chaos-2015>.
8. Rivas CI, Corona VP, Gutierrez JF, Hernández L. Metodologías actuales de desarrollo de software. Revista de Tecnología e Innovación [Internet]. 2015 [Consultado: 2021 agosto 15]; 2(5): 980-986. Disponible en: https://www.ecorfan.org/bolivia/researchjournals/Tecnologia_e_innovacion/vol2num5/Tecnologia_e_Innovacion_Vol2-Num5_6.pdf
9. Cervantes Ojeda J, Gómez Fuentes MC. Taxonomía de los modelos y metodologías de desarrollo de software más utilizados. [Internet]. Universidades. 2012 [Consultado: 21 octubre 20]; (52):37-11. Disponible en: <https://www.redalyc.org/articulo.oa?id=37326902005>
10. Ortega D, Uzcátegui E, Guevara MM. EAI: Un framework de arquitectura empresarial orientado a servicio en correspondencia con MDA. [Internet]. Universidad, Ciencia y Tecnología. 2012 [consultado: 2021 septiembre 18]; 16(62): 23-10. Disponible en: http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S1316-48212012000100003&lng=es&tlng=es
11. Pérez OA. Cuatro enfoques metodológicos para el desarrollo de Software RUP – MSF – XP - SCRUM. [Internet]. Inventum. 2011[consultado: 2021 septiembre 20]; (10): 64-15. Disponible en: <https://revistas.uniminuto.edu/index.php/Inventum/article/download/9/9/>



12. Molina Ríos J, Zea Ordoñez M. Metodologías de desarrollo en Aplicaciones Web. [Internet]. Revista de Postgrado FaCE-UC. 2017 [consultado: 21 octubre 21]; 11 (21): 245-26. Disponible en: <http://arje.bc.uc.edu.ve/ari21/art16.pdf>
13. Calvo-Valverde LA. Metodología iterativa de desarrollo de software para microempresas. [Internet]. Tecnología en Marcha. 2015 [consultado: 21 octubre 10]; 28 (3): 99-17. Disponible en: <https://www.scielo.sa.cr/pdf/tem/v28n3/0379-3982-tem-28-03-00099.pdf>
14. López Menéndez de Jiménez RE. Metodologías Ágiles de Desarrollo de Software aplicadas a la Gestión de Proyectos Empresariales. [Internet]. Revista Tecnológica. 2015 [consultado: 2021 septiembre 25]; (8): 6-6. Disponible en: <http://www.redicces.org.sv/jspui/bitstream/10972/2917/1/Articulo1.pdf>
15. Vite Cevallos H, Montero K, Cuesta J. Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. [Internet]. Espirales: Revista Multidisciplinaria de Investigación. 2018. [consultado: 2021 septiembre 09]. Disponible en: DOI: <https://dx.doi.org/10.31876/re.v2i17.269>
16. Cáceres P, Marcos E, Kybele G. Procesos ágiles para el desarrollo de aplicaciones Web. [Internet]. Taller de Web Engineering de Las Jornadas de Ingeniería del Software y Bases de Datos. 2001. [consultado: 2021 noviembre 1]. Disponible en: <http://dlsi.ua.es/webe01/articulos/s112.pdf>
17. Brooks F. The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition. Ed. Addison Wesley. 2da edition. 1995. [consultado: 2021 noviembre 1]. Disponible en: <http://www.cesarkallas.net/arquivos/livros/informatica/Addison.Wesley.The.Mytical.Month.Esays.on.Software.Engineering.20th.Anniversary.Edition.pdf>
18. Cadavid AN, Fernández Martínez JD, Morales Vélez J. Revisión de metodologías ágiles para el desarrollo de software. [Internet]. 2013. [consultado: 2021 noviembre 1]. Disponible en: https://uac.edu.co/images/stories/publicaciones/revistas_cientificas/prospectiva/volumen-11-no-2/4_articulo_vol_11_2.pdf
19. Canós J, Letelier P, Penadé MC. Metodologías Ágiles en el Desarrollo de Software. [Internet]. Universidad Politécnica de Valencia. 2014. [consultado: 2021 noviembre 1]. Disponible en: <http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf>
20. Ríos E, Wilson S. Desarrollo de un sistema informático para los procesos de cosecha y post cosecha de la camaronera Pampas de Cayanca. Tesis de grado, Facultad de ingeniería de sistemas, Escuela Politécnica Nacional, Quito. 2008. [consultado: 2021 noviembre 1]. Disponible en: <https://bibdigital.epn.edu.ec/bitstream/15000/1072/1/CD-1905.pdf>
21. Cantelar de Francisco N, Tapanes Fernández T, Cabrera Cantelar N, Cantelar Martínez B, Chao Sautié MI, Velázquez Ramos M. Salto cualitativo en las acreditaciones de las maestrías en el Instituto "Pedro Kouri" (IPK) 2004-2014. Revista Cubana de Medicina Tropical [Internet]. 2015 diciembre [citado 2019 octubre 24]; 67(3). Disponible en: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0375-07602015000300013&lng=es.
22. Cabrera-Cantelar N, Cantelar-de-Francisco N, Chao-Sautié M, Cantelar-Martínez B, Valcárcel-Izquierdo N. Impacto de la formación de la especialidad de Microbiología Médica en



Cuba, en el modelo educativo de posgrado del Instituto de Medicina Tropical "Pedro Kourí". Revista Cubana de Medicina Tropical [revista en Internet]. 2017 [citado 2020 febrero 3]; 69(3): [aprox. 0 p.]. Disponible en: <http://www.revmedtropical.sld.cu/index.php/medtropical/article/view/285>

Conflictos de interés

Los autores declaran no tener conflictos de intereses y ajustarse a las normas éticas de publicación.

Declaración de autoría

Mario Barrientos Rodríguez: Ingeniero en Ciencias Informáticas. Máster en Informática en Salud. Servicios de Pagos RED S.A. La Habana, Cuba. Realizó la conceptualización del trabajo, análisis e interpretación de datos, creación del software utilizado, contribuciones sustanciales a la concepción y diseño del trabajo, supervisión y validación de la información, redacción del borrador, revisión y edición del original.

Grisel Zacca González. Doctora en Estomatología. Especialista en Estomatología General Integral. Máster en Investigación en Servicios de Salud, Máster en Bibliotecología y Ciencias de la Información. Doctora en Ciencias de la Información. Investigadora y Profesora auxiliar. Centro Nacional de Información de Ciencias Médicas, Infomed. La Habana, Cuba. Participó en la supervisión, redacción y revisión del trabajo.

Marta Castro Peraza. Doctora en Medicina. Especialista en Psiquiatría. Doctora en Ciencias Médicas. Profesora e Investigadora Titular. Universidad de Ciencias Médicas de La Habana. Instituto de Medicina Tropical "Pedro Kourí". La Habana, Cuba. Participó en el análisis e interpretación de datos, redacción y revisión del trabajo.

Delmis Álvarez Gainza: Ingeniera en Informática. Máster en Epidemiología. Instituto de Medicina Tropical "Pedro Kourí". La Habana, Cuba. Colaboró en la conceptualización del trabajo, análisis e interpretación de datos, creación del software utilizado, contribuciones sustanciales a la concepción y diseño del trabajo, redacción y revisión del borrador.

María Josefina Vidal Ledo. Licenciada en Cibernética-Matemática. Doctora en Ciencias de la Salud. Profesora e Investigadora Titular. Escuela Nacional de Salud. La Habana, Cuba. Participó en la supervisión, redacción y revisión del trabajo.

Damián Valdés Santiago. Licenciado en Ciencias de la Computación. Máster en Ciencias Matemáticas. Profesor Asistente. Facultad de Matemática y Computación. Universidad de La Habana. Cuba. Participó en la supervisión, redacción y revisión del trabajo.

