

MODELO Y PRÁCTICAS ESENCIALES DE LA METODOLOGÍA DAC INTEGRANDO LOS MÉTODOS ÁGILES, PMBOK Y CMMI-DEV

MODEL AND PRACTICES OF DAC METHODOLOGY INTEGRATING THE ÁGILE METHODS, PMBOK AND CMMI-DEV

Ing. Aleli Sánchez Méndez

*Universidad de las Ciencias Informáticas, Dirección de Informatización, Líder de proyectos en el Departamento de Desarrollo. La Habana - Cuba
aleli@netcons.com.cu*

(Recibido el 04-07-2014. Aprobado el 07-11-2014)

Resumen. El presente artículo tiene como objetivo elaborar una metodología para el desarrollo de software de gestión basado en componentes, integrando las prácticas ágiles, estándares internacionales de gestión de proyectos y calidad de software y el modelo CMMI, para la mejora de las evaluaciones en los indicadores que miden la ejecución del proyecto, la satisfacción del cliente, la calidad del proceso y del producto en proyectos de software. Esta metodología se llama DAC, siglas de Desarrollo Ágil con Calidad. Surge ante la necesidad de algunas entidades desarrolladoras de software de adaptarse a los cambios, brindar software con calidad a sus clientes en períodos cortos de tiempo y reducir siempre que sea posible los gastos. DAC es un compendio de prácticas de PMBOK, CMMI-DEV, ISO/IEC 12207, XP, Scrum, FDD y el Manifiesto Ágil. Al aplicar DAC en una fábrica de software la misma puede brindar a sus clientes la seguridad de tener como premisa la calidad en el proceso de desarrollo. Esto garantiza por ende un nivel aceptable de calidad en los productos los cuales pueden entregarse en períodos cortos de tiempo de forma incremental aportando valor de negocio para el cliente. La metodología DAC está sustentada en una investigación de tipo explicativa donde se empleó, entre otros, el método experimental, realizando un pre experimento con pre y post prueba con un solo grupo. Ha sido aplicada en una entidad desarrolladora de software durante un año obteniendo resultados alentadores en cuanto a la ejecución del proyecto y resultados de evaluaciones de calidad a procesos y productos.

Palabras clave: calidad de software, CMMI, DAC, enfoque ágil, metodologías ágiles, metodología de desarrollo de software, proceso de desarrollo de software.

Abstract. This research aims to develop a methodology for the development of management software based on components, integrating agile practices, international standards of project management and software quality and the CMMI model for improving assessment indicators measuring project execution, customer satisfaction, quality of process and product in software projects. This methodology is called DAC, which stands for Quality Agile Development. This approach emerged as a response of some software development companies to adapt to changes, providing quality software to customers in short periods of time and reduce costs whenever possible. DAC is a compendium of practices from PMBOK, CMMI-DEV, ISO/IEC 12207, XP, Scrum, FDD and Agile Manifesto. Applying DAC in a software factory that can provide its customers the security of having quality premised on the development process. This therefore ensures an acceptable level of quality in products which can be delivered in short time periods incrementally, adding business value to the customer. The DAC methodology is supported by an investigation of explicative type where it was applied, among others, the experimental method by performing a pre experiment with pre and post test with a single group. It has been implemented in a software developer organization for one year with encouraging results in regard to compliance with production schedules, customer satisfaction and results of evaluations of process and product quality.

Keywords: agile approach, agile methodologies, CMMI, DAC, software development methodology, software development process, software quality.

1. INTRODUCCIÓN

En el año 2001 diecisiete especialistas, convocados por Kent Beck, definieron el término “Métodos Ágiles” para referirse a las metodologías que estaban surgiendo como alternativa a las tradicionales. Los integrantes de la reunión resumieron en el Manifiesto Ágil cuatro valores y doce principios sobre los que se basan los métodos alternativos (Beck et al., 2001).

Al consultar el Chaos Manifiesto (The Standish Group International, 2013) se puede constatar que uno de los factores de éxito de los proyectos en el 2012 fue acatar un proceso ágil. En el caso específico de los proyectos pequeños se identificaron en este reporte un total de 10 puntos de éxito para la implementación de un proceso ágil.

En los últimos años la adopción de los métodos ágiles para el desarrollo se ha incrementado. Según el World Quality Report (Reporte de Calidad Mundial) del 2013-2014 (HP, Ayer Sogeti, & Capgemini, 2013) el 83% de las empresas usan metodologías ágiles para el desarrollo de sus aplicaciones, ya que éstas les permiten adaptarse mejor a los cambios del mercado. El 46% de estas empresas no disponen, sin embargo, de técnicas de prueba consistentes para estas metodologías. El World Quality Report se basa en entrevistas a 1500 directores de TI de 25 países.

Por otro lado el Modelo de Integración de Capacidad y Madurez (CMMI, por sus siglas en inglés) en su revisión 1.3 (CMMI Product Team, 2010) es el patrón de la mejora de proceso en el dominio del desarrollo de software pues provee un enfoque sistemático y sistemático para la mejora de procesos. Según estadísticas de CMMI Institute, (2014) ha habido más de 9500 evaluaciones de 84 países hasta marzo del 2014. Entre estos países se encuentra Cuba con una certificación de CMMI para Desarrollo (CMMI-DEV), de la Universidad de las Ciencias Informáticas. A nivel mundial los países con mayor cantidad de evaluaciones CMMI son China, Estados Unidos, India, España, Japón, Corea del Sur, México, Brasil, Francia y Taiwán (CMMI Institute, 2014).

En cuanto a la gestión de proyectos el estándar por excelencia es el Cuerpo del Conocimiento para la Gestión de Proyectos (PMBOK, Project Management Body of Knowledge), Desarrollado por el Project Management Institute (PMI, 2013). El PMI actualiza la versión del PMBOK cada 4 años –(Meléndez De La Cruz, 2013).

En él se explican una serie de puntos o factores de éxito para los proyectos; entre estos destaca la correcta elección de los procesos de gestión de proyectos, así como el enfoque de desarrollo del software y por ende la metodología (PMI, 2013).

Con el fin de hacerse de un nombre y prestigio en la industria del software las empresas optan por modelos y estándares como CMMI y PMBOK para certificar la calidad de su proceso de desarrollo y gestión del software. Para esto necesitan mostrar un conjunto de evidencias, por lo general documentales. En este punto la empresa en ocasiones se ve ante la disyuntiva de escoger entre aplicar el modelo o estándar y aplicar métodos ágiles cuando en realidad todo es una cuestión de interpretación.

Equivocadamente se ven a los métodos ágiles como anti-documentos sin comprender realmente que las prácticas ágiles se hicieron para ser flexibles y adaptables a los cambios y las necesidades propias de cada proyecto. Por lo general los estándares y modelos te dicen qué tienes que hacer y posibles evidencias para demostrar que lo haces, pero no te dicen el cómo, ni te sujetan a utilizar formatos de documentos específicos para llevar registros de evidencias.

El crecimiento de la tecnología hace que los proyectos más modernos sean poco exitosos ya que los cambios tecnológicos se realizan con tanta rapidez que es muy difícil poder planificar a largo plazo. El cliente promedio de productos y soluciones de software de gestión exige entregas con frecuencias cortas debido a la necesidad de informatización de sus áreas de procesos. Al ser estos procesos cada vez más grandes y complejos el uso de un método tradicional para el desarrollo de software de gestión conllevaría un largo desarrollo y por ende una entrega tardía del software, a riesgo de que en el momento que este sea terminado los procesos hayan cambiado tanto que sea necesario modificarlo todo.

Apostar por los métodos ágiles favorece el éxito de los proyectos y la satisfacción de los clientes. Enriquecer estos métodos con modelos, estándares y normas de calidad constituye el broche de oro para la gestión de proyectos en el entorno actual. El agilismo y la calidad no tienen por qué utilizarse como términos opuestos. Una integración de ambos permitirá enaltecer la calidad manteniendo la agilidad en el desarrollo.

Este artículo tiene como objetivo presentar una metodología para el desarrollo de software de gestión, integrando las prácticas ágiles, estándares

internacionales de gestión de proyectos y calidad de software y el modelo CMMI, para la mejora de las evaluaciones en los indicadores que miden la ejecución del proyecto, la satisfacción del cliente así como la calidad del proceso y del producto en proyectos de software.

2. METODOLOGÍA DE INVESTIGACIÓN

Para lograr el objetivo planteado fue necesario seguir un conjunto de pasos, logrando en cada uno un objetivo parcial.

Paso 1: elaborar un marco teórico sobre los modelos de procesos para el desarrollo de software, modelos y estándares de calidad, modelos y estándares de gestión de proyectos, el enfoque ágil para el desarrollo, sus prácticas más usadas, sus metodologías más populares, el desarrollo basado en reutilización de componentes y la posible integración de estos elementos, así como herramientas de apoyo al proceso de desarrollo.

Paso 2: definir un modelo del proceso de desarrollo para software de gestión basado en componentes que integre métodos y prácticas ágiles de desarrollo, CMMI y los estándares de calidad y gestión de proyectos más adecuados.

Paso 3: definir las actividades, tareas, roles y productos de trabajo que conformarán una metodología de desarrollo de software de gestión basado en componentes que adopte el modelo definido así como proponer técnicas, prácticas y herramientas a utilizar en beneficio de un mejor resultado en la aplicación de la misma.

Paso 4: validar la metodología propuesta a través de los resultados obtenidos de su aplicación como un proceso definido para los proyectos de informatización de la Universidad de las Ciencias Informáticas (UCI) y el criterio de expertos.

La investigación es de tipo explicativa ya que se analizará el estado actual del problema identificando posibles causas, comportamientos frecuentes y formas de resolver el problema mediante una metodología o modelo. Para esto se aplicaron entre otros los métodos sistémico y experimental.

Se realizará un preexperimento con pre y post prueba con un solo grupo. La población total está constituida por los 6 proyectos de informatización en la UCI. Se selecciona el universo como muestra debido a que son pocos proyectos y todos proveen información necesaria para el experimento.

De la prueba se espera encontrar diferencias significativas entre los proyectos, demostrando que mejoró el grupo experimental luego de aplicado el experimento.

3. CONSTRUCCIÓN DE UN MODELO

3.1. Integrando Ágil, CMMI, ISO/IEC 12207 y PMBOK en DAC

En (2012) se trata de establecer algunas ideas que demuestran que “CMMI no tiene ningún requisito que impida a una organización conseguir una clasificación de CMMI mientras usa un proceso ágil”, además se ejemplifican casos reales en los que ha sido posible recibir una clasificación de CMMI aplicando métodos ágiles para el desarrollo.

En la actualidad se encuentran además varios estudios de casos exitosos en la mejora de procesos usando CMMI dentro de un contexto ágil tales como Pikkarainen y Mäntyniemi (2006), Boehm y Turner (2003) y Anderson (2005); Glazer et al. (2008). Además se puede decir que Bernal Cam y Calderón Valverde (2011), Fernández Díaz (2009), Navarro y Garzás (2010) y Sutherland y Ruseng (2008) estudian probables y palpables beneficios de esta integración. En el caso de Paulk (2001) se estudia específicamente la metodología de Programación Extrema (XP, eXtreme Programming) desde la perspectiva de CMMI. De la misma forma existen otros que dan argumentos en contra del uso de ambos de manera conjunta de acuerdo con la investigación de Matalonga (2012).

Ejemplos de investigaciones que explican cómo adaptar los procesos de PMBOK en proyectos que desarrollan ágil son Cottmeyer (2010), Sliger (2006a, 2006b). También existen estudios comparativos entre ambas tendencias como Fitsilis (2008) y Udo y Koppensteiner (2003); y otros como D. G. O'Sheedy, Xu, y Sankaran, (2010), Grey (2012) y D. O'Sheedy y Sankaran (2013) que muestra resultados concretos de la integración entre ágil y PMBOK.

En la actualidad por lo general las empresas parten de aplicar una metodología junto a un modelo específico. De forma independiente metodologías y modelos tienen ventajas y desventajas a raíz de las características propias de cada proyecto y terminan por aplicarse junto a un conjunto de prácticas empíricas y no documentadas de cada equipo de proyecto. Esto les permite a los equipos solventar los problemas y deficiencias que puedan tener al aplicar una metodología de las ya existentes junto a un

modelo de calidad. Esta práctica tan común trae consigo el hecho de que el conocimiento y la descripción del proceso real recaen en las personas y la transmisión de este conocimiento depende de estas.

De ahí la necesidad de establecer una metodología que recopile esas buenas prácticas tanto empíricas como definidas y las concentre en un modelo.

La metodología que se propone, Desarrollo Ágil con Calidad (DAC), trata de parecerse a la forma en que empíricamente trabajan los proyectos ágiles estableciendo al mismo tiempo pautas de calidad, procesos definidos y prácticas ágiles.

En las primeras versiones de CMMI y PMBOK no se tuvo en cuenta su aplicación junto a métodos ágiles de desarrollo de software. Sin embargo en las últimas versiones de ambos se introducen elementos, notas, procesos específicos para su aplicación en estos entornos.

En el PMBOK se definen 47 procesos de dirección de proyectos, 5 grupos de procesos de dirección de proyectos y 10 Áreas de Conocimiento de la Dirección de Proyectos (Meléndez De La Cruz, 2013).

CMMI-DEV contiene 22 áreas de proceso. De esas áreas de proceso, 16 son áreas de proceso base, 1 es un área de proceso compartida y 5 son áreas de proceso específicas de desarrollo.

Las áreas de procesos se organizan en cuatro categorías: Gestión de Procesos, Gestión de Proyectos, Ingeniería y Soporte (CMMIProductTeam, 2010).

En la Norma ISO/IEC 12207 (IEEE, ISO, & IEC, 2008) los procesos del ciclo de vida del software se agrupan en procesos de acuerdos, procesos de proyecto-activación organizacional, procesos de proyecto, procesos técnicos, procesos de implementación de software, procesos de soporte al software y procesos de reutilización del software.

La metodología DAC tiene ocho etapas del ciclo de vida llamadas fases o procesos: inicio del proyecto, análisis y diseño de alto nivel, desarrollo de requisitos, construcción del producto, cierre de iteración, liberación del producto, transición del producto y cierre del proyecto.

También cuenta con dos áreas de procesos de protección: gestión de proyectos y soporte; y las disciplinas de ingeniería: modelado del negocio, arquitectura y diseño, especificación de requisitos, implementación, pruebas, despliegue y mantenimiento.

En la Tabla 1 se muestra cómo integrar DAC con el PMBOK en cuanto a los grupos de procesos.

Tabla 1 Correlación entre DAC y PMBOK.

DAC	PMBOK
Inicio del proyecto	Grupo de procesos de iniciación
Análisis y diseño de alto nivel	Grupo de procesos de planificación
Desarrollo de requisitos	Grupo de procesos de ejecución Grupo de procesos de seguimiento y control
Construcción del producto	
Cierre de iteración	
Liberación del producto	
Transición del producto	
Cierre del proyecto	Grupo de procesos de cierre

Para incorporar a DAC las áreas de proceso de CMMI se definieron como complemento a los ocho procesos básicos de DAC un proceso para área de proceso CMMI de las siguientes:

- Gestión de Proyectos: REQM, PP, PMC, SAM, IPM, OPD, OPF, RSKM, QPM.
- Soporte: CM, MA, PPQA, DAR, OT, OPP, CAR, OPM.¹

Estos procesos de gestión de proyectos y soporte son de apoyo y solo son de obligatorio cumplimiento los procesos de gestión de la configuración, gestión de requisitos, aseguramiento de la calidad de procesos y productos, medición y análisis, monitoreo y control, planificación del proyecto y gestión de acuerdos con proveedores. El resto son opcionales a decisión de cada entidad desarrolladora.

Las áreas de proceso de la categoría ingeniería de CMMI se definen dentro de los ocho procesos básicos vinculadas a las disciplinas de ingeniería. Las áreas del conocimiento de PMBOK están incorporadas en la definición de los procesos de gestión de proyecto.

¹ Por cuestión de espacio se referencian las áreas de procesos por sus siglas según CMMIProductTeam, (2010).

De igual forma los procesos de la ISO/IEC 12207 se definieron dentro de los procesos de gestión de proyecto, soporte y las disciplinas de ingeniería como se muestra en la tabla 2.

Tabla 2 Integración de ISO/IEC 12207 y CMMI

CMMI ¹	ISO/IEC 12207
TS	Instalación del software Implementación del software Diseño arquitectónico del software Diseño detallado del software Construcción del software Ingeniería de dominio Gestión de reutilización de software Gestión de reutilización de activos
RD	Análisis de requisitos del software
PI	Instalación del software Integración del software
VER	Pruebas de calificación del software Verificación del software
VAL	Pruebas de calificación del software Soporte de aceptación del software Mantenimiento del software Validación del software
REQM	Definición de requisitos de los involucrados
PP	Planificación del proyecto
PMC	Evaluación y control del proyecto
SAM	Adquisición Proceso de oferta
IPM	Gestión de la Información Gestión del Modelo del Ciclo de Vida Gestión de la Infraestructura Gestión de los Recursos Humanos Gestión de la Calidad
CM	Gestión de la Configuración Gestión de la Configuración del Software
MA	Medición
RSKM	Gestión de Riesgos
PPQA	Aseguramiento de la Calidad del Software Revisión del Software Auditoría del Software
DAR	Resolución de Problemas del Software Gestión de Decisiones
CAR	Resolución de Problemas del Software Gestión de Decisiones

3.2. El modelo DAC

La metodología DAC es un proceso de desarrollo colaborativo, recursivo-iterativo, incremental y guiado por procesos y requisitos. Su modelo del proceso es una adaptación del modelo en cascada a los modelos programación extrema y desarrollo concurrente, incorporando elementos de los modelos incremental y evolutivo en cada iteración. Está enfocado a proyectos pequeños o proyectos grandes divididos en sub-proyectos que desarrollan software de gestión basado en componentes.

DAC plantea que el problema una vez identificado y definido debe ser descompuesto en problemas más pequeños, y si es necesario, realizar con estos la

misma operación. Cada sub-problema será resuelto mediante un componente y el problema resuelto será el software o producto final; por lo que las entregas en DAC son a nivel de iteración, en la que habrá obligatoriamente un incremento del producto a partir de la solución de un componente del mismo. Además en cada iteración se define como mínimo un hito a cumplir por cada fase.

Al finalizar cada iteración se realiza la integración del componente al producto obtenido hasta el momento realizando pruebas de integración. Las iteraciones no tienen que desarrollarse todas al mismo tiempo sino que depende del plan del proyecto.

En la Fig. 1 se puede ver el modelo del proceso DAC y en la Fig. 2 los componentes del modelo más detallados. Entre el Inicio y Cierre del proyecto se realizan iteraciones por entregas de versiones del producto. Entre las fases de Desarrollo de Requisitos, Construcción del Producto y Cierre de Iteración ocurren iteraciones concurrentes a nivel de componentes. Además, entre las fases de Desarrollo de Requisitos y Construcción del Producto puede ocurrir un ciclo pues a medida que los requisitos son descritos estos pueden ir entrando a la fase de Construcción del Producto.

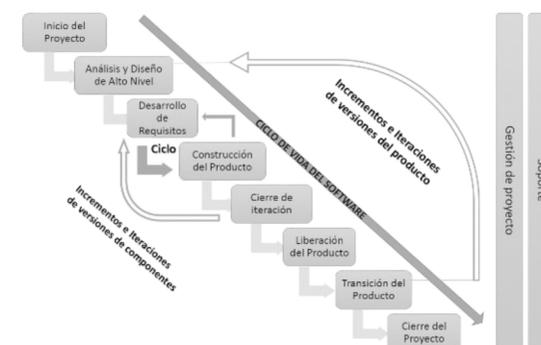


Fig. 1. Modelo del proceso DAC

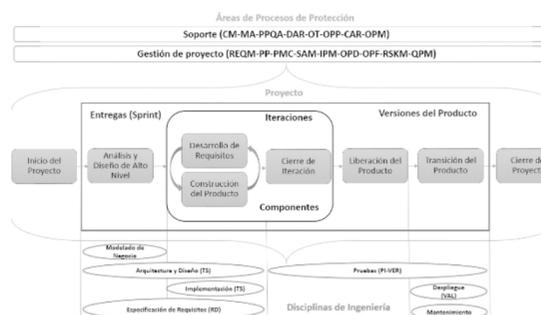


Fig. 2. Componentes del modelo DAC

4. DEFINIENDO LAS PRÁCTICAS

La aplicación de los valores y principios del manifiesto ágil (Beck et al., 2001) así como prácticas ágiles para la gestión de proyectos mejoran considerablemente los resultados de calidad del proyecto medidos en: indicadores de ejecución del proyecto, número de no conformidades encontradas en evaluaciones de calidad del proceso y productos de trabajo, satisfacción del cliente.

Basada en los principios y valores del manifiesto ágil (Beck et al., 2001) DAC defiende lo siguiente:

- Es prioridad para el proyecto satisfacer los acuerdos tomados con el cliente a través de la entrega temprana y continua de software de valor que cumpla con los atributos de calidad definidos.
- El cambio es bienvenido pero controlado, dejando siempre constancia de las solicitudes realizadas y las decisiones a tomar al respecto. Los cambios no pueden comprometer el éxito y la calidad del proyecto.
- Los involucrados relevantes, tanto externos como internos, deben estar en constante comunicación y realizar reuniones de chequeo del avance del proyecto de forma frecuente.
- Es vital para la gestión de la comunicación en el proyecto explotar las bondades de las nuevas tecnologías y la red. No obstante se declara como la forma más efectiva de comunicación la conversación cara a cara.
- Construir el proyecto con individuos capacitados en todo momento garantiza la mitad del éxito.
- La atención continua a la excelencia técnica enaltece la agilidad y determina la calidad.
- La reunión diaria es una técnica eficaz para mejorar la comunicación y la resolución de problemas; mientras que la reunión de chequeo y análisis de resultados, en intervalos regulares, se aprovecha para que el equipo reflexione sobre la forma de ser más efectivo y ajustar su conducta en consecuencia.
- La planeación colaborativa, iterativa y detallada no quita agilidad, brinda control.
- Evitar a toda costa tener documentación repetida.

Las metodologías ágiles de desarrollo de software como XP, Scrum, proceso guiado por funcionalidades (FDD, Feature Driven Development) y otras, prevalecen cada vez más en la industria del Software. DAC toma elementos de estas metodologías.

De XP (Beck y Andres, 2004; Jeffries, Anderson, y Hendrickson, 2001) se retoma la idea de realizar un plan de entregas, en este caso versiones del producto,

realizando durante el mantenimiento de la primera versión nuevas iteraciones para desarrollar las siguientes versiones del software. Se propone además retomar la idea de las siguientes prácticas XP adaptadas al concepto de DAC:

- El uso de la metáfora mediante diagramas de paquetes lógicos a partir del estudio del negocio o los requisitos de alto nivel.
- Diseño simple, abarcando únicamente el modelado del sistema mediante diagramas de paquetes o componentes solo hasta el nivel de funcionalidad. Se incluye como parte del diseño el mapa de navegación y el modelo de datos. No se realizan en DAC diagramas de clases o tarjetas clase-responsabilidad-colaboración (CRC). No son necesarios si en la implementación se aplican correctamente los patrones arquitectónicos y de diseño definidos.
- La producción del código está dirigida por pruebas de unidad desde el nivel de funcionalidad hasta el nivel de módulos para evitar no conformidades futuras en revisiones de calidad del producto.
- La refactorización del código antes de las pruebas finales contribuyen a la calidad del software disminuyendo la ocurrencia de no conformidades.
- Integración continua de cada componente del software una vez sea liberado por el grupo de aseguramiento y control de la calidad.
- Estándares de programación para fomentar la reutilización del código y la comunicación entre programadores.

DAC también asume todos los valores de Scrum (Palacio; Schwaber & Beedle, 2002). Esta metodología tiene buenas prácticas que se pueden aplicar en DAC:

- Revisión de las iteraciones con el equipo de proyecto. Las iteraciones en DAC pueden ser tan largas como se determine en la planificación con el cliente. En DAC hay dos tipos de iteraciones: iteraciones por versiones del producto e iteraciones por versiones de componentes.
- Desarrollo incremental obteniendo en cada iteración al menos un componente del software pudiendo ser este una entrega pactada o parte de ella.
- Desarrollo evolutivo, aunque en las fases iniciales de cada versión del producto se define una arquitectura y diseño de alto nivel, esta se va actualizando y mejorando durante las iteraciones.
- Colaboración entre los miembros del equipo de desarrollo. DAC pone especial énfasis a compartir y socializar el conocimiento partiendo

del hecho de que un equipo de proyecto es una comunidad de desarrollo. También se propone el uso de repositorios de activos de productos y componentes.

- DAC define varios tipos de reuniones: reunión de inicio de una iteración, reunión diaria, reunión de análisis de resultados (máximo cada 30 días), reunión de chequeo de acuerdos con involucrados, reunión de cierre de iteración.

Por otra parte, de la metodología FDD (Palmer & Felsing, 2001) se retoma en DAC la idea del desarrollo de un modelo inicial de alto nivel, que se corresponde con la metáfora de XP. DAC indica que debe establecerse una arquitectura base para poder iniciar el desarrollo a partir de la priorización y agrupamiento de los requisitos del cliente o de alto nivel. Estos requisitos se traducen en objetivos del proyecto, que guiarán el proceso de desarrollo mediante su desglose, refinamiento, análisis, especificación y validación en las distintas fases.

También se incluyen actividades como el estudio de los documentos, el desarrollo de un modelo en áreas y un modelo global. De estos modelos salen las propuestas de componentes en distintos niveles de empaquetamiento y de cada componente las iteraciones necesarias para desarrollarlos.

En Letelier (2013) se presenta una lista creada por este autor de 42 prácticas ágiles “a modo de carta de restaurante para que cada equipo configure su propio menú”. Además, explica que “Este enfoque conlleva la promoción de una estrategia de implantación que apuesta por combinar prácticas cogidas desde diferentes métodos ágiles”. La mayoría de estas prácticas se aplican en DAC de forma íntegra; sin embargo otras deben aplicarse haciendo algunas adaptaciones. Entre estas:

- Formar equipos pequeños (si el trabajo está organizado se puede dividir un equipo grande en pequeños grupos de desarrollo).
- El equipo se autoorganiza y toma las decisiones técnicas (a veces un equipo que se autoorganiza tiende a acomodarse y no planificar de la manera más eficiente).
- Colocalización de miembros del equipo, todo el equipo trabajando en el mismo espacio físico (con el uso de las tecnologías de la información y comunicaciones esto no es necesario, al menos no siempre).
- Que los integrantes del equipo no tengan solo algunas actividades fijas asignadas (es ideal un equipo así pero en DAC se le da mucha importancia a la especialización de cada miembro en su rol).

- Documentar, pero solo lo estrictamente necesario, que sea rentable el aprovechamiento de la documentación respecto del esfuerzo asociado a elaborarla (en DAC es utilizar cada uno de los productos de trabajo definidos en la metodología de la manera más eficiente y en el momento que corresponda).
- Establecer pautas para gestionar convenientemente el retrabajo (mientras sea posible evitar el retrabajo y si ocurre debe analizarse, priorizarse e incorporarse a la planificación).

Uniendo todo lo anteriormente expuesto junto a un conjunto de buenas prácticas generales en el desarrollo de software a continuación se resumen las prácticas de DAC.

Calidad en la gestión de los recursos humanos:

- Realizar las reuniones necesarias planificadas.
- Comunicación mediante las redes e internet y cara a cara siempre que sea posible.
- Capacitación constante del equipo.
- Socialización del conocimiento.
- Colaboración constante.
- 40 horas semanales.

Calidad de los productos de trabajo:

- Definir la arquitectura de alto nivel al inicio de cada versión.
- Planificación conjunta entre cliente y equipo de proyecto.
- Arquitectura y diseño basados en componentes y patrones.
- Refactorización de código.
- Integración continua.
- Estándar de código.
- Pruebas en cada iteración y versión (pruebas funcionales, de integración, no funcionales, aceptación alfa y beta).
- Desarrollo evolutivo-incremental.
- Evaluar la calidad de cada entregable.
- Gestionar líneas base.
- Revisar y gestionar inconsistencias entre los productos de trabajo.

Calidad en los procesos:

- Documentar, firmar y satisfacer los acuerdos con involucrados y compromisos con el plan.
- Controlar y firmar las solicitudes de cambio.
- Planificar y ejecutar el aseguramiento de la calidad del proceso y del producto.
- Planificar guiado por entregas y requisitos.
- Adaptar los procesos definidos.
- Seguir los procesos definidos.

- Analizar los resultados.
- Documentar los procesos y sus guías.
- Debe existir un administrador de la calidad y la configuración.
- Firmar actas de aceptación de los entregables.
- Monitorear los procesos.
- Elaborar productos de trabajo simples que cumplan con su propósito de uso.

5. VALIDACIÓN DE LA PROPUESTA

La metodología fue aplicada durante un año en el Departamento de Desarrollo de la Dirección de Informatización de la Universidad de las Ciencias Informáticas en 6 proyectos de desarrollo. Se aplicó una configuración de la metodología incorporando las áreas de procesos del nivel dos de CMMI.

A continuación se mostrarán los resultados del experimento teniendo en cuenta las siguientes variables:

Recopiladas en informes de resultados de calidad:

- Índice de no conformidades de adherencia a procesos y productos (NC 1).
- Índice de no conformidades en pruebas de software (NC 2).

Recopiladas en informes de estado de ejecución del proyecto:

- Índice de ejecución del proyecto (IE).
- Índice de rendimiento de la ejecución (IRE).
- Índice de rendimiento de la planificación (IRP).
- Índice de rendimiento de los recursos humanos (IRRH).
- Índice de rendimiento de la eficacia (IREF).

En la Fig. 3 se muestran los resultados promedios de calidad y ejecución antes y después de la aplicación de la metodología en la muestra seleccionada.

En el caso de las no conformidades, a mayor porcentaje peor el resultado. En el caso de los indicadores de ejecución, a mayor porcentaje mejor resultado.

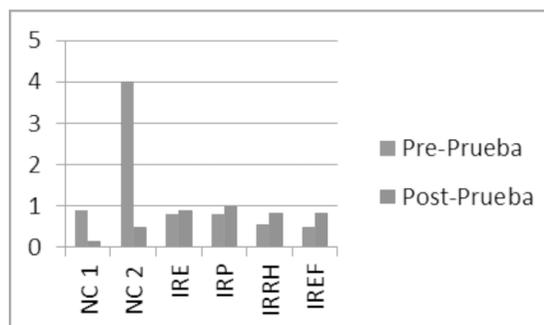


Fig. 3. Resultados del experimento

Evidentemente hubo un cambio en los resultados de los proyectos, comenzando por el propio hecho de recopilar y analizar los resultados con frecuencia como parte de los procesos de apoyo de la metodología.

El intercambio con los miembros del proyecto fue fundamental. Se realizaron entrevistas en grupo a los miembros de cada proyecto. En la post-prueba el porcentaje de no conformidades de adherencia a procesos y productos disminuyó en un 83,34 % con respecto a la revisión anterior. Además los indicadores relevantes mejoraron obteniendo buenas evaluaciones superiores al 85 % en todos los indicadores.

La adherencia a los procesos definidos en DAC propició planificar por entregas los proyectos y proveer a los clientes versiones del software en frecuencias cortas. Cada entrega de software estuvo precedida de revisiones de calidad, liberaciones de líneas base, auditorías a la configuración y pruebas de software en los distintos niveles.

Los proyectos seleccionados habían sido abortados con anterioridad en sus intenciones de liberación de productos por la empresa CALISOFT. Posterior a la implementación de la metodología tres de los proyectos lograron una liberación de la primera versión de los productos en desarrollo.

5.1. Trabajos futuros

Como trabajo futuro se prevé validar la propuesta por el criterio de expertos así como medir y analizar todos los indicadores previstos como parte del diseño de la investigación (Tabla 3).

Tabla 3 Dimensiones e indicadores para analizar

Dimensión	Indicador
Aplicabilidad en el proyecto	Estado del proyecto
	Estado del proceso en el proyecto
	Estado del equipo de proyecto
	Estado del producto desarrollado en el proyecto
Calidad de la metodología	Cubrimiento del enfoque ágil
	Cubrimiento del modelo CMMI
	Cubrimiento de estándares de gestión de proyectos
	Cubrimiento de estándares de calidad
	Capacidad de aplicación en el desarrollo basado en componentes y líneas de producto
	Capacidad de aplicación en el desarrollo de software de gestión
	Cubrimiento de las características definidas (Características propias de la metodología declaradas)
	Adaptabilidad
	Capacidad de generalización
	Existencia de elementos conceptuales
	Existencia de una guía de actividades y tareas
	Definición de roles que intervienen
	Definición de productos de trabajos generados
Nivel de aceptación	
Calidad del proceso en revisiones y auditorías	Cantidad de no conformidades por impacto
	Grado de cobertura de procesos
	Grado de cobertura de productos
	Adherencia a procesos
Calidad del producto de acuerdo a pruebas de software formales	Adherencia a productos
	Cantidad de no conformidades por criticidad
Calidad del producto en explotación	Cantidad de no conformidades por tipo
	Cantidad de errores en producción
Aceptación del producto por el cliente	Cantidad de usuarios afectados por errores en producción
	Nivel de aceptación del producto
Aceptación del producto por el usuario	Nivel de cumplimiento de acuerdos y cronogramas
	Nivel de aceptación del producto
	Nivel de uso del producto

6. CONCLUSIONES

La metodología Desarrollo Ágil (DAC) con Calidad provee a la entidad desarrolladora un conjunto de procesos documentados, con sus guías de aplicación, propuestas de productos de trabajo, paquete de productos de trabajo reutilizables y guías de adaptación, como fuente principal de conocimiento.

La metodología DAC define actividades en su ciclo de vida que contribuyen a un enfoque de calidad obteniendo las evidencias necesarias a través de productos de trabajo sencillos como son las revisiones de calidad para la liberación de las líneas bases en cada fase del proyecto, definición temprana y ejecución constante del plan de pruebas teniendo en cuenta distintos niveles, auditorías a la configuración y establecimiento de estándares.

La metodología DAC constituye una alternativa al desarrollo de software siguiendo un enfoque tradicional para la aplicación de modelos y estándares internacionales como PMBOK, ISO y CMMI. Los procesos de apoyo de DAC pueden ser configurados y adaptados por la entidad desarrolladora para adaptarse a un modelo estándar deseado.

Un proyecto que implemente DAC puede planificar las entregas a nivel de versiones del producto desarrollando de manera ágil y concurrente. El uso de los principios y prácticas de esta metodología guían al proyecto en la obtención de un producto con calidad y la satisfacción del cliente en etapas tempranas.

Al aplicar la metodología en seis proyectos de desarrollo reales se puede decir que propicia mayor organización y control en los proyectos. Favorece la documentación y tratamiento oportuno de los cambios solicitados y las acciones correctivas y preventivas ante la ocurrencia de riesgos. El análisis de resultados y la retrospectiva permite al equipo de proyecto mejorar de manera continua.

Se debe aún validar la metodología mediante el criterio de expertos y aplicándola en un entorno empresarial para probar su efectividad en este sector.

REFERENCIAS

Anderson, D. J. (2005). *Stretching agile to fit CMMI level 3 - the story of creating MSF for CMMI reg; process improvement at Microsoft corporation*. Paper presented at the Agile Conference, 2005. Proceedings.

- Beck, K., y Andres, C. (2004). *Extreme programming explained: embrace change* Addison Wesley.
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Sutherland, J. (2001, marzo). Manifiesto Ágil, recuperado de: www.agilemanifesto.org.
- Bernal Cam, C., & Calderón Valverde, J. L. (2011). *Aplicación de BPM, CMMI y métodos ágiles en el desarrollo de software para empresas de producción y exportación*. (Tesis de Grado) Universidad Peruana de Ciencias Aplicadas, Lima.
- Boehm, B., y Turner, R. (2003). *Balancing agility and discipline: A guide for the perplexed*: Addison-Wesley/Pearson Education.
- CMMI Institute. (2014). Maturity Profile Report (pp. 13). Pennsylvania: Carnegie Mellon University.
- CMMIProductTeam. (2010). CMMI® for Development, (Version 1.3.). Pennsylvania: Carnegie Mellon University.
- Cottmeyer, M. (2010). The Agile Project Manager Retrieved 2014, recuperado de: http://www.versionone.com/pdf/V1_The_Agile_Project_Manager.pdf
- Fernández Díaz, Y. (2009). *Estudio sobre la correspondencia entre prácticas CMMI y prácticas ágiles y su aplicación en Pyme*. (Tesis de maestría), Universidad Politécnica de Madrid, Madrid.
- Fitsilis, P. (2008). Comparing PMBOK and Agile Project Management software development processes. In T. Sobh (Ed.), *Advances in Computer and Information Sciences and Engineering* (pp. 378-383): Springer Netherlands.
- Glazer, H., Dalton, J., Anderson, D., Konrad, M., Shrum, & S. (2008). CMMI or Agile: Why Not Embrace Both? recuperado de: <http://www.sei.cmu.edu/library/abstracts/reports/08tn003.cfm>.
- Grey, J. (2012). *The Development of a Hybrid Agile Project Management Methodology*. North-West University, Potchefstroom Campus.
- HP, Ayer Sogeti, & Capgemini. (2013). World Quality Report.
- IEEE, ISO, & IEC. (2008). Standard for Information-Software life cycle processes (Vol. ISO/IEC 12207: 2008). New York: The Institute of Electrical and Electronics Engineers, Inc.
- Jeffries, R., Anderson, A., & Hendrickson, C. (2001). *Extreme programming installed*: Addison Wesley.
- Letelier, P. (2013). Carta de Prácticas Ágiles: Arma tu propio menú ágil. *Agilismo at work*, (pp. 13) Recuperado de : <http://agilismoatwork.blogspot.com.es>
- Matalonga, S. (2012). *Agile vs CMMI: Reasons why the "vs." is sticking for the time being*. Agile Product and Project Management, 13(15).
- Meléndez De La Cruz, F. (2013). Nueva versión del PMBOK – Edición 2013, Lima.
- Navarro, J. M., & Garzás, J. (2010). Experiencia en la implantación de CMMI-DEV v1.2 en una micropyme con metodologías ágiles y software libre. *REICIS Revista Española de Innovación, Calidad e Ingeniería del Software*, 16(1), 6-15.
- O'Sheedy, D., & Sankaran, S. (2013). Agile Project Management for IT Projects in SMEs: A Framework and Success Factors. *The International Technology Management Review*, 3(3), 187-195.
- O'Sheedy, D. G., Xu, J., & Sankaran, S. (2010). Preliminary Results of a Study of Agile Project Management Techniques for an SME Environment. *International Journal of Arts and Sciences*, 3(7), 278-291.
- Palacio, J. El modelo Scrum. 2006. Disponible en: http://navegapolis.net/files/s/NST-010_01.pdf
- Palmer, S. R., & Felsing, M. (2001). *A practical guide to feature-driven development*: Pearson Education.
- Paulk, M. C. (2001). Extreme programming from a CMM perspective. *Software, IEEE*, 18(6), 19-26.

- Pikkarainen, M., & Mäntyniemi, A. (2006). *An Approach for Using CMMI in Agile Software Development: Experiences from three case studie*. Paper presented at the Proceeeding of the 2006 SPICE Conference.
- PMI. (2013). *A Guide to the Project Management Body of Knowledge*. Pennsylvania: Project Management Institute.
- Schwaber, K., & Beedle, M. (2002). *Scrum: The Art of Doing More with Less*. Redmond, WA: Microsoft Press.
- Sliger, M. (2006a). A project manager's survival guide to going agile. *Rally Software Development Corporation*.
- Sliger, M. (2006b). Relating PMBoK Practices to Agile Practices. Retrieved from StickyMinds.com web site.
- Sutherland, J., & Ruseng, C., Johnson, Kent. (2008). *Scrum and CMMI Level 5: The Magic Potion for Code Warriors*. Paper presented at the Proceedings of the 41st Hawaii International Conference on System Sciences, Hawaii.
- The Standish Group International, I. (2013). *The CHAOS Manifesto. Think Big, Act Small* (pp. 48).
- Udo, N., & Koppensteiner, S. (2003). Will Agile Development Change The Way We Manage Software Projects? Agile From a PMBOK® Guide Perspective. *Proc. PMI Global Congr., Project Manage. Inst*, 22-23.

