

Experiencia en la construcción de la Metodología Ágil de Desarrollo Colaborativo de Software

Ing. Kleivymar Montilla¹

Ing. Mariangel Molero¹

MSc. Julie C. Vera^{1,2}

Centro Nacional de Desarrollo e Investigación
en Tecnologías Libres - CENDITEL. Mérida, Venezuela.¹

Universidad de los Andes. Mérida, Venezuela.²

kmontilla@cenditel.gob.ve

mmolero@cenditel.gob.ve

jvera@cenditel.gob.ve

Fecha de recepción: 02/10/2019

Fecha de aceptación: 02/11/2019

Pág: 138 – 146

Resumen

El área de la ingeniería de software ha evolucionado de forma exponencial, creándose nuevas industrias y productos tecnológicos, que ameritan la reformulación de los estándares de calidad y metodologías de trabajo. La Fundación Centro Nacional de Desarrollo e Investigación en Tecnologías Libres (CENDITEL) diseñó en el año 2007, la Metodología para el Desarrollo Colaborativo de Software Libre (MDCSL), donde se describen los procesos y actores que participan en la construcción de un software. No obstante, debido a la evolución exponencial que se ha presentado en el área de la ingeniería de software, así como la irrupción de las metodologías ágiles y las nuevas necesidades tecnológicas de las organizaciones, fue necesario iniciar un proceso de revisión y construcción de la versión 3.0 de la MDCSL, partiendo de la evaluación de los procesos que se llevan a cabo para el desarrollo del Sistema Nacional Integrado de Información Forestal.

Palabras Clave: Conceptualización, Desarrollo, Metodologías ágiles, Software.

Resumen

The area of software engineering has evolved in a way exponential, creating new industries and technological products, which merit the reformulation of quality standards and work methodologies. The National Center for Development and Research in Free Technologies (CENDITEL) designed in 2007, the Methodology for Collaborative Development of Free Software (MDCSL), which describes the processes and actors involved in the construction of software. However, due to the

exponential evolution that has occurred in the area of software engineering, the emergence of agile methodologies and the new technological needs of organizations, it was necessary to begin a process of revision and construction of version 3.0 of the MDCSL, based on the evaluation of the processes that are carried out for the development of the National Integrated Forest Information System.

keywords: Conceptualization, Development, Agile Methodologies, Software.

Introducción

En los últimos años, la Tecnología de Información (TI), ha sido incorporada de forma masiva en diferentes áreas de nuestra vida, siendo una pieza clave para el funcionamiento eficiente de las organizaciones públicas y privadas. Uno de los elementos que conforman el ecosistema de las TI es el software, una herramienta informática dirigida a gestionar, levantar, procesar o almacenar datos e información.

Es preciso señalar además que la ingeniería de software, ha evolucionado en cuanto al desarrollo de métodos, técnicas que sustenten el diseño, construcción y documentación de los programas informáticos, creando elementos y metodologías dirigidas a garantizar que cada uno de los procesos se lleven a cabo de manera eficiente, bajo ciertos estándares de calidad que aseguren su sostenibilidad, escalabilidad y apropiación. Las metodologías de desarrollo de software se han ajustado a las necesidades en cuanto al tiempo y tipo de aplicación, pasando de metodologías estáticas -paso a paso-, a metodologías ágiles, parte de ellas enfocadas al levantamiento de requerimientos y a la codificación de las funcionalidades, restando importancia a la conceptualización, documentación y administración del software.

Dentro de la ingeniería del software, se han desarrollado otros elementos que están dirigidos a completar la forma en cómo se construye un programa informático, siendo la calidad un componente importante y transversal, que determinará el cumplimiento de los requerimientos descritos por los usuarios, los objetivos y metas planteadas, para lo cual es necesario que el proceso de desarrollo de software se encuentren en constante evaluación por parte de un analista de calidad (QA)¹, un profesional que se encarga de monitorear, revisar y verificar que cada uno de los procedimientos se realicen de la mejor manera, una evaluación que se sustenta además en la metodología de desarrollo seleccionada.

Es bajo este contexto, que el Centro de Desarrollo e Investigación en Tecnologías Libres (CENDITEL), crea la línea de investigación *Aseguramiento de Calidad del Software* en el año 2007, esto con el objetivo de estudiar cada uno de los procesos involucrados en el desarrollo de software, publicando en el año 2015 la segunda versión de la Metodología para el Desarrollo Colaborativo de Software Libre (MDCSL), la cual surge como una revisión para mejorar los procesos de la primera versión publicada en el 2007. Esta segunda versión está orientada a mejorar el proceso de desarrollo de software libre con base a buenas prácticas de desarrollo, en las cuales se hace énfasis en la documentación, a fin de mejorar la calidad del mismo,

¹Se utiliza el anglicismo QA (Quality Assurance)

prestando atención en la mantenibilidad y portabilidad de proyectos de software libre. Una metodología conformada por tres procesos: la conceptualización, administración y construcción de aplicaciones de software libre; describiéndolos de forma ordenada, pero tomando en cuenta que es un ciclo interactivo, donde las actividades de cada proceso pueden ser ejecutadas de forma recursiva.

No obstante la introducción de las TI, de nuevas técnicas, tecnologías y el crecimiento exponencial de datos, ha creado la necesidad de estructurar metodologías cada vez más ágiles que permitan desarrollar de forma rápida las aplicaciones, manteniendo los mismos estándares de calidad y optimizando el uso de los recursos (horas hombres, tiempo, equipos). Además, las organizaciones, demandan software cada vez más especializados, que incluyan la posibilidad de realizar grandes operaciones de computo, procesando grandes cantidades de datos en tiempo real, generando visualizaciones intuitivas para el usuario. Como es el caso, del Sistema Nacional Integrado de Información Forestal (SINIIF), un gestor de datos que está desarrollando CENDITEL, como parte del proyecto “Ordenación Forestal Sustentable y Conservación de Bosques en la Perspectiva EcoSocial” financiado por la Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO).

Las características del SINIIF, han permitido al equipo de aseguramiento de calidad, utilizarlo como caso de estudio para evaluar la MDCSL versión 2.0, específicamente el proceso de conceptualización del software, por medio de entrevistas y reuniones de trabajo con la analista de calidad (agente externo), un proceso que ha permitido iniciar la construcción de la versión 3.0 de la MDSL, que tiene como premisa la flexibilidad, interacción, recursividad y transversabilidad de sus procesos.

En esta experiencia, se pretende en primer lugar, colocar en contexto a nuestro lector, acerca de la MDCSL versión, 2.0., para luego exponer los resultados de las reuniones de trabajo realizadas con la analista de calidad, presentando una primera aproximación de procesos que serán descritos en la nueva versión de la metodología.

Contexto

Los avances tecnológicos se están generando constantemente y de forma acelerada, dirigidos a mejorar la eficiencia en el uso de los recursos (tiempo, espacio, humano, capital, financiero). Específicamente los software, como productos tecnológicos, se han convertido en una necesidad en los distintos ámbitos como el educativo, gerencial, comunicacional y económico, debido a que permiten optimizar procesos, levantar, procesar y gestionar grandes cantidades de datos en tiempo real; convirtiéndose en una pieza clave para la inserción y sostenibilidad de las organizaciones en un entorno cada vez más competitivo y globalizado.

Esta evolución tecnológica ha generado un alto crecimiento en la industria del software a nivel mundial, que como todo producto, cuenta con una serie de requerimientos, insumos y estándares de trabajo para su fabricación. A estos estándares se le denominan metodologías de desarrollo, un conjunto de procedimientos que determinan la forma en cómo se desarrolla un software, y que además se rigen por los principios de la ingeniería del software descritos

por Pressman, R. y Troya, J. (2010) [4]: rigurosidad, modularidad, abstracción, anticipación de cambios y generalidad.

A partir de la creciente demanda de software han surgido diversas metodologías que se ajustan a los objetivos y necesidades del tipo de programa informático a construir, encontrando metodologías de cascada, espiral, de prototipos, de programación externa y ágiles. Para desarrollar software de calidad, que se ajusten a los requerimientos de los usuarios, se necesita hacer uso de metodologías de desarrollo que guíen, controlen y aseguren el cumplimiento de ciertos estándares de calidad del software, puesto que éstas ofrecen patrones estandarizados de construcción conformados a su vez por etapas, fases o procesos, que se descomponen en actividades y tareas sujetas a productos y roles.

Es bajo este contexto, que en CENDITEL se crea en el año 2007 la línea de investigación Aseguramiento de Calidad del Software, la cual se ha centrado en diseñar formas de trabajo para el desarrollo de aplicaciones bajo el enfoque de software libre, fundamentada en la aplicación de prácticas de documentación que garanticen la apropiación de lo software desarrollados, elemento que asegura el cumplimiento de las libertades del software libre.

Durante el primer año, se elaboró la propuesta de un modelo funcional para la construcción de la metodología de desarrollo, presentando diagramas y formatos a emplear en dicho modelado. Luego en el 2011 se logran realizar dos avances, que plantean un modelo de aseguramiento de calidad en el desarrollo de software libre, donde se describen las métricas de evaluación de la calidad del software libre, y la estructura y roles de los miembros del equipo que llevarán a cabo el desarrollo de un proyecto de software.

Posteriormente en el 2012 se publica la primera versión estable la MDCSL. Para el 2013 se proponen una serie de mejoras a los procesos que componen la Metodología de Desarrollo de Software Libre, esto en base a recomendaciones indicadas por desarrolladores y a la experiencia en el proceso de validación del modelo de Aseguramiento de Calidad abordado en CENDITEL.

Seguidamente en el año 2014 se elabora una segunda versión de la metodología que contempla nuevas técnicas de gestión, la simplificación de flujos de trabajo, la incorporación de actividades de empaquetado y gestión de versiones, y la actualización de la lista de herramientas informáticas.

La segunda versión de la MDCSL fue publicada en el año 2015, por los investigadores: Johanna Alvarez y Victor Bravo, miembros de la Línea de Investigación Aseguramiento de Calidad de Software, esta metodología tiene como objetivo promover un proceso de orientación, aprendizaje y enseñanza en torno a la práctica virtuosa de desarrollo de software, hecho que permitiría garantizar la calidad de las aplicaciones que se generan en todas y cada una de sus fases de construcción.

La MDCSL está sustentada en la visión de práctica expuesta por MacIntyre, citado por Álvarez, Bravo y Villasana (2014)[1], explicada como “[...] una actividad humana cooperativa, socialmente establecida, en la que se realizan los bienes inmanentes a la actividad misma, según reglas y pautas de excelencia que definen la propia actividad.” (p.9).

La metodología MDCSL, presta mayor atención a la documentación del software y al proceso de construcción (especificación de casos de uso y codificación), a fin de facilitar la apropiación

de los procesos ejecutados, facilitando así su usabilidad, escalabilidad, mantenimiento y la mejora del software por parte de los usuarios. La MDCSL está formada por 3 procesos: Conceptualización de Proyectos de Software Libre, Administración de Proyectos de Software Libre y Construcción de Aplicaciones de Software Libre; integrados y relacionados entre sí, incorporando además el uso de herramientas tecnológicas para la documentación del desarrollo, específicamente la plataforma *Trac*, un gestor que por medio del plugin de MDCSL diseñado, sirve como una guía para cada una de las actividades que conforman los tres procesos. (Álvarez y Bravo, 2015.) [3]

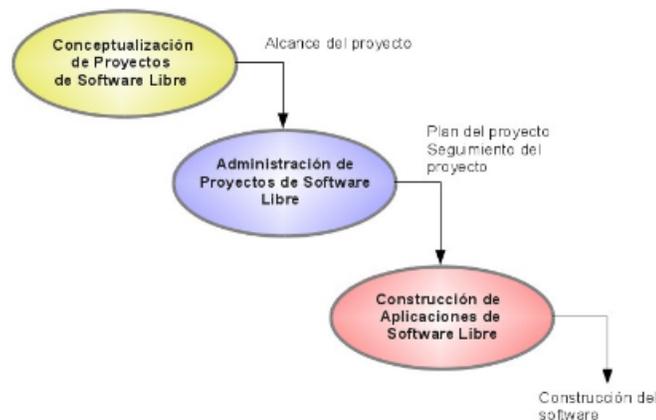


Figura 1: Procesos que conforman la metodología de desarrollo colaborativo de software libre.

Fuente: tomado de Álvarez y Bravo, ob. cit., p.6. [3]

Álvarez y Bravo, describen cada uno de los procesos de la siguiente manera:

1. Conceptualización: es un proceso conformado por una serie de actividades dirigidas a levantar la información necesaria para modelar luego los procesos a automatizar en la aplicación. Su objetivo es: “[...]comprender el dominio de la aplicación a desarrollar así como los problemas o necesidades de los usuarios en relación a dichos procesos” (Álvarez y Bravo, ob. cit., p. 3.)[3].
2. Administración: este proceso comprende actividades asociada a la planificación de las tareas, definiendo el como se realizara la evaluación y seguimiento de las tareas.(ídem, p. 16.)[3].
3. Construcción: “Este conjunto de actividades se agrupan en las siguientes fases: Especificación de Requerimientos, Análisis y Diseño, Codificación, Pruebas y Liberación.” (ídem, p. 28.) [3].

Cada uno de los proyectos de desarrollo ejecutados desde el año 2016 dentro de CENDITEL, se han construido bajo la versión 2.0 de la MDCSL, proyectos que han surgido desde la misma

organización; en el caso del SINIIF, contamos con un usuario que tiene claro sus necesidades y requerimientos, buscan un software para recopilar, sistematizar y procesar información existente en bases de datos asociados a los recursos forestales desde diversas perspectivas, automatizando la generación de reportes, mapas temáticos e indicadores que faciliten la evaluación y el monitoreo de los recursos forestales y su manejo. Un software enmarcado en el proyecto “Ordenación Forestal Sustentable y Conservación de Bosques en la Perspectiva EcoSocial”.

Dentro de este marco, el equipo de aseguramiento de calidad pretende evaluar el desempeño de la MDCSL implementándolo en el desarrollo del SINIIF, centrando la atención en la primera fase de la metodología la conceptualización del proyecto de software. Para ello, la analista de calidad, evaluó los elementos que conforman la propuesta de desarrollo, así como la especificación de requerimientos del SINIIF. Los resultados fueron presentados y debatidos por el grupo de trabajo, arrojando la necesidad de realizar una actualización a la MDCSL, para así incorporar elementos ágiles y redefinir el primer proceso, con el objetivo de crear una base sólida -concepto del software-, antes de iniciar con los otros procesos. En la siguiente sección, se presentan los resultados obtenidos, así como las propuestas planteadas.

Proceso de Construcción de MDCSL versión 3.0

El equipo de aseguramiento de calidad, inició el proceso de actualización de la MDCSL, en función de generar cambios que permitan contar con una estructura metodológica ágil y flexible, adaptándose a los constantes cambios que se generan en el ámbito de la ingeniería de software. Para ello, se realizó en una primera etapa, una revisión de la versión 2.0 de la metodología, de sus procesos generales como conceptualización, administración y construcción, centrando la atención en el primer proceso.

De acuerdo a la Figura 1, los procesos que conforman la versión 2 de la metodología son secuenciales, es decir, que debe culminar uno para iniciar otro, las actividades son interactivas, existe retroalimentación entre ellas. Sin embargo, la visión secuencial de los procesos, incrementa el riesgo de retrasos en la gestión del proyecto, en especial en el proceso de construcción de software. De forma análoga, se debe enaltecer la estructura sistemática que brinda ésta metodología; puesto que está constituida por procesos que se desglosan en grandes actividades y fases, y a su vez éstas se descomponen en tareas, indicando para cada una de las tareas las recomendaciones para realizarlas, los productos a obtener como salida, así como también los responsables en cada una de éstas. Sin duda alguna que el marco metodológico planteado en dicha metodología orienta detalladamente el ciclo de desarrollo de un software.

Por otro lado, la MDCSL, esta fundamentado filosóficamente en las prácticas virtuosas y la visión de construcción social, de trabajo colaborativo, como se reseña en Alvarez, et al., ob. cit.[1]. Un elemento clave que hace falta describir de forma en explícita es cómo se genera el trabajo colaborativo dentro de la gestión de proyectos de software.

En esta primera revisión de la versión 2.0 de la metodología, realizada se encontraron los siguientes elementos:

- Define de forma clara las reglas, actividades, tareas y roles, así como los productos

generados al final de cada actividad.

- Para el desarrollo de las actividades, la MDCSL presenta una serie de herramientas tecnológicas libres, que sirven de apoyo para la realización de las tareas.
- Respecto al enfoque colaborativo, en los documentos que sustentan la MDCSL, se establece como premisa que la calidad del software dependiera en gran medida al trabajo colaborativo. Un enfoque, derivado de la visión del Proyecto GNU/Linux, que da sustento y sentido social a los proyectos de software. (Alvárez, et al., ob. cit.[1]). No obstante, esta visión no se ve reflejada claramente en la MDCSL, de allí que sea necesario dejar en evidencia este enfoque en cada uno de los procesos de la metodología, estableciendo los aspectos y formas de trabajo que garantizarían desarrollar el proyecto de manera colaborativa.

Para la conceptualización debe considerarse la base de cualquier proyecto de software, de cada uno de los procesos involucrados y además qué determina el ¿qué es?, ¿para qué?. De allí, que sea necesario reestructurar la visión inicial de los tres procesos planteados en la versión 2.0 de la MDCSL de la siguiente forma:



Figura 2: Reformulación de los procesos.
Fuente: Elaboración propia

- La conceptualización, será el centro del proyecto de software, en el se delimitará el concepto del producto tecnológico a desarrollar, ¿qué es?. Partiendo de este punto, se ejecutarán los otros procesos, de forma paralela, interactiva y con ciclos de retroalimentación.

- **Administración:** involucra actividades asociadas al plan del proyecto, seguimiento de las actividades y tareas asignadas a cada uno de los integrantes. Así como a la instalación, configuración y gestión de los servidores de prueba y producción en caso que apliquen.
- **Diseño:** este proceso dependerá directamente del levantamiento de requerimientos realizado en la conceptualización, definiendo los casos de uso de acuerdo al levantamiento inicial, la forma en como se especificarán las funcionalidades (flujogramas, diagramas de casos de uso, prototipo no funcional comentado, entre otros) dependerá del tipo de proyecto.
- **Construcción:** involucra el proceso de codificación de las funcionalidades, documentación, de acuerdo a las especificaciones descritas.
- **Evaluación:** realización de pruebas funcionales y no funcionales, evaluar los requerimientos de los servidores de acuerdo a las pruebas de estrés: así como la calidad del software, durante todo el proceso de desarrollo.

Parte de la nueva propuesta, se centra en la conceptualización, definiendo algunos elementos claves que debe contener la nueva MDCSL y que surgieron luego de la evaluación del proceso de conceptualización realizado al SINIIF, por parte de la analista de calidad:

- **Delimitar los conceptos:** los proyectos de desarrollo de software pueden estar enmarcados en diversos ámbitos multidisciplinarios, de allí que toda propuesta de desarrollo debe contar con un glosario de términos que permita unificar criterios entre el usuario y todos los miembros del equipo de trabajo.
- **Usar instrumentos para limitar los requerimientos.**
- **Definir los procesos de validación de la propuesta de desarrollo, interfaz gráfica, requerimientos y la especificación de requerimientos con los usuarios, para disminuir riesgos de retrasos o inconformidad por parte del usuario.**
- **Incorporar en el equipo de trabajo asesores especialistas en el área, por ejemplo para el caso del SINIIF, es necesario incorporar un profesional en el área forestal.**
- **Realizar reuniones -cortas- semanales, entre los analistas funcionales y los desarrolladores, para describir el diseño.**

Consideraciones finales

La MDCSL versión 2.0, cuenta con un sólido sustento teórico para la gestión de proyectos de software colaborativo, fomentando buenas prácticas de documentación. No obstante, es necesario actualizar la metodología, para especificar la transversalidad de ciertos procesos (administración, diseño, construcción y evaluación), así como la retroalimentación entre ellos. Y

establecer a la conceptualización como la base de cualquier proyecto de desarrollo de software, debido a que de este dependerá el entendimiento y comprensión de lo que el usuario desea y necesita, disminuyendo riesgos y pérdida de tiempo en la ejecución del resto de los procesos.

Agradecimiento

A la ingeniera Diony Alviárez, profesora de la Universidad Politécnica Territorial Klever Ramirez - Mérida (UPTM) que ha sido pieza fundamental en el proceso de construcción de evaluación y monitoreo del proceso de conceptualización del SINIIF.

Bibliografía

- [1] Alvarez, J., Bravo, V. y Villasana, D. (2014). Esbozo de conceptualizaciones del primer documento del Estudio de prácticas virtuosas en el área de desarrollo de software libre desde un enfoque filosófico. (Documento en Línea). Centro Nacional de Desarrollo e Investigación en Tecnologías Libres. Disponible en: http://calidad-sl.cenditel.gob.ve/files/2011/06/conceptualizaciones_PracticasVirtuosas_1702141.pdf
- [2] Alvarez, J. y Solé, S. (2015). Experiencia en Aseguramiento de Calidad en el ámbito del Software Libre. *Revista Electrónica Conocimiento Libre y Licenciamiento CLIC*. Nro. 8, Año 5. Disponible: http://convite.cenditel.gob.ve/files/2015/01/ExperienciaRevistaCLIC_n8_1.pdf
- [3] Alvarez J. y Bravo, V. (2015). *Metodología para el Desarrollo Colaborativo de Software Libre Versión 2.*. CENDITEL. Ministerio del Poder Popular para la Educación Universitaria, Ciencia y Tecnología.
- [4] Pressman, R., y Troya, J. (2010). *Ingeniería del software*. McGraw Hill