

PROYECTO DESARROLLO DE UNA HERRAMIENTA DE SOFTWARE BASADA EN TÉCNICAS DE BIG DATA QUE CONTRIBUYA A LA PERMANENCIA DE LOS ESTUDIANTES EN PROCESOS DE EDUCACION VIRTUAL

Herramienta de Formación Virtual METODOLOGIAS Y MARCO DE TRABAJO

Mauricio H. Chavarro

Versión: 1.0

Fecha: 21/11/2019

HFV



Queda prohibido cualquier tipo de explotación y, en particular, la reproducción, distribución, comunicación pública y/o transformación, total o parcial, por cualquier medio, de este documento sin el previo consentimiento expreso y por escrito de MAURICIO H. CHAVARRO M. mchavama@gmail.com

HOJA DE CONTROL

Organismo	UNAD		
Proyecto	Herramienta Tecnológica para Formación Virtual		
Entregable	METODOLOGIAS Y MARCO DE TRABAJO		
Autor	Mauricio Hernán Chavarro Martínez		
Versión/Edición	01.00	Fecha Versión	21/11/2019
Aprobado por		Fecha Aprobación	21/11/2019
		Nº Total de Páginas	25

REGISTRO DE CAMBIOS

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial	Mauricio H. Chavarro M.	21/11/2019

CONTROL DE DISTRIBUCIÓN

Nombre y Apellidos
UNAD
UJTL
QUEOS - Desarrollo

TABLA DE CONTENIDO

1	Introducción.....	5
1.1	Propósito	5
1.2	Cobertura del Sistema.....	5
1.2.1	Nombre del Sistema	5
1.2.2	Alcance	6
1.3	Implicados	6
1.4	Definiciones, acronimos y abreviaturas	7
1.5	Referencias	11
2	Descripción General.....	12
2.1.1	Acerca del producto	12
2.1.2	Requerimientos para el desarrollo	12
2.1.2.1.	Necesidades del proceso de desarrollo.....	12
2.1.2.2.	Necesidades en cuanto al equipo de trabajo.....	13
2.1.2.3.	Necesidades de Planeación, Ejecución y Control	13
2.1.3	Principios, metodología y Marcos de trabajo escogidos.....	14
3	Aplicación.....	15
3.1	Principios de Agile	15
3.1.1	Aspectos del uso de Agile.....	16
3.1.2	Retos y riesgos del uso de Agile.....	16
3.1.3	Beneficios de la adopción de Agile.	16
3.1.4	Pre-requisitos para el uso de Agile	17
3.1.5	Proceso Agile para el desarrollo de HFV.	17
3.1.5.1.	Chartering.....	18
3.1.5.2.	Descubrimiento de las necesidades de Usuario.....	19
3.1.5.3.	Arquitectura de Alto nivel.....	21
3.2	Marco de Referencia Scrum.....	22
3.2.1	Estimación de Historias de Usuario	22
3.2.2	Pila de Producto (Backlog)	23
3.2.3	Release	23
3.2.4	Pila del Sprint.....	23
3.2.5	Planeación del Sprint.....	24
3.2.6	Scrum Diario	24
3.2.7	Revisión del sprint	24
3.2.8	Retrospectiva del Sprint.....	24
3.3	Anexos	25

1 Introducción

Este documento describe los principios, las metodologías y los marcos de trabajo a aplicar en la Gerencia y Desarrollo del proyecto en las actividades relacionadas con la definición e implementación de la Herramienta de Formación Virtual (HFV).

Estas metodologías, marcos de trabajo y principios no incluyen las actividades relacionadas con las actividades de *Data Mining*, *Business Intelligence*, *Data Science* u otras materias relacionadas y/o similares. Concretamente, este documento no incluye descripciones de la metodología CRISP[8].

Este documento se elabora específicamente para las actividades de definición e implementación de software del proyecto “PRESTAR EL SERVICIO TÉCNICO DEL DESARROLLO DE UNA HERRAMIENTA DE SOFTWARE BASADA EN TÉCNICAS DE BIG DATA QUE CONTRIBUYA A LA PERMANENCIA DE LOS ESTUDIANTES EN PROCESOS DE EDUCACION VIRTUAL”, que en adelante identificaremos por la sigla HFV (anacronismo de Herramienta de Formación Virtual).

1.1 Propósito

El propósito de este documento es el de seguir de guía y establecer un orden y organización en las actividades del grupo de trabajo y demás interesados en el prototipo a implementar.

Adicionalmente, este cumple con los siguientes objetivos:

1. Ilustrar a todos los miembros del equipo e interesados acerca de las metodologías, marcos de trabajo y principios a usar en la implementación de HFV.
2. Referir a todos los miembros del equipo e interesados a documentos que extienden y profundizan en las metodologías, marcos de trabajo y principios a usar en la implementación de HFV.
3. Cumplir con uno de los entregables del proyecto (“1. Informe donde se evidencie de manera explícita las metodologías y marcos de trabajo a usar por el desarrollador de la herramienta”).

1.2 Cobertura del Sistema

1.2.1 Nombre del Sistema

Herramienta de Formación Virtual (HFV).

1.2.2 Alcance

Se debe desarrollar un prototipo de herramienta digital basada en minería y analítica de datos que permita alertar tanto al director de curso o tutor de una plataforma educativa virtual como Moodle, como a los estudiantes respecto al posible bajo desempeño utilizando como insumos un análisis de los logs de comportamiento en la plataforma.

Para lograr este alcance se deberán tener en cuenta el logro de los siguientes objetivos:

- El prototipo deberá alcanzar un nivel de desarrollo TRL 4 [8]. Sobre este prototipo en ese nivel de desarrollo se debe probar la factibilidad de la herramienta.
- El uso de una metodología relacionada con los procesos utilizados (de desarrollo) en las metodologías de proyectos ágiles en conjunto con una metodología genérica para proyectos que involucren minería de datos.
- La herramienta final deberá ser adaptable a la plataforma MOODLE de la UNAD.
- Después de comprobarse su funcionalidad exitosa sobre MOODLE, debe poder ser llevada a las demás plataformas de instituciones de educación superior en el departamento de Antioquia.
- La herramienta deberá permitir que la información transformada y resultante sea procesada y producida de forma estructurada.

Ventajas de la Implementación.

- El sistema permitirá dar servicio a un conjunto de aproximadamente 3.923 estudiantes en el departamento de Antioquia.
- El sistema será escalable en número de estudiantes y regiones.
- El sistema permitirá que el estudiante y el profesor puedan tener una guía que ayude en la mejora en el desempeño del primero, basándose en técnica de Data Mining sobre la información de los LOGS de la plataforma CMS, Moodle, de la UNAD.

1.3 Implicados

- Grupo de Trabajo de implementación en el ciclo de vida de desarrollo del software.
- Interesados.
- Colciencias (Como financiador del Proyecto).

1.4 Definiciones, acronimos y abreviaturas

Abreviatura	Nombre	Descripción
CMS ó VLE	Sistema de Administración de Cursos (por sus siglas en Inglés: Course Management System) Ambiente o Plataforma de Educación Virtual (por sus siglas en Inglés: Virtual Learning Environment)	Es una plataforma basada en la Web para los aspectos digitales de los cursos de estudio, generalmente dentro de las instituciones educativas. Presentan recursos, actividades e interacciones dentro de la estructura del curso y proporcionan las diferentes etapas de evaluación. Los VLE también suelen informar sobre la participación; y tener cierto nivel de integración con otros sistemas institucionales. (<i>Wikipedia.com</i>)
MOODLE		Moodle es una herramienta de gestión de aprendizaje (LMS), o más concretamente de Learning Content Management (LCMS), de distribución libre , escrita en PHP. (<i>Wikipedia.com</i>)
Agile		El desarrollo ágil de software envuelve un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental , donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Así el trabajo es realizado mediante la colaboración de equipos autoorganizados y multidisciplinarios, inmersos en un proceso compartido de toma de decisiones a corto plazo. (<i>Wikipedia.com</i>)
CI/CD		En Ingeniería de Software, CI/CD o CICD se refiere a las prácticas combinadas de Integración Continua (CI) y Despliegue Continuo (CD).
CRISP	Cross-Industry Standard Process for Data Mining	Es un método probado para orientar sus trabajos de minería de datos. [5].

Kanban		Kanban es un método para gestionar el trabajo intelectual, con énfasis en la entrega justo a tiempo, mientras no se sobrecarguen los miembros del equipo. En este enfoque, el proceso, desde la definición de una tarea hasta su entrega al cliente, se muestra para que los participantes lo vean y los miembros del equipo tomen el trabajo de una cola. (<i>Wikipedia.com</i>)
MariaDB		MariaDB es una bifurcación desarrollada por la comunidad, soportada comercialmente, del sistema de gestión de bases de datos relacionales MySQL (RDBMS), de software libre y de código abierto bajo la Licencia Pública General GNU. El desarrollo está liderado por algunos de los desarrolladores de MySQL. (<i>Wikipedia.com</i>)
MongoDB		MongoDB es un programa de base de datos orientado a documentos multiplataforma. Clasificado como un programa de base de datos NoSQL, MongoDB utiliza documentos similares a JSON con esquema. MongoDB es desarrollado por MongoDB Inc. y licenciado bajo la Licencia Pública del Lado del Servidor (SSPL). (<i>Wikipedia.com</i>)
MySQL		MySQL es un sistema de gestión de bases de datos relacionales (RDBMS) de código abierto . ^[6] Su nombre es una combinación de "My", el nombre de la hija del cofundador Michael Widenius , ^[7] y "SQL", la abreviatura de Structured Query Language . (<i>Wikipedia.com</i>)
MySQL Workbench		MySQL Workbench es una herramienta de diseño de bases de datos visuales que integra el desarrollo , administración , diseño de bases de datos , creación y mantenimiento de bases de datos SQL en un único entorno de desarrollo integrado para el sistema de bases de datos MySQL .

MVP	Mínimum Viable Product	Un producto mínimo viable (PMV) es “la versión de un nuevo producto que permite a un equipo recoger con el mínimo esfuerzo la máxima cantidad de conocimiento validado acerca de los consumidores”. Ries, Eric. Minimum Viable Product: a guide . <i>Startup Lessons Learned</i> . August 3, 2009.
NoSQL		Una base de datos NoSQL (que originalmente se refiere a "no SQL " o "no relacional") ^[1] proporciona un mecanismo para el almacenamiento y la recuperación de datos que se modela en medios distintos de las relaciones tabulares utilizadas en las bases de datos relacionales .
Numpy		NumPy es una extensión de Python, que le agrega mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices. (Wikipedia.com)
Pandas		En Computación y Ciencia de datos, pandas es una biblioteca de software escrita como extensión de NumPy para manipulación y análisis de datos para el lenguaje de programación Python. (Wikipedia.com)
PHPMyAdmin		phpMyAdmin es una herramienta de administración gratuita y de código abierto para MySQL y MariaDB .
PMI	Project Management Institute	Organización estadounidense sin fines de lucro que asocia a profesionales relacionados con la Gestión de Proyectos. (Wikipedia.com)
Python		Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. (Wikipedia.com)
SCRUM		Scrum es un marco de trabajo para desarrollo ágil de software.

		(Wikipedia.com)
SQL	Por sus siglas en inglés Structured Query Language ; en español lenguaje de consulta estructurada)	SQL es un lenguaje de dominio específico utilizado en programación, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales. Morteo, Bocalandro, Francisco, Nicolás (2004). <i>Un enfoque práctico de SQL</i> . Ediciones Cooperativas. ISBN 987-1076-61-4 . (Wikipedia.com)
SQL		SQL (Structured Query Language) es un lenguaje específico de dominio utilizado en la programación y diseñado para administrar datos contenidos en un sistema de administración de bases de datos relacionales (RDBMS) o para el procesamiento de secuencias en un sistema de administración de flujo de datos relacional (RDSMS). Es particularmente útil en el manejo de datos estructurados, es decir, datos que incorporan relaciones entre entidades y variables. (Wikipedia.com)
TRL	Technology Readiness Assessment	[8]

1.5 Referencias

- [1] A guide to the project management body of knowledge (PMBOK® guide), Sixth Edition. Project Management Institute (PMI), 2017.
- [2] A Practical Guide to Distributed Scrum. Woodward, Surdek & Ganis. IBM Press, 2017.
- [3] Agile product management with Scrum: creating products that customers love. Roman Pichler. Pearson Education, Inc., 2010.
- [4] Fundamentals of Software Engineering. Carlo Ghezzi. Dino Mandrioli, Mehdi Jazayeri. Pearson, 2003.
- [5] Manual CRISP-DM de IBM SPSS Modeler. IBM. IBM Press, 1992, 2012.
- [6] Scrum Manager, guía de formación, V 2.6. Alexander Menzinsky, Gertrudis López, Juan Palacio. Info 4 Media SL, 2016.
- [7] The Art of Agile Development. James Shore and Shane Warden. 2008 O'Reilly Media, Inc.
- [8] U.S. Department of Energy. Technology Readiness Assessment Guide. DOE G 413.3-4A, 2011.

2 Descripción General

2.1.1 Acerca del producto

HFV es un prototipo de un sistema que permitirá establecer métricas de la interacción de los estudiantes con el CMS a partir del LOGs de transacciones de MOODLE, caracterizar a cada estudiante, modelar su comportamiento y visualizar dichos modelamientos. Finalmente, tanto estudiantes como profesores podrán ser alertados por la herramienta de los riesgos posibles identificados como resultado del análisis resultante de las observaciones dadas por la herramienta.

Este sistema tendrá como principal atributo de calidad la recepción de los insumos puros e inalterados desde el sistema CMS de la Universidad. Adicionalmente, la información que se recibirá tendrá el carácter de reservada y, en la medida de lo posible, se mantendrá confidencial cualquier medición hecha sobre cada uno de los estudiantes.

El Prototipo se ha conceptualizado para trabajar inicialmente sobre un conjunto de estudiantes, pero es posible que se pueda extender a un conjunto mayor de estudiantes de la UNAD.

2.1.2 Requerimientos para el desarrollo

El alcance del desarrollo de la herramienta correspondiente al producto antes descrito es el de un proyecto de implementación de un prototipo TRL 4[8] que cubrirá las etapas del ciclo de vida de desarrollo de software.

Siguiendo lo indicado por las últimas prácticas de tecnología informática, este desarrollo se realizará con el conjunto de principios dictados por Agile [3].

2.1.2.1. Necesidades del proceso de desarrollo

La justificación del desarrollo con el marco de principios de Agile se sustenta en que se requiere un proceso de trabajo que sea, entre otros:

1. Exploratorio,
2. Experimental,
3. Incremental,
4. Adaptivo [4] y
5. Confiable.

La razón de que el desarrollo sea Exploratorio indica que esta característica se implementará a través de un prototipo, cuyos estados iniciales, permitan la elicitación de requerimientos y características del prototipo final requeridas por los investigadores de la UNAD y de la UJTL.

El que este sea Experimental permitirá analizar características técnicas o de implementación del producto final, como por ejemplo la utilización de Librerías de Python como Pandas, Numpy, otros modelos de *Business Intelligence* o de la implementación de cálculos usando estos tipos de Librerías u otro mecanismo técnico como SQL.

La característica de Incremental está directamente relacionada con SCRUM, en el sentido que se entregarán versiones del prototipo con funcionalidades incrementales, de tal forma que se construirá un Producto de Línea Base, que incluso pudiese llegar

a ser un MVP, para sobre él ir adicionando nuevos modelos de caracterización y funcionalidades. Esto está de acuerdo con las entregas continuas a través de Releases de SCRUM. Se analizará incluso la implementación de procesos de despliegue modalidad CI/CD para el prototipo.

Al ser Adaptivo el prototipo de la HFV deberá poder aceptar diferentes requisitos de los interesados, mas que todo en el sentido de granularidad de la información base para las caracterizaciones, así como de los periodos de medición.

Finalmente, la HFV deberá ser Confiable en el sentido que sus resultados deberán ser **correctos** y consistentes con lo requerido, adicionalmente éste deberá tener la disponibilidad requerida en la oportunidad requerida.

2.1.2.2. Necesidades en cuanto al equipo de trabajo

Por otro lado, que la conformación y construcción del Equipo de Trabajo (incluyendo los Interesados) deberá permitir:

1. Comunicación efectiva, frecuente y constante.
2. Trabajo conjunto entre el equipo de desarrollo y los “dueños” del negocio,
3. Conversaciones verbales,
4. Trabajo por objetivos y Resultados,
5. Motivación y
6. Distribución geográfica del equipo.

Cualquier equipo deberá tener una comunicación efectiva, mientras las características de frecuencia y constancia se justifican en cuanto se priorizan estas características antes que la formalización documental de la información, los acuerdos y las definiciones. Aun cuando esto no implica que se desestimen u obvien estos últimos.

Deberá existir una comunicación fluida entre desarrolladores e interesados, de tal forma que no haya zonas “oscuras” que las partes no puedan ver.

Se dará prioridad a las comunicaciones verbales y cara-a-cara antes que las documentales, v.g. papel, e-mail, memorandos, etc. Lo anterior debido a que la comunicación verbal y/o cara-a-cara permite dilucidar dudas muy pronto, a clarificar aspectos de duda y a establecer compromisos entre las partes.

Se tendrán ciclos cortos de trabajos con resultados palpables y tangibles que puedan ser medibles y trazables.

Se requiere un equipo de desarrollo y de interesados (investigadores y profesores, entre otros) altamente motivados, lo cual se logra produciendo resultados muy pronto (*quick wins*).

Debido a que se trabajará con grupos establecidos en dos *campus* universitarios de una metrópoli de difícil y lenta tele-conmutación, de un *campus* diferente en otra ciudad y de un equipo de desarrollo también en sitio diferente, se propenderá por reuniones virtuales remotas para lograr mayor frecuencia de estas.

2.1.2.3. Necesidades de Planeación, Ejecución y Control

Siendo este proceso a ser ejecutado para una entidad estatal, como lo es la UNAD, se requieren mecanismos formales de planeación, ejecución y control, los cuales se realizarán siguiendo los lineamientos dados por las prácticas dictadas por el PMI, mediante el PMBOK versión 6[1].

Sin embargo, no se seguirán los lineamientos estrictos, sino que estos se combinarán con el marco de trabajo de Scrum y algunas herramientas relacionadas para hacerlo consistente con los principios de Agile.

2.1.3 Principios, metodología y Marcos de trabajo escogidos.

Dadas las consideraciones expuestas en el numeral 2.1.2, se escoge para desarrollar la herramienta HFV, lo siguiente:

1. Principios de desarrollo de Software.

Los dictados por Agile.

2. Metodología de Gerencia de Proyectos.

Se usarán elementos de PMI consistentes con los principios de Agile para la Gerencia del Proyecto.

Como alternativas de gerencia de proyectos existen otras, entre las cuales la más conocida es Prince. Sin embargo, se usará PMP por ser la más conocida y usada en Colombia.

3. Marco de Trabajo.

Se usará el marco de trabajo proporcionado por Scrum.

Sin embargo, hay que tener en cuenta que existen otras opciones tales como Kanban.

3 Aplicación

3.1 Principios de Agile

Como se mencionó en el capítulo anterior, para el proyecto HFV se usarán principios de Agile. A continuación, se transcriben los 4 principios del manifiesto Agile:

“

1. Individuos e interacciones sobre procesos y herramientas
2. Software funcionando sobre documentación extensiva
3. Colaboración con el cliente sobre negociación contractual
4. Respuesta ante el cambio sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.”

© 2001

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

El manifiesto Agile fue inicialmente publicado por los 17 científicos anteriores y reside oficialmente en la página <https://agilemanifesto.org/iso/es/manifesto.html>.

Lo más importante de Agile es entender que las afirmaciones de la izquierda son priorizadas sobre las de la derecha sin sustituir u obviar estas últimas. Por lo tanto, es ideal poder complementar estos con principios de metodologías como PMI

Los principios anteriores se complementan con los siguientes 12 para desarrollo de software:

“

1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
3. Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
7. El software funcionando es la medida principal de progreso.
8. Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.

10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para la continuación ajustar y perfeccionar su comportamiento en consecuencia.”.

Estos principios se implementarán a través de los métodos proporcionados por Scrum y PMI.

3.1.1 Aspectos del uso de Agile

Implicaciones del uso de Agile en el proyecto HFV:

1. Adaptivo. En el sentido que el prototipo permitirá:
 - a. Entregar versiones (*releases*) del software de la herramienta (prototipo HFV frecuente y constantemente. Dichas versiones serán incrementales como se especificó antes.
 - b. Aceptar el cambio. Las diferentes versiones permitirán cambios en las fórmulas y granularidad de la selección de información.
 - c. Diseñar bien y producir con excelencia técnica. y
 - d. Mejorar continuamente el producto.

3.1.2 Retos y riesgos del uso de Agile.

Tocará tener especial cuidado en los siguientes aspectos del desarrollo y del producto:

El modelado de la Base de datos, el diseño arquitectónico y el diseño detallado presenta retos importantes debido a la velocidad de desarrollo, entregas e implementaciones continuas, apertura a y frecuencia de los cambios. El riesgo se mitiga utilizando técnicas como:

- Reusabilidad de código, componentes y librerías,
- Bajo acoplamiento entre los componentes,
- Interfaces delgadas,
- Bases de Datos de amplio uso, con esquemas flexibles y herramientas de diseño ágiles. Como por ejemplo el uso de Bases de datos basadas en SQL o NoSQL de amplio uso como MySQL, SQLServer, MariaDB o MongoDB combinadas con herramientas de manejo como MySQLWorkBench o PHPMyAdmin.

Control débil o pérdida de Control o No predictibilidad en los eventos consecuentes con las nuevas versiones (*Releases*) y la velocidad de desarrollo. Lo anterior se mitiga con mecanismos propuestos por Agile y Scrum en el sentido de propiciar comunicación continua, efectiva, cara a cara y en general, con la participación de desarrolladores e interesados (Profesores entre estos) en reuniones frecuentes y continuas, así como de técnicas de control propuestas por la metodología PMI.

3.1.3 Beneficios de la adopción de Agile.

Dentro de los principales beneficios del uso de Agile se cuentan los siguientes, en su orden:

1. Habilidad para manejar las prioridades cambiantes.
2. Visibilidad del proyecto.

3. Productividad el equipo de trabajo incrementada.
4. Alta velocidad de entrega/"Time to market".
5. Alta moral del equipo.
6. Alineación entre el negocio y la TI.
7. Alta Calidad del Software.
8. Predictibilidad del avance del proyecto.
9. Reducción de riesgos del proyecto.
10. Disciplina de Ingeniería.
11. Mantenibilidad del software.
12. Buen manejo de equipos distribuidos.

Adicionalmente, se ha comprobado que el uso de Agile tiene una probabilidad de éxito mayor que el modelo de cascada, independientemente del tamaño del proyecto (ver <https://www.infoq.com/articles/standish-chaos-2015/>).

3.1.4 Pre-requisitos para el uso de Agile

El proyecto de desarrollo requerirá de los siguientes compromisos:

1. Colaboración cercana entre los interesados, investigadores, científicos, estudiantes y asistentes con el equipo de desarrollo.
2. Respaldo y liderazgo del Propietario del Proyecto, el Gerente por parte de la UNAD y el supervisor de esta misma Universidad.
3. El equipo deberá ser construido para digerir el cambio con el uso de buenas prácticas de Ingeniería. Por buenas prácticas de tienen las siguientes:
 - a. Construcción y despliegue automatizado o semi-automatizado.
 - b. Integración continua.
 - c. Pruebas automatizadas o semi-automatizadas.
 - d. Propiedad colectiva del equipo inter-disciplinario.
4. Equipos funcionales cros-funcionales.
5. Ambiente colaborativo.

3.1.5 Proceso Agile para el desarrollo de HFV.

El proceso de desarrollo se circunscribirá al proceso esquematizado en la siguiente figura.

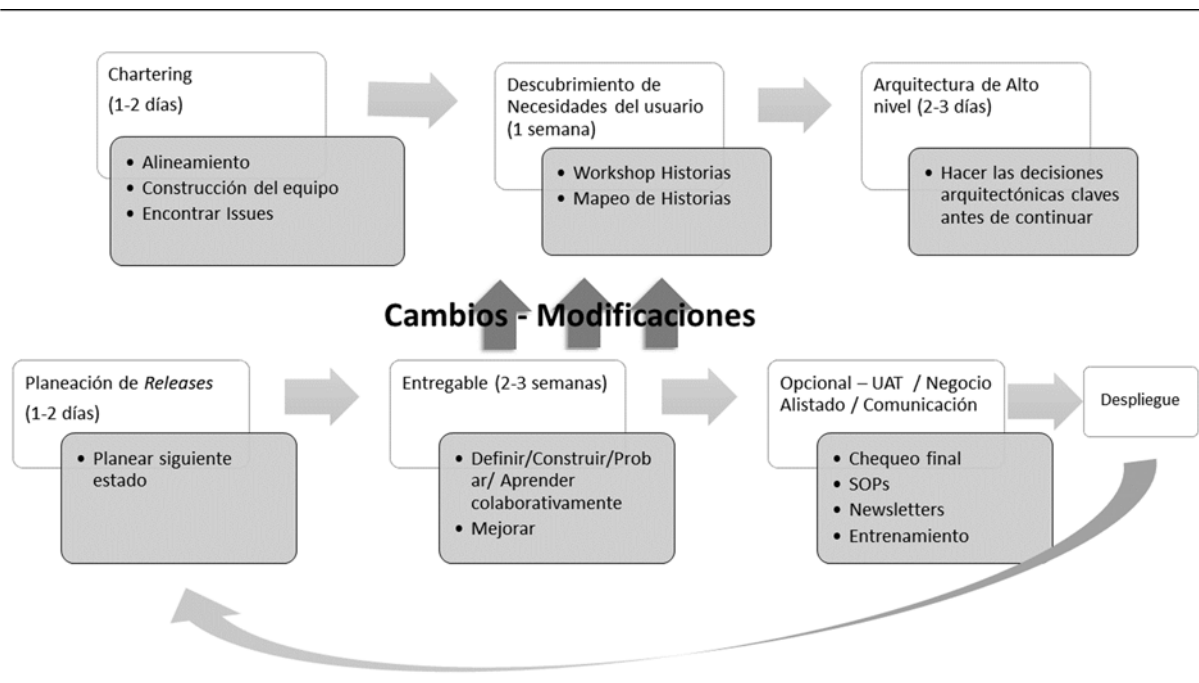


Figura 1 - Diagrama de procesos con filosofía Agile

El proceso esquematizado muestra en la parte superior las actividades de definición de Historia y requerimientos, así como del diseño. Este conjunto de actividades se realizará al comienzo del proyecto y se alimentarán en la medida que existen cambios, modificaciones o mejoras.

En la parte inferior se bosquejan las actividades de construcción las cuales son cíclicas de acuerdo con las varias iteraciones que se definan y planeen, las cuales en lenguaje Scrum se referirán a Sprints y Releases.

A continuación, se describen cada uno de los subprocesos y/o actividades. El de Chartering, Descubrimiento de Necesidades de Usuario y Arquitectura de Alto Nivel en esta sección mientras que, las demás se describirán en la próxima sección de Scrum.

3.1.5.1. Chartering

El Project Charter es un documento avalado por la metodología PMI cuyo propósito es alinear a todos los participantes, desarrolladores e interesados, en los objetivos generales del proyecto. Como lo establece el PMBOK V6[]: "La carta del proyecto establece una asociación entre las organizaciones de desarrollo y la solicitante".

1. Así mismo en este documento se establecen aspectos como:
2. El origen del caso del negocio,
3. Los acuerdos entre las partes,
4. Factores ambientales,
5. Requerimientos de alto nivel,
6. Descripción del proyecto, fronteras y entregables claves,
7. Riesgo general del Proyecto,
8. Cronograma general resumido,
9. Recursos financieros pre-aprobados,
10. Lista de Stakeholders claves,
11. Requerimientos de aprobación del Proyecto,

12. Project Manager asignado, responsabilidad y Nivel de Autoridad y
13. Nombre y Autoridad del patrocinador u otra persona autorizando el proyecto

3.1.5.2. Descubrimiento de las necesidades de Usuario

El propósito principal de estas actividades es el de conseguir identificar los requerimientos de los usuarios. La herramienta principal de la identificación y definición de requerimientos se da alrededor de la **Conversación**, calificada esta como **efectiva** y **frecuente**. Esta será principalmente cara-a-cara y tendrá preponderancia sobre otra forma de comunicación: papel, e-mail, mensajes, entre otros. Aun cuando se recomienda documentar las diferentes conversaciones a través de Grabaciones y Ayudas de memoria.

Al ser este proceso de tipo Adaptivo, se basa en el Descubrimiento, en lugar de la Recolección, de las necesidades del usuario. El Descubrimiento se da a través de conversación frecuente y efectiva alrededor de componentes visuales del prototipo del producto que ilustren suficientemente sobre las necesidades a Elicitar o de las implementaciones de estas necesidades a validar.

Documento de requerimientos

A diferencia del modelo de cascada, el levantamiento de requerimientos no gira en torno a un documento de requerimientos congelado, sino que este, más bien, es un producto vivo y evolucionable que registra los hallazgos del proceso iterativo de descubrimiento.

Los requerimientos identificados sufren un refinamiento progresivo a través del proceso de Descubrimiento.

CCC

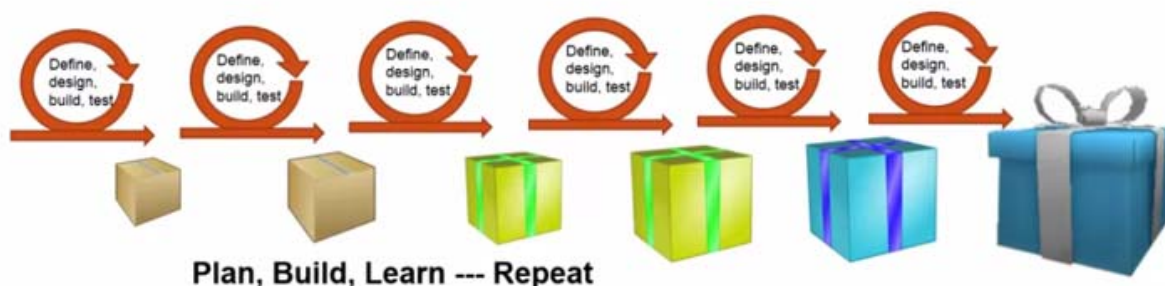
Para el Descubrimiento se utilizará la técnica conocida como CCC (Carta-Conversación-Confirmación), la cual consisten en la elaboración de Historias de Usuario, a través de las cuales se ilustran, describen y caracterizan las necesidades de los usuarios. Como primera medida, cada Historia de Usuario (HU) se registrará en una Tarjeta. Posteriormente, se sostendrán conversaciones entre los miembros del equipo, incluyendo desarrolladores e interesados, donde los demás miembros del equipo pueden aportar, refinar o mejorar el contenido de esta. Finalmente, se realizará la confirmación de la HU a través de su confirmación por el PO y el SM y se hará una retrospectiva de la misma con el ánimo de ser mejorada. Dicha retrospectiva podrá ser ejecutada dentro de la construcción de la HFV. Una vez que cada HU es confirmada, la misma será insumo para el Documento de Requerimientos.

El refinamiento progresivo se realizará para cada una de las cartas (tarjetas) a través del proceso iterativo Definir-Diseñar, Construir y Probar. Al final se obtienen no solo historias refinadas, sino prototipos (o productos) incrementales.

En cada HU se deberán identificar los usuarios, crear las personas y cada participante construirá las historias del tema seccionado. Para esto existentes diferentes aproximaciones para el desarrollo:

- De arriba hacia abajo
- De abajo hacia arriba

- Formato libre.



Criterios de Aceptación

Un elemento muy importante es que para cada HU se defina(n) su(s) criterio(s) de aceptación. Esto se hará básicamente resolviendo los siguientes interrogantes:

1. Una vez que una HU es terminada, ¿qué validará el cliente para verificar que dicha HU está culminada?
2. ¿Qué probaremos para confirmar que dicha HU está terminada?
3. ¿Cómo mostraremos el software correspondiente en las revisiones de producto?

Dentro del desarrollo de la HFV, los criterios de aceptación serán definidos a través de Scripts de Pruebas correspondientes. Dichos scripts formarán parte de la Planeación y la Ejecución del Plan de Pruebas.

Se recomienda que los criterios de aceptación sean realizados con la participación del PO y el equipo de desarrollo, incluyendo los miembros del equipo dedicado a pruebas.

Características e HUs

Tamaño

Las HU pueden ser a nivel de una funcionalidad del negocio, del usuario o del desarrollo. En el primer caso si una HU agrupa varias HUs, describiendo una Función del negocio, la llamaremos épica.

En el segundo caso, cuando la historia es más pequeña y específica al usuario, la llamaremos Historia de usuario. En el tercer y último caso, una HU puede ser abierta en otras Historias de Desarrollo para la construcción del producto.

Por objetivo

Hay HUs que no cumplen una función necesitada por el usuario, pero son desarrolladas para investigar o explorar características técnicas de herramientas de desarrollo, arquitectónicas u otro tipo de exploración técnica. Estas tienen unos claros criterios de aceptación y están muy limitadas en el tiempo.

Existirán también HU asociadas requerimientos no-funcionales, como, por ejemplo: Diseño arquitectónico de la Base de Datos, Implementación de características de seguridad o de desempeño, entre otras.

Bien construida

Para que una HU sea considerada bien construida deberá tener las siguientes características:

- Independiente,
- Negociable,
- Valorable,
- Estimable,
- Pequeña y
- Verificable.

Técnicas para generación de las HU

Se usarán básicamente Workshops de HU y Mapas de Historias.

Workshops de HU

Se realizarán reuniones (en su mayoría) remotas, preferiblemente cara-a-cara aun cuando por problemas de desempeño de red se puede omitir video, donde se realizarán conversaciones para registro en las tarjetas de la HUs, discusión, refinamiento y mejora de éstas., así como de su trabajo de retrospectiva.

Las conversaciones serán grabadas y se realizarán ayudas de memoria sobre dichas grabaciones con los elementos principales de las conversaciones y compromisos. Como salida importante se tendrán tarjetas actualizadas.

A las reuniones deben asistir el PO, el SM y los miembros del equipo de desarrollo necesarios.

El objetivo fundamental es trabajar tantas HUs como sea posible.

La duración estará entre 1 o varias horas.

Mapas de Historias

Como ayuda de las conversaciones y para tener una visión de alto nivel del conjunto de HUs se realizarán tableros con mapas de HUs, donde a nivel alto se especifiquen los diferentes *releases*, y los elementos que los componen. Dichos elementos corresponderán a conjuntos de HUs.

Otros objetivos de los Mapas de Historias son:

- Organizar y priorizar los *Backlogs*,
- Comprender y comunicar las necesidades de usuarios y
- Planear y Desarrollar *Releases*.

Adicionalmente, los Mapas de Historias también servirán para:

- Comprender piezas faltantes del descubrimiento de necesidades (HUs).
- Ayudar en la comunicación a diferentes niveles.
- Ayudar a contar una historia.

3.1.5.3. Arquitectura de Alto nivel

Una vez se tenga un conjunto de HUs suficiente, se diseñará el sistema. Dicho diseño incluirá el diseño arquitectónico de alto nivel en lo que tiene que ver con los componentes de software y el diseño del modelo de Datos (La base de Datos).

El documento de diseño incluirá otros aspectos fundamentales como el diccionario de datos, componentes a implementar, requerimientos de red, hardware y software, así como requerimientos de desempeño y seguridad, entre otros.

3.2 Marco de Referencia Scrum

Scrum es definido como un marco de referencia para el desarrollo de proyectos basados en los principios de Agile. Dicho de otra forma, Scrum[2] es un conjunto métodos ágiles en el sentido que ellos soportan la filosofía Agile.

Para desarrollar un método de Scrum técnico [6] es indispensable tener los siguientes elementos:

- **Roles**
 - Dueño de producto (PO)
 - Equipo de desarrollo (DT)
 - Scrum Master (SM)
- **Eventos**
 - El Sprint
 - Reunión de planificación
 - Scrum diario
 - Revisión de sprint
 - Retrospectiva de sprint
- **Artefactos**
 - Pila de producto
 - Pila de sprint

Para los roles del proyecto de desarrollo de la HFV, el PO será nombrado del grupo de Investigadores o Interesados de alguna de las Universidades que conforman la alianza, mientras el SM será el Lider de Proyecto externo para el desarrollo de la HFV. El DT será conformado por la empresa contratada para el desarrollo de la HFV.

Los eventos de Scrum se complementarán con los Workshops de Descubrimiento de necesidades como reuniones iniciales.

Una vez identificados las HUs del primer *release* y construida la primera versión del Mapa de historias, se estructurará una pila (backlog) con las historias definidas y se efectuará un estimativo en tiempo para el desarrollo de cada una de ellas.

3.2.1 Estimación de Historias de Usuario

Dentro de las múltiples técnicas de estimación para el desarrollo de HUs, se usarán en este proyecto la técnica de **días ideales**, donde la mínima unidad de tiempo por HU se define en días. Se tendrán en cuenta las características de Esfuerzo vs. Duración, Estimación vs. Precisión y Relativo vs. Absoluto. Se establecerán contenedores de historias por tamaños de *buckets* de acuerdo con la serie de Fibonnaci (1, 2, 3, 5, 8, ...) buscando mayor precisión.

La estimación será realizada por el grupo de desarrollo, en menos de un día y utilizando la técnica de *planning poker* (cada miembro escribe su estimativo en una carta y después de analiza cada historia y se decide en equipo el estimado final,

después de conversaciones de concertación).

Una vez realizadas todas las estimaciones, se priorizan las mismas de acuerdo con las necesidades y se apilan de acuerdo con la necesidad para estimar el primer Sprint.

Complementariamente usaremos un tipo de diagrama Gantt, en Excel o MS Project, donde se registrarán las HUs, y se calendarizarán usando precedencias de las mismas.

Las Historias de Usuario, junto con sus atributos serán registradas en la herramienta Trello©.

3.2.2 Pila de Producto (Backlog)

Una vez estimadas y priorizadas las historias, se llevarán al primer *backlog* del producto (correspondiente al 1er. *release*). La pila de Producto deberá contener TODAS las Historias de Usuario, las cuales reflejan las necesidades del Usuario para constituir el producto.

De acuerdo con la duración y priorización se planeará cuales historias podrán ser desarrolladas en el primer Sprint del producto (Idealmente de 2 a 3 semanas). Con el numero de Historias remanentes se efectuará este mismo proceso para un 2do Sprint y así sucesivamente hasta agotar el Backlog y completar y entregar el primer *release*.

3.2.3 Release

Cada release se determinará con base en tiempo o alcance. En nuestro primer reléase, este será determinado con base en el alcance, el cual estará constituido por una Línea Base que contenga:

1. Diseño arquitectónico y Modelo de Datos de la HFV.
2. Implementación del ambiente de desarrollo.
3. Escogencia de herramientas y plataformas.
4. ETL desarrollado en su parte de Carga a la herramienta, partiendo de archivo .csv de interfaz al CMS.
5. Desarrollo e implementación del almacenamiento de datos granulares de métricas y su cálculo.
6. Consulta, tipo tabla, de métricas en varias dimensiones: Estudiante, curso, programa, métrica y/o periodicidad y combinación de ellas.
7. Gráfica de algunas métricas.

Los *releases* serán descritos visualmente dentro de un mapa de historias realizado en CardBoard©, sin perjuicio de que la lista también sea llevada en Excel (y/o en un diagrama tipo Gantt).

3.2.4 Pila del Sprint

En la planificación del Sprint, se constituye la lista de Tareas de Usuario que se realizarán en ese correspondiente Sprint. En cada una se indicará el esfuerzo (en días) para realizarla. La descomposición de Tareas deberá permitir una granularidad suficiente para poder efectuar monitoreos a diario de los avances y estado de cada una de las taras por los diferentes miembros del Equipo de Desarrollo.

3.2.5 Planeación del Sprint

Con el Backlog de Sprint y sus correspondientes tareas estimadas y priorizadas se efectúa la **Planeación del Sprint**. En este último evento se determinan los objetivos del Sprint y las tareas para alcanzarlos.

Se planeará la duración de este para estar entre 2/3 semanas.

Se trabajará con un tablero de control que tenga la lista del backlog inicial del Sprint, una lista de tareas por hacer, otra de tareas en desarrollo y otras de tareas hechas.

Se utilizará la herramienta Trello© para el registro de Historias de Usuarios y para la planeación y seguimiento de las listas antes mencionadas.

3.2.6 Scrum Diario

Es una reunión diaria de duración muy breve entre los miembros del equipo. En esta reunión se tratan cuestiones como:

Qué se hizo ayer,

Qué se hará hoy,

Cosas que se requieren o se pueden requerir, o impedimentos u obstáculos.

Independientemente de la reunión de Scrum, en lo posible se deben propiciar mecanismos de comunicación que ayuden a resolver obstáculos, fallas o dudas en el menor tiempo posible. Incluso sin esperar a la próxima reunión de Scrum para subsanarlos.

3.2.7 Revisión del sprint

Esta actividad comprenderá el análisis e inspección de las HUs, posibles cambios de ellas, de los productos generados o en proceso del **release** correspondiente, y adaptación de la pila del producto si resulta necesario.

3.2.8 Retrospectiva del Sprint

Esta tarea se realiza al finalizar el Sprint. Esta comprende la revisión de los aspectos de lo sucedido dentro del Sprint, registro en una lista de las lecciones aprendidas a tener en cuenta en el próximo Sprint o en un proyecto futuro.

Se creará un plan de mejoras para el siguiente Sprint.

3.3 Anexos