

ESTRUCTURAS DESGLOSADAS DE TRABAJO (EDT) EN LA GESTIÓN DE ALCANCE DE PROYECTOS DE DESARROLLO DE SOFTWARE

WORK BREAKDOWN STRUCTURES (WBS) IN THE SCOPE MANAGEMENT OF SOFTWARE DEVELOPMENT PROJECTS

MAP. M P Rojas Puentes¹, Msc. C J Parada¹, Msc. JL Leal Pabón¹

¹ *Universidad Francisco de Paula Santander, Grupo de Investigación en Desarrollo e Ingeniería de Software (GIDIS), Av. Gran Colombia No 12E-96 Colsag, Norte de Santander, Colombia. Tel.: 577566, Ext. 201 y 203
email: {pilarrojas, janethpc, jessicalorenalp}@ufps.edu.co*

Resumen: Las EDT y el diccionario de la EDT, conforman la línea base del alcance del proyecto y tiene una estructura jerárquica que representa el trabajo a realizar de los entregables o resultados tangibles, cada una de estas puede ser asignada a un trabajador específico; asimismo, es el insumo para el desarrollo del cronograma. Las EDT en proyectos de software han demostrado claramente que esta técnica puede ser aplicada eficientemente beneficiando la gestión de los proyectos. Como resultado de la investigación se plantea el modelo de EDT para las metodologías de desarrollo de software Scrum, XP, Scrum/XP, Crystal y Kanban como herramienta para los gestores de proyectos de Tecnologías de la información.

Palabras clave: Estructura Desglosada de Trabajo, PMBOK®, PRINCE 2, Desarrollo de Software, Metodologías Ágiles

Abstract: The WBS and the WBS dictionary form the baseline of the project's scope and have a hierarchical structure that represents the work to be carried out on the deliverables or tangible results, each of these can be assigned to a specific worker; likewise, it is the input for the development of the schedule. WBS in software projects have clearly demonstrated that this technique can be applied efficiently benefiting project management. As a result of the research, the EDT model is proposed for the Scrum, XP, Scrum / XP, Crystal and Kanban software development methodologies as a tool for IT project managers.

Keywords: Breakdown Structure of Work, PMBOK, PRINCE 2, software development, agile methodologies

1. INTRODUCCION

Alcanzar el cumplimiento de los objetivos de un proyecto en específico, en particular los proyectos de desarrollo de software, es requerida la implementación de una EDT que precise aquellos esfuerzos necesarios, las asignaciones respectivas de los compromisos a un componente determinado de la organización donde por medio de la EDT se estima un

presupuesto y cronograma conveniente para la ejecución efectiva del trabajo.

Es así, que la implementación de una EDT define y organiza la eficacia sistémica del proyecto e incorpora el trabajo taxativo en la estructuración del alcance previamente aceptado y vigente. En donde la meta proyectada está comprendida en el nivel inferior de los componentes de la EDT, identificados como paquetes de trabajo. El presente artículo realiza un

comparativo entre los estándares de gestión de proyectos más empleados a nivel mundial, que aplican el proceso de crear EDT; además, se analiza la adaptación del concepto de EDT en las metodologías para el desarrollo de software, que facilita el desarrollo de las actividades en esta labor.

2. METODOLOGÍA DE DESARROLLO DEL PROYECTO

2.1 Población y muestra

La investigación realizó un análisis comparativo del desarrollo de las EDT, entre los estándares PMBOK® y PRINCE2, a través de una revisión documental y la adaptación del método de estudio de similitud MSSS para la caracterización de las metodologías de desarrollo de productos software y las herramientas TI en el diseño de las EDT.

En el estudio realizado por Parada et al (2018, p. 3) a una población de 1426 profesionales en Ingeniería de Sistemas, “el 73% manifestaron tener un conocimiento y apropiación de las siguientes metodologías ágiles en el desarrollo de software; Scrum (43,42%), Scrum / XP híbrido (19,70%), XP (10,50%) y Crystal (9,2%)”. Asimismo, Duque et al. (2019), en un estudio realizado a 51 empresas desarrolladoras de software ubicadas en las ciudades de Cali, Medellín, Bogotá y el Eje Cafetero, confirman que dentro de las metodologías ágiles más utilizadas para el desarrollo de software en Colombia son Scrum (81%), XP (11%), Kanban (4%) y adaptación propia (4%).

3. CONCEPTUALIZACIÓN

3.1 Que es EDT

La EDT, en inglés Work Breakdown Structure o WBS, es un instrumento importante de planificación que vinculan objetivos con recursos y actividades en un marco lógico (Tausworthe, 1979), la cual consiste en dividir un proyecto o actividades de alta complejidad en subproyectos, tareas, subtareas, paquetes de trabajo, etc., en sus unidades más manejables, de tal modo que luego se pueda establecer una estructura jerárquica de estas mismas, permitiendo calcular el tiempo requerido para realizar cada una, como las horas de trabajo y el dinero invertido (Bordons, 2018). Los paquetes de trabajo son utilizados para congregar las actividades en donde el trabajo es planeado y controlado (PMI®, 2017).

Los proyectos de software basados en funciones pueden emplear técnicas similares que consisten en utilizar una estructura de desglose de funciones. Cuando un proyecto suministra servicios profesionales, una práctica común es tomar todos los entregables planificados para crear una EDT. Las EDT dividen el trabajo por fases del proyecto (p. Ej., Fase de diseño preliminar, Fase de diseño crítico) deben garantizar que las fases estén claramente separadas por un entregable (p. Ej., Un documento de Revisión preliminar del diseño aprobado o un documento de Revisión de diseño crítico aprobado). (Taylor, 2003)

3.2 Estructuras de EDT:

La estructura de la EDT se puede representar en tres niveles, donde en el segundonivel se establecen las etapas del ciclo de vida del proyecto y el tercer nivel los entregables o productos; asimismo, algunos utilizan el segundonivel para definir los entregables o productos y el tercer nivel de descomposición se establecen los paquetes de trabajo asociados al ciclo de vida para desarrollar el entregable y en otros casos para llegar al nivel de paquetes de trabajo se deberá descomponer a un siguiente nivel. Una EDT con una descomposición jerárquica excesiva puede representar un mayor esfuerzo en recursos y trabajo. Aunque el término EDT se ha empleado como una etiqueta para todos los diagramas jerárquicos del alcance de un proyecto, al instante de la práctica existen distintos tipos de estructuras orientadas a "entregables" como: EDT Orientado a Verbos, EDT Orientado al Sustantivo y EDT de Fase Temporal. (Taylor, 2018)

3.3 EDT – Gestión de Alcance:

Cuando emprendemos un proyecto es de suma importancia asegurarse de que la Gestión del Alcance del Proyecto (GAP) debe ser preciso y medible para poder determinar si se desarrolló lo propuesto (Sanz, 2017). Este define lo que entregará el proyecto, por lo cual el alcance es la visión que los interesados tienen del mismo; a partir de la GAP se puede especificar e inspeccionar qué contiene y qué le falta incluir al proyecto (PM4DEV, 2016).

La GAP consta de una sucesión de tareas y pasos diseñados para ayudar al director del proyecto a gestionar los entregables del mismo, compuesto por la asignación del trabajo, verificación y adaptación del alcance. Helman (2018) indica que la GAP abarca tanto el alcance del producto como el alcance del proyecto.

Los procesos de GAP que se suelen examinar para la definición del alcance es que las entradas cuenten con la carta del proyecto, activos de los procesos organizativos y documentación de requisitos; en las salidas se obtienen la actualización de los documentos del proyecto y lo que se estipuló en el alcance. (Helgason, 2010). La inadecuada GAP, es una de las principales causas de fracaso de los proyectos, debido a que el director ni brindó suficiente tiempo a concretar el trabajo, ni se determinó adecuadamente el alcance por parte de los interesados, ni hubo una buena gestión del alcance agregando trabajo o presupesto al proyecto, conocido comúnmente como variación del alcance, conduciendo a sobre costos más altos de lo planeado y una ampliación en la fecha de culminación del proyecto (PM4DEV, 2016).

Para que lo anterior no suceda es recomendable describir cómo el equipo del proyecto administra la definición del alcance, explica cómo se realiza la verificación del trabajo del proyecto y la gestión y el control del alcance (Helgason, 2010). Esto es, la GAP debe incluir la explicación del proceso aplicado para preparar el enunciado del alcance del proyecto; establece los procesos para la creación de la EDT; se discute sobre cómo verificar la precisión de los entregables y qué proceso se emplean en la aceptación; realiza un resumen de los detalles del proceso en la aprobación de solicitudes de cambio de alcance y la obtención de los formularios de solicitud (Heldman, 2018).

3.4 Estándar PMBOK®

La Guía de los Fundamentos para la Dirección de Proyectos (PMBOK®) fue desarrollada para proporcionar herramientas, técnicas, habilidades, conocimientos y procesos para la dirección de proyectos que pueden aplicarse a la mayoría de los proyectos. Su aplicación puede favorecer el aumento de las posibilidades de éxito del proyecto, aun así, se debe tener en cuenta que los conocimientos descritos no siempre son aplicables del mismo modo para todos los proyectos, por lo que el cómo será aplicado es determinado por la empresa, organización y/o director del proyecto en concordancia a sus características (PMI, 2013).

Este estándar cuenta con 5 grupos de procesos los cuales se congregan a su vez en 10 áreas de conocimiento, empleados en la mayoría de los proyectos. El PMBOK® describe detalladamente los procesos de cada área de conocimiento y su relación con los grupos de procesos, permitiendo que durante la dirección del

proyecto la implementación se ejecute práctica y eficientemente (Fernández et al., 2015).

3.5 Estándar PRINCE2:

El estándar es una metodología aplicable a proyectos en entornos controlados conocidos comúnmente como Projects In Controlled Environments (PRINCE2), este método se encuentra estructurado para llevar a cabo la gestión eficiente de proyectos en la industria informática. Es una herramienta metodológica que brinda buenas prácticas para optimizar el desarrollo de un proyecto y mejora las posibilidades de éxito. (Montes y Pérez, 2014). PRINCE2 es empleado exitosamente en Australia y Canadá, e implementado en empresas como Microsoft, IBM, Ottawa, Health, Ontario, etc. (Buehring, 2020; Ledoux, 2014).

Esa metodología envuelve por medio de temáticas, la calidad, la organización, los planes, el progreso y el riesgo del proyecto, comprendido por un análisis de posibilidades que serán examinados durante el ciclo de vida del proyecto y argumentado a través de todo el proyecto como adquisición de los resultados deseados. (Prendi y Barrero, 2010). PRINCE2 presenta un enfoque orientado más a la práctica que a la enseñanza en la gestión de proyectos, definiendo plantillas, responsabilidades y roles de los distintos interesados, resultando ser una composición de 7 procesos (Hedeman&Seegers, 2009).

La Tabla 1, representa el análisis comparativo del EDT realizado entre los dos estándares.

Tabla 1: Comparativo EDT en PMBOK® y PRINCE2

Variables	PMBOK®	PRINCE2
Estructura	Árbol jerárquico	Descripción detallada de las actividades
Niveles	Min 3	N.A.
Paquetes de trabajo	Nivel operativo, se definen las actividades.	Descripciones detalladas y se realiza el diagrama de flujos
Diccionario	Si contiene. Por cada paquete se detallan: duración, recursos, hitos costos, fechas de inicio/fin, supuestos y criterios de aceptación	Si contiene. Cada paquete contiene descripciones de los productos, técnicas a utilizar, tolerancias, fechas de inicio/fin

3.6. Metodologías de desarrollo de software (MDS)

Para llevar a cabo un proyecto de software, existen distintas MDS, donde se siguen ciertos procesos para lograr la finalización del proyecto; sin embargo, esta es una decisión difícil para el director del proyecto donde se requiere tiempo para seleccionar la metodología que se ajuste más a las condiciones que tiene el proyecto y el producto a desarrollar; es el caso de características como: el tiempo y presupuesto del proyecto (Pagliari, 2007).

Actualmente, existe gran variedad de MDS para la construcción de software. En el presente artículo se realiza sobre algunas de las metodologías más aplicadas.

3.6.1. Metodología de Programación Extrema - XP

Este método divide el proceso de desarrollo de software en partes pequeñas, en lugar de planificar, diseñar y desarrollar softwares completos. La desagregación de la especificación de los requerimientos logra reducir los costos de cambiar el programa a lo largo del proceso de desarrollo. Las tareas están separadas de la participación de cualquier programador, incluso si el programa no está escrito, el código y el códec se ve como dos desarrolladores que usan la misma computadora. Los propietarios de proyectos pueden agregar fácilmente nuevos requisitos al proceso mediante este método; por lo tanto, se adapta fácilmente a las pequeñas, medianas y grandes empresas (Grünloh et al., 2015).

3.6.2. Metodología Scrum

Bajo esa metodología el desarrollo de software se lleva a cabo en un entorno complejo, donde los requisitos de software creados por el propietario del proyecto se denominan historias y todas las historias componen la cartera de productos, por lo que el enfoque durante el progreso se basa en las reuniones diarias de 15 minutos llamadas Daily Scrum. Una tarea no puede ser asignada ni definida por el director del proyecto ni por nadie más, lo que hace que la tarea sea un proceso realizado junto con los integrantes del equipo; los propietarios del proyecto reciben comentarios después del final de cada sprint, siendo la metodología adecuada para pequeñas, medianas y grandes empresas (Haslwanter y Fitzpatrick, 2017).

3.6.3. Metodología Crystal

La metodología se enfoca y les da más importancia a las personas que a las herramientas o procesos, el método Crystal incluye muchos elementos metodológicos y no tratan todos los proyectos de la misma forma, sino que utilizan procesos y herramientas personalizadas de acuerdo a la naturaleza del proyecto. En la metodología Crystal, se desarrolla y utiliza solo aquellas metodologías requeridas para el trabajo o proyecto y también se cuenta con un enfoque iterativo, este recibe comentarios del propietario del proyecto después de finalizar cada repetición, siendo apropiado para pequeñas, medianas y grandes empresas (Turner y McGee-Lennon, 2013).

3.6.4. Metodología Scrum-XP

Metodologías híbridas como Scrum/XP, permiten aplicar acabadidad la filosofía del agilismo. A partir de este conjunto de metodologías se puede desarrollar requerimientos funcionales desde un inicio que son parte del producto en construcción. Como toda metodología ágil, Scrum/xp se adapta a los cambios que surgen durante el desarrollo del producto; a partir de Scrum se realiza la gestión del proyecto y con XP y Scrum se desarrolla el producto. (Carrasco et al., 2019).

3.6.5. Metodología Kanban

Kanban es un método de gestión de producción apoyado en un flujo de trabajo visual que se fundamenta en la autogestión de los procesos, eliminando la programación centralizada. Kanban permite realizar una disposición de las tareas, permitiendo la monitorización del flujo de trabajo para realizar los ajustes convenientes a lo largo del proceso que permitan incrementar su eficiencia. Es una herramienta para conseguir la producción "Justo a tiempo –JIT–". (Arango et al., 2015). Kanban promueve los principios de: calidad se aplica desde el principio; minimización del desperdicio; mejora continua; flexibilidad; construcción y mantenimiento a largo plazo.

4. RESULTADOS

Las buenas prácticas que entregan los estándares de la administración de proyectos como PMBOK® y PRINCE2 permiten a los gerentes de proyectos aplicar los procesos que se ajusten a la naturaleza del proyecto y a los activos y factores ambientales del negocio, el cual aporta el camino para dar cumplimiento a los objetivos del proyecto.

Dentro de la administración de proyectos, se presenta la etapa de planeación con el alcance, el cronograma y el estimado de los fondos para el avance de las actividades; asimismo, los acuerdos para suplir aquellos requisitos y necesidades de los clientes junto con la calidad de los productos. (Gáfaró et al, 2015)

En la etapa de planeación se expone la línea base del Alcance del proyecto, en ésta se determina el que se va a entregar al finalizar el proyecto, se diseña el desglose de trabajo que incluye todo el trabajo del proyecto y el diccionario de EDT para conocer el detalle de los paquetes de trabajo.

Según Silva et al (2018), en los proyectos no exitosos se observa que la variable con peor desempeño es el cumplimiento al cronograma con un porcentaje del 44%; donde, se puede inferir que los desempeños bajos se deben a una deficiente EDT, teniendo en cuenta que la EDT representa el trabajo a realizar en el proyecto, con tareas, esfuerzo estimado y los recursos disponibles para el progreso de las actividades; finalmente, la calidad y cumplimiento de la EDT establecerá la calidad del cronograma (Castro, 2017)

La EDT y el diccionario EDT representan una entrada para el desarrollo del cronograma, empleado en el proceso de ejecución y medición del rendimiento y avance del proyecto. En los proyectos de software nos enfrentamos a la selección de una metodología para gestionar el proyecto y una metodología para la construcción del producto software.

En el presente artículo se exponen las MDS más utilizadas en Colombia por los gerentes de proyectos TI y en coherencia con la investigación realizada a los graduados de Ingeniería de Sistemas de la Universidad Francisco de Paula Santander, donde se establece el ciclo de vida y los artefactos de Scrum, XP, Scrum/XP, Crystal y Kanban. Es así que un proceso de desarrollo de software hace énfasis a la estructura utilizada para llegar a la creación de un artefacto de software. También se puede referir a él como “ciclo de vida del software” o “proceso del software”.

En el caso de Scrum se ilustra en la Tabla No. 2 las etapas y artefactos y en la Figura No. 1 la EDT propuesta para la metodología.

Tabla 2: Metodología SCRUM

Fases	Artefactos
Inicio	<ul style="list-style-type: none"> Documento: alcance, tiempos globales, costos, Stakeholder
Planeación general del proyecto	<ul style="list-style-type: none"> Lista de Producto (Product Backlog) Documento sprint, entregables del sprint
Desarrollo de Sprints	<ul style="list-style-type: none"> Plan sprint: cronograma Artefactos por cada etapa del ciclo de vida del producto: análisis, diseño, desarrollo, pruebas Acta entrega conformidad del sprint Lecciones aprendidas del sprint Documentación del usuario
Cierre	<ul style="list-style-type: none"> Acta de cierre

Fuente: Elaboración Propia.

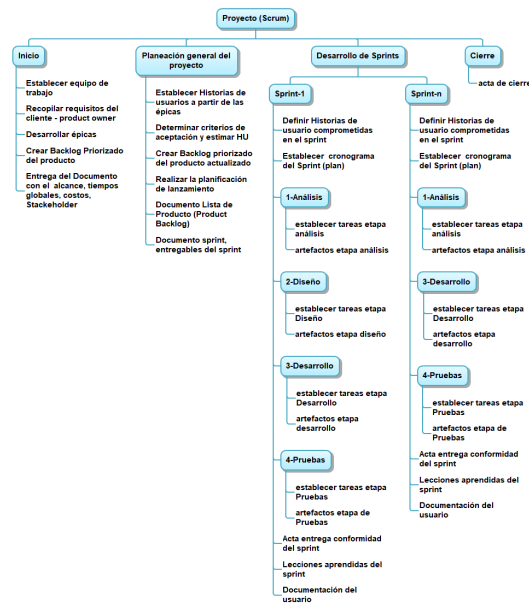


Figura No. 1. EDT propuesta metodología SCRUM. Fuente: Elaboración Propia.

La Tabla 3, expone el ciclo de vida y los artefactos de la metodología XP, junto con la Figura No. 2 con la propuesta de la EDT.

Tabla 3: Metodología XP

Fases	Artefactos
Exploración	<ul style="list-style-type: none"> Historias de usuario
Planificación de entrega	<ul style="list-style-type: none"> Plan de entregas Cronograma de entrega Estimación de esfuerzo
Iteraciones	<ul style="list-style-type: none"> Arquitectura del sistema Plan de iteración
Producción	<ul style="list-style-type: none"> Plan de pruebas En esta fase se actualizan algunos documentos Aprobación del cliente de la entrega realizada
Mantenimiento	<ul style="list-style-type: none"> En esta fase se actualizan algunos documentos
Cierre del proyecto	<ul style="list-style-type: none"> Conformidad de las entregas realizadas

Fuente: adaptación de (Ochoa y Tarazona, 2017)

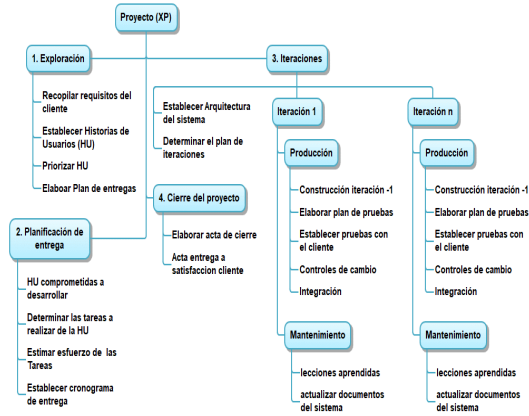


Figura No. 2. EDT propuesta metodología XP Fuente: Elaboración Propia

La Tabla 4, expone el ciclo de vida y los artefactos de la metodología Crystal Clear y la Figura No. 3. Ilustra la EDT propuesta para la aplicación de la metodología.

Tabla 4: Metodología Crystal Clear

Fases	Artefactos
Actividad de Alquiler (Chartering)	<ul style="list-style-type: none"> La Declaración de la Misión ESTRUCTURA DE LOS EQUIPOS Y CONVENIOS: Documento Registro De Stakeholders Plan de lanzamiento Listado de Riesgos El Plan iteración Los casos de uso y el archivo de requerimientos
Ciclo de Entrega (The Delivery Cycle)	<ul style="list-style-type: none"> El modelo de roles de usuario La Descripción de la Arquitectura El modelo de dominio común El Código Fuente El Código de Migración Además, después de cada iteración se deberán realizar también: <ul style="list-style-type: none"> Los resultados del taller de reflexión Documento de estado del proyecto Documento de estado iteración Los informes de errores hasta ese momento
El Ritual De Terminación: El Cierre Del Proyecto (Completion Ritual: The Project Wrap-Up)	<ul style="list-style-type: none"> El texto de ayuda del usuario. Las pruebas

Fuente: adaptación de (Ochoa y Tarazona, 2017)

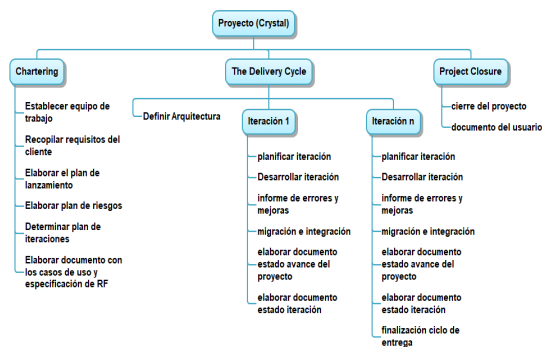


Figura No. 3. EDT propuesta metodología Crystal Clear. Fuente: Elaboración Propia

La Tabla 5, expone el ciclo de vida y los artefactos de la metodología Scrum/XP; asimismo, en la Figura No. 4 se ilustra la EDT respectiva.

Tabla 5: Metodología Scrum/XP

Fases	Artefactos
Planeación	<ul style="list-style-type: none"> Historias de usuario
Diseño	<ul style="list-style-type: none"> Product back log Spring back log Documento de análisis Plan de pruebas Arquitectura del sistema
Desarrollo	<ul style="list-style-type: none"> Retroalimentación de algunos entregables
Cierre	<ul style="list-style-type: none"> Acta de cierre

Fuente: Elaboración Propia

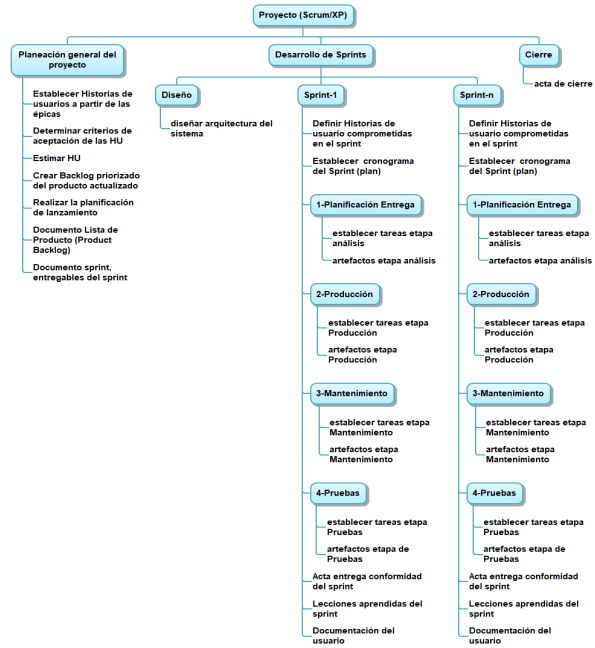


Figura No. 4. EDT propuesta metodología Scrum/XP. Fuente: Elaboración Propia

La Tabla 6, expone el ciclo de vida y los artefactos de la metodología Kanban y la Figura No. 5 se ilustra el Desglose de Trabajo.

Tabla 6: Metodología Kanban

Fases	Artefactos
Priorización de requisitos	<ul style="list-style-type: none"> Documento con la lista de requisitos a desarrollar que definen los sub entregables o entregables
Actividades en desarrollo	<ul style="list-style-type: none"> Lista de requisitos en desarrollo asociadas al entregable o sub entregables en desarrollo Lista de requisitos para desarrollar en cola
Actividades en testeo	<ul style="list-style-type: none"> Lista de entregables o sub entregables en prueba Documento pruebas para desarrollar en cola
Actividades terminadas	<ul style="list-style-type: none"> Documentación de los entregables terminados

Fuente: Elaboración Propia



Figura No. 5. EDT propuesta metodología Kanban. Fuente: Elaboración Propia

En el mercado hay herramientas de TI disponibles que permiten construir el Desglose de Trabajo de los proyectos, la asignación de los recursos, tiempos y costos de las actividades. Algunas herramientas se sincronizan con Project; donde, se carga la EDT en Project y se podrá dar continuidad al desarrollo del cronograma. Las herramientas que se exponen en la Tabla 7, son de uso licenciado y todas entregan la posibilidad de modo prueba gratuita, excepto Gloomaps que es de uso Free.

Tabla 7: Herramientas TI –EDT

Herramienta TI	URL
GanttProject	https://www.ganttproject.biz/
WBS Char Pro	http://www.criticaltools.com/wbschartprosoftware.htm
WBS Schedule Pro	http://wbsstool.azurewebsites.net/es/Home.aspx
WBS Stool	http://wbsstool.azurewebsites.net/es/Home.aspx
Cacoo	https://cacoo.com/es/examples/wbs-diagram-software
easyproject	https://www.easyproject.com/es/
Planhammer	https://planhammer.io/es/wbs-chart-software.html
Gloomaps	https://www.gloomaps.com/

Fuente: Elaboración Propia

5. CONCLUSIONES

Todos los métodos de desarrollo de software pueden dividirse en dos categorías: tradicional (más antiguo y "más pesado") y ágil (nuevo enfoque, "más ligero"). Es por ende que se ha observado una tendencia de popularidad creciente en la vinculación de las EDT para con estos métodos en muchas organizaciones, por lo que la mayoría de los proyectos están apoyados por estas metodologías otorgando una mayor satisfacción, sensación de eficacia, mejora en la calidad y transparencia.

Elegir la MDS adecuada para un proyecto puede ser fundamental y tener un gran impacto (positivo o negativo) en lo mismo. No existe un modelo universal para todo tipo de proyectos de software, por lo que cada uno de

ellos tiene sus propias ventajas, desventajas y alcance de uso, lo que hace relevante la implementación de buenas prácticas en la gestión de proyectos de TI, empleando la EDT como instrumento de planificación que permita ejecutar el proyecto de manera controlada. Se ha demostrado que las EDT son una herramienta eficaz para los proyectos de desarrollo de software, siendo un elemento clave para la planificación y el éxito del proyecto en varios grados de complejidad. No se afirma que ningún modelo sea mejor que otro, esto depende del proyecto o la aplicación a desarrollar eligiéndose la metodología más se adecue al proyecto. Aunque en lugar de utilizarse una metodología, se puede utilizar un híbrido de dos o más metodologías, dado que la tecnología se desarrolla a un ritmo rápido siendo necesario desarrollar nuevas metodologías.

REFERENCIAS

- Arango Serna, Martín Darío. Campuzano Zapata, Luis Felipe. Zapata Cortes, Julián Andrés. (2015). Mejoramiento de procesos de manufactura utilizando Kanban. Revista Ingenierías Universidad de Medellín, vol. 14, No. 27 pp. 221-234 ISSN 1692-3324. Medellín, Colombia. P 221-233
- Bordons, M. C. de. (2018). Una comparativa de software gratuitos de gestión de proyectos. Retrieved from <https://idus.us.es/handle/11441/88499>
- Buehring, S. (2020). PRINCE2 vs the PMBOK® Guide | A comparison. Retrieved September 9, 2020, from <https://www.knowledgetrain.co.uk/project-management/pmi/prince2-and-pmbok-guide-comparison>
- Carrasco Gonzaga, Marcos Klender. Ocampo Pazos, Willian Javier. Ulloa Meneses, Luis Javier. Azcona Esteban, Jon (2019). Metodología híbrida de desarrollo de software combinando XP y SCRUM. Mikarimin. Revista Científica Multidisciplinaria ISSN 2528-7842. P. 109-116.
- Castro, V. (2017) pm4dev, 2009 serie de gerencia para el desarrollo Gestión del Cronograma del Proyecto Gerencia de proyectos para organizaciones de desarrollo
- Duque Madrid, Nathalia Bibiana y Solórzano Pardo, José Hember (2019). Prácticas y metodologías ágiles en la industria de software en Colombia. Tendencias en la investigación universitaria. Vol. VIII (Una visión desde Latinoamérica) p. 60-72.
- Fernandez, K., Garrido, A., Ramírez, Y., & Perdomo, I. (2015). PMBOK y PRINCE 2 similitudes y diferencias -PMBOK and

- PRINCE 2, similarities and differences. *Revista Científica*, 3(23), 111. <https://doi.org/10.14483/udistrital.jour.RC.2015.23.a9>
- Gáfaró, R. E. M., Contreras, M. R., & Villamizar, L. A. E. (2017). Gestión de alcance en proyectos de desarrollo de videojuegos. *Revista colombiana de tecnologías de avanzada (RCTA)*, 1(27).
- Grünloh, C., Walldius, Å., Hartmann, G., & Gulliksen, J. (2015). Using online reviews as narratives to evoke designer's empathy. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 9296, pp. 298–315). Springer Verlag. https://doi.org/10.1007/978-3-319-22701-6_22
- Haslwanter, J. D., & Fitzpatrick, G. (2017). Why do few assistive technology systems make it to market? The case of the HandyHelper project. *Universal Access in the Information Society*, 16(3), 755–773. <https://doi.org/10.1007/s10209-016-0499-3>
- Hedeman, B., & Seegers, R. (2009). *PRINCE2™ 2009 Edition-A Pocket Guide*. (S. Newton, Ed.) (Van Haren). Van Haren.
- Heldman, K. (2018). *PMP: project management professional exam study guide*. (J. Minatel, Ed.). Indiana: John Wiley & Sons, Inc.
- Helgason, V. (2010). *Project scope management*. University of Reykjavík. Retrieved from https://skemman.is/bitstream/1946/9886/1/M.Sc_CSM_ScopeManagement_VidarHelgason.pdf
- Ledoux, J. (2014). *Prince2 & PMBOK Comparison Demystified*. Retrieved September 9, 2020, from <https://es.slideshare.net/CTESolutions/prince2-pmbok-comparison-demystified-29846454>
- Montes, J. D., & Pérez, M. D. (2014). Comparación de metodologías de gerencia de proyectos Prince2 y Pmbok5. Retrieved from <http://repository.ean.edu.co/handle/10882/7024>
- Ochoa, J., & Tarazona, G. (2017). *Aplicación web para la construcción de edo de proyectos bajo el estándar pmi*. Universidad Francisco de Paula Santander.
- Pagliari, C. (2007). Design and evaluation in ehealth: Challenges and implications for an interdisciplinary field. *Journal of Medical Internet Research*, 9(2), 1–14. <https://doi.org/10.2196/jmir.9.2.e15>
- Parada, Carmen & Puentes, M & Vera-Rivera, Fredy. (2018). Study of the use of agile methodologies in the development of software construction projects in Colombia. *Journal of Physics: Conference Series*. 1126. 012056. 10.1088/1742-6596/1126/1/012056.
- PM4DEV. (2016). *PROJECT MANAGEMENT FOR DEVELOPMENT ORGANIZATIONS*. Project Management for Development Organizations. Retrieved from www.pm4dev.com
- PMI®, A. (2017). *Guide to the Project Management Body of Knowledge*, 157.
- PMI, A. (2013). *Guide to the project management body of knowledge (PMBOK guide)*. In Project Management Institute, 5.
- Prendi, E., & Barrero, A. (2010). *PMBOK vs PRINCE2*. Retrieved September 9, 2020, from <https://www.slideshare.net/evergreenpm/pmbok-vs-prince2>
- Sanz, R. (2017). *La Gestión del Alcance del Proyecto*. Retrieved September 12, 2020, from <https://uv-mdap.com/programa-desarrollado/bloque-i-el-ciclo-de-vida-del-proyecto/modulo-3-planificacion-del-proyecto/la-gestion-del-alcance-del-proyecto/>
- Silva, H & Pérez, Torcoroma & Puentes, M. (2018). Adoption of project management methodologies in Colombia project manager's perspective. *Journal of Physics: Conference Series*. 1126. 012032. 10.1088/1742-6596/1126/1/012032.
- Tausworthe, R. C. (1979). The work breakdown structure in software project management. *Journal of Systems and Software*, 1, 181–186. [https://doi.org/10.1016/0164-1212\(79\)90018-9](https://doi.org/10.1016/0164-1212(79)90018-9)
- Taylor, M. D. (2003). *How to Develop Work Breakdown Structures*. Retrieved from https://cs.anu.edu.au/courses/comp3120/local_docs/readings/HowToDevelopWBS.pdf
- Taylor, M. D. (2018). *WBS Types (Work Breakdown Structure)*. Retrieved September 8, 2020, from <https://project-management.com/wbs-types-work-breakdown-structure/>
- Turner, K., & McGee-Lennon, M. (2013). Advances in telecare over the past 10 years. *Smart Homecare Technology and TeleHealth*, 1, 21. <https://doi.org/10.2147/SHTT.S42674>