Blue WATCH: Un marco metodológico balanceado para el desarrollo de software en pequeñas empresas

Jonás Montilva C., William Montilva C., and Judith Barrios A.
Universidad de Los Andes, Facultad de Ingeniería, Escuela de Sistemas,
Departamento de Computación, Grupo de Investigación GIDYC, Mérida, Venezuela
{jonas,william,ijudith}@ula.ve

RESUMEN

La industria mundial del software está conformada, en su mayoría, por empresas que tienen menos de 25 empleados. El desarrollo de software, en estas empresas, se lleva a cabo bajo diferentes enfoques que van desde lo artesanal (no gestionado e improvisado) hasta lo altamente disciplinado (planificado, controlado y documentado). Agilidad y disciplina representan dos extremos donde se ubican los métodos de desarrollo de software usados en este importante sector de la economía mundial. Un balance entre estas dos corrientes es considerado, hoy en día, como el enfoque más apropiado para este tipo de empresas. Este artículo describe un marco de trabajo metodológico balanceado para el desarrollo de software, denominado Blue WATCH, el cual está dirigido al desarrollo de aplicaciones empresariales de pequeña o mediana complejidad y tamaño. Este marco de trabajo es apropiado para pequeños grupos de desarrollo de software de 2 – 10 personas y fue concebido, expresamente, para ser utilizado en pequeñas y medianas empresas de desarrollo de software. El marco de trabajo puede ser fácilmente adaptado a las características particulares de cada empresa, proyecto y grupo de trabajo.

Keywords: Ingeniería del Software, métodos de desarrollo de software, agilidad y disciplina en el desarrollo de software.

ABSTRACT

The majority of the software enterprises, in the world, are composed by less than 25 employees. The existing software development approaches are located in a spectrum that goes from an unplanned and undisciplined hacking process to a highly disciplined and managed process. Agility and discipline are two extremes of this spectrum. Today, many authors recognize the importance of acheiving a balance between agility and discipline. In this article, we propose a balanced methodological framework for sofware development, called Blue WATCH. This framework is oriented to the development of small and medium size enterprise applications by small teams composed by 2 – 10 people. It was conceived to be applied in very small enterprises. This framework can be easily adapted to any small enterprise, project and development team.

Keywords: Software Engineering, Software development methods, Agility and discipline in software development.

1. INTRODUCCIÓN

Varios estudios realizados, recientemente, en diferentes países demuestran que la industria mundial de Tecnologías de Información (TI) está compuesta en su mayoría por Pequeñas y Medianas Empresas (PYMES). Un estudio realizado, a nivel mundial, por Laporte, Alexandre y Renault (2008) encontró, por ejemplo, que: (1) en Europa el 85% de las empresas del sector TI tienen entre 1 y 10 empleados; (2) en Canadá las empresas de ese sector tienen menos de 25 empleados; y (3) en Brasil las pequeñas empresas representan el 70% del total de aquellas enmarcadas en esta actividad económica.

Dentro del sector TI, se ubican las empresas dedicadas al desarrollo de software (EDS). El estudio realizado por el equipo conducido por Laporte encontró, también, que el 75% de las empresas EDS analizadas tienen menos de 25 empleados. Otro estudio realizado por investigadores de la Universidad de Los Andes, en Venezuela, encontró que, en ese país, el 75 % de empresas dedicadas al desarrollo de software tienen menos de 20 empleados (Rivero, Montilva, Barrios & Murúa, 2009).

Un amplio número de métodos, modelos de mejora, estándares y prácticas, tanto aquellos publicados en la literatura como los distribuidos comercialmente, están dirigidos a medianas y grandes empresas (con más 50 empleados). Por ejemplo, el proceso RUP™ (*Rational Unified Process*) de IBM es uno de los métodos de desarrollo de software más conocido internacionalmente. Sin embargo, su uso en pequeñas empresas de desarrollo de software (PEDS), es decir aquellas que tienen entre 1 y 25 empleados, es muy limitado debido al elevado número de actividades, artefactos y roles que el método impone. Similarmente, modelos de mejoras de procesos, tales como el CMMI (Software Engineering Institute [SEI], 2010), ISO 9000 e ISO 15504, son difíciles de aplicar en las PEDS, tal como se demuestra en el trabajo realizado por Pino, García y Piattini (2008).

Una de las razones que dificultan el uso, en PEDS, de los métodos, prácticas y modelos mencionados anteriormente, es su carácter disciplinado. Métodos disciplinados, como RUP (Krutchen, 2000), Gray WATCH (Montilva, Barrios & Rivero, 2008) y TSP (Humphrey, 2006), requieren grupos de trabajo organizados que tengan mucha experiencia, tiempo y dedicación. Estos métodos prescriben el proceso de desarrollo con mucho rigor y requieren que su aplicación esté guiada por planes previamente elaborados y detallados.

A partir del año 2001, a raíz del lanzamiento del Manifiesto Ágil (Beck, 2001), ha surgido una amplia colección de métodos, modelos y prácticas que, en su mayoría, pueden ser aplicados, por grupos de trabajo relativamente pequeños, para desarrollar aplicaciones de mediana o pequeña complejidad y tamaño.

A pesar de que los métodos, modelos y prácticas ágiles están más orientados a pequeños grupos de trabajo, existen críticas sobre su efectividad en la producción de software de alta calidad. Algunas de ellas se centran en la escasa documentación que estos métodos y modelos generan, lo cual afecta el mantenimiento de las aplicaciones. Para solventar las dificultades presentes, en estos dos enfoques contrapuestos, Barry Boehm aboga por un enfoque balanceado que equilibre los mejores aspectos de ambos enfoques (Boehm, 2002).

En este artículo, se describe un marco de trabajo metodológico para el desarrollo de software, denominado Blue WATCH. Este marco de trabajo es balanceado y está dirigido al desarrollo de aplicaciones empresariales de pequeña o mediana complejidad y tamaño. Es más apropiado para pequeños grupos de desarrollo de software de 2 – 10 personas y fue concebido expresamente para ser utilizado en pequeñas y medianas empresas de desarrollo de software. El marco de trabajo es iterativo, extensible, adaptable, reutilizable, completo y balanceado.

El resto de este artículo está organizado como sigue: La sección 2 establece las diferencias entre los enfoques de desarrollo de software ágil y disciplinado y plantea la necesidad de un enfoque balanceado. En la sección 3, se especifican las características y los requisitos que debe tener un método de desarrollo de software, para que sea considerado como balanceado. Los aspectos generales del marco metodológico Blue WATCH están descritos en la sección 4; mientras que la estructura y componentes de dicho marco de trabajo se detallan en la sección 5. Finalmente, la sección 6 presenta las conclusiones de este trabajo.

2. LOS ENFOQUES ÁGILES Y DISCIPLINADOS

En la actualidad, el desarrollo de software se puede llevar a cabo de maneras muy diversas, que van desde un enfoque completamente ágil hasta un enfoque totalmente disciplinado. La tabla 1 resume las principales características de estos dos enfoques y facilita su comparación.

Como puede apreciarse en la tabla 1, los dos enfoques de desarrollo de software son contrapuestos. Cada uno tiene ventajas que se manifiestan dependiendo del contexto en el cual el enfoque se aplique. Así, por ejemplo, el enfoque disciplinado es más apropiado para medianas y grandes empresas, que posean elevados niveles de madurez y capacidad en la producción de software; mientras, que el enfoque ágil ha demostrado ser más útil en empresas pequeñas o medianas donde elevados niveles de madurez y capacidad no sean requeridos. La madurez y capacidad de una empresa, para producir software de alta calidad, está en relación directa con la mejora de los

procesos que ella emplee para desarrollar y mantener sus aplicaciones. Una empresa es madura y capaz si sus procesos de software están definidos, son gestionados y se mejoraran continua o gradualmente (SEI, 2010).

Pese a que un porcentaje pequeño de las PEDS (menos del 18%) se encuentran certificadas, bajo uno de los modelos de mejora CMMI, ISO 9000 o SPICE, existe, en estas empresas, alto interés y necesidad de mejorar sus niveles de capacidad y madurez (Laporte et al., 2008). Para ello, un balance entre agilidad y disciplina es indispensable.

Boehm y Turner (2003) plantean la necesidad de balancear ambos enfoques debido a que: (1) ninguno de los dos enfoques constituye una "bala de plata" para resolver los problemas del desarrollo de software; (2) cada uno de los dos enfoques tienen dominios o territorios donde su aplicación es más efectiva; y (3) los métodos son importantes; pero, también, lo son la gente, los valores, la comunicación y las expectativas gerenciales.

Tabla 1: Comparación entre los Métodos Ágiles y Métodos Disciplinados

Métodos Ágiles	**Métodos Disciplinados**
Basados en el **Manifiesto Ágil,** dan mayor valor a: • Individuos y su interacción • Código que funcione • Colaboración con el cliente • Respuesta al cambio	Basados en la **Calidad Total**, dan mayor valor a: • El proceso de desarrollo • Los productos incluyendo el código • Contrato con el cliente • Seguimiento de un plan
Centrados en las personas	Centrados en los procesos
Énfasis en el código y la cooperación	Énfasis en los productos y la organización
Procesos mínimos con poca documentación	Procesos bien definidos y documentados
Formalidad del proceso reducida a un mínimo	Alta formalidad en el proceso de desarrollo
Procesos no repetibles	Procesos repetibles
Resultados dependen del grupo de trabajo	Resultados predecibles
Proyectos pequeños o medianos	Grandes proyectos de desarrollo
Grupos de desarrollo pequeños	Grupos de desarrollo numerosos
Principales representantes: • XP, Scrum, Crystal Light Methods, Feature Driven Development, Adaptive Software Development, Lean Development	Principales representantes: • RUP, OOSE, OOAD, Personal Software Process (PSP), Team Software Process (TSP), Gray WATCH

3. CARACTERÍSTICAS Y REQUISITOS DEL DESARROLLO DE SOFTWARE EN PYMES

El tamaño de una empresa es un factor que influye significativamente en la manera en que el desarrollo de software se lleva a cabo. Las PEDS tienen características muy particulares que las diferencian de las grandes empresas del sector. Por ejemplo, en las PEDS: los grupos de trabajo son pequeños (1-5 personas); los proyectos son, generalmente, de corta duración (1-6 meses); la gestión del proyecto es bastante informal; los recursos humanos, financieros y tecnológicos asignados al proyecto son limitados; y la madurez y capacidad de la empresa para desarrollar software de alta calidad es muy baja (Rivero, et al., 2007, 2009).

A estas características se agregan otras relacionadas con las limitaciones económicas que las PEDS tienen. Este tipo de empresas dispone de muy pocos recursos económicos para invertir en: la capacitación de los grupos de trabajo, la mejora de los procesos de software, la certificación de la calidad y el uso disciplinado de procesos de soporte, tales como el aseguramiento de la calidad y la gestión de riesgos (Laporte et al., 2008).

En base a estas características, se ha establecido un conjunto mínimo de requisitos que deben cumplir los métodos y modelos de procesos de desarrollo de software empleados en las PEDS (Cabral, 2010):

- Deben estar orientados a las características particulares de las PEDS.
- Deben ser balanceados, es decir, deben equilibrar prácticas ágiles con disciplinadas.

- Deben estar dirigidos a pequeños grupos de trabajo (2-10 personas).
- Deben ser flexibles y adaptables a las características de cada proyecto, aplicación y grupo de trabajo.
- Deben ser completos, es decir, deben describir tanto los procesos técnicos del desarrollo de software, como los procesos de gestión del proyecto. Todos estos procesos deben ser balanceados.
- La gestión de proyectos debe apoyarse en procesos de soporte balanceados, tales como: aseguramiento de la calidad, control de configuración, verificación/validación continua y gestión de riesgos.

4. EL MARCO DE TRABAJO METODOLÓGICO BLUE WATCH

Blue WATCH es un marco de trabajo metodológico para el desarrollo de aplicaciones empresariales de mediana complejidad y/o tamaño. Algunas típicas aplicaciones, en las que este marco metodológico se puede aplicar son las siguientes: comercio electrónico, gobierno electrónico, sistemas de información web, portales corporativos, aplicaciones para dispositivos móviles y aplicaciones colaborativas.

Un marco de trabajo metodológico para el desarrollo de software es un conjunto de modelos que describen los productos, procesos y actores que deben participar en un proyecto de desarrollo de software. Un marco metodológico se diferencia de un método de desarrollo de software en que el primero sirve de guía o patrón para la elaboración del segundo. Una premisa importante usada para diseñar Blue WATCH como marco metodológico es la conclusión dada por Boehm y Turner (2003) que establece que "es mejor construir un método gradualmente, que adaptarlo a partir de uno muy elaborado". Blue WATCH puede ser usado como un patrón al cual, las empresas que deseen utilizarlo, le van agregando elementos metodológicos, o lo van modificando progresivamente, para crear un método propio adaptado a las características particulares de la empresa, de sus proyectos y de sus grupos de trabajo.

Blue WATCH está compuesto por tres modelos. El primero de ellos, denominado Modelo de Productos, describe el conjunto de productos intermedios y finales que se deben elaborar durante el desarrollo de cualquier aplicación empresarial de pequeña o mediana complejidad. El segundo modelo, llamado Modelo de Procesos, describe las actividades técnicas, de gestión y de soporte que se requieren ejecutar para elaborar una aplicación. Finalmente, el Modelo de Actores identifica los roles requeridos para desarrollar una aplicación y recomienda una estructura organizacional para el grupo de trabajo. Los tres modelos son patrones metodológicos adaptables y extensibles.

El Modelo de Procesos es representado en la figura 1 a través de una cadena de valor. Los procesos ubicados en la parte superior de la cadena de valor son los procesos técnicos que Blue WATCH recomienda seguir para desarrollar una aplicación empresarial. En la parte inferior de la cadena, se ubican los procesos de gestión del proyecto y los procesos de apoyo; ambos se llevan a cabo a lo largo de la ejecución de todos los procesos técnicos. La sección 5.2 describe cada uno de estos procesos y, en especial, el Desarrollo de Versiones.

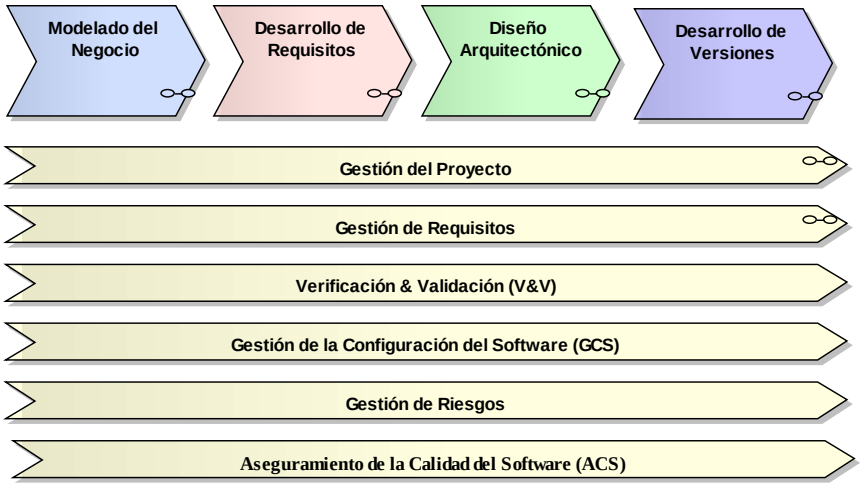

Figura 1. Procesos de desarrollo de software según el marco metodológico Blue WATCH

Blue WATCH está basado en el enfoque de Desarrollo de Software Guiado por Modelos - MDA (Mellor et al., 2003); enfoque mediante el cual, una aplicación se desarrolla a través de la transformación de modelos elaborados usando el lenguaje UML. Dicha transformación se apoya en herramientas CASE que facilitan la elaboración de los modelos y su conversión automática. Los modelos producidos, siguiendo el marco metodológico Blue WATCH, son transformados a lo largo de una serie de procesos técnicos del desarrollo, que van desde el Modelado del Negocio hasta las Pruebas de la Aplicación, pasando por los procesos de Desarrollo de Requisitos, Diseño Arquitectónico, Diseño Detallado y Programación & Integración.

4.1 LOS CICLOS DE DESARROLLO DEL MARCO METODOLÓGICO BLUE WATCH

El Modelo de Procesos del Blue WATCH identifica y ordena los procesos técnicos, de gestión y de soporte que se requieren para desarrollar una aplicación empresarial. Este modelo le indica, al grupo de desarrollo, cómo deben llevarse a cabo estos procesos.

Este modelo es iterativo, versionado e incremental. Es iterativo porque sus procesos se ejecutan cíclicamente, es decir, se repiten un número determinado de veces hasta alcanzar el resultado deseado. Es versionado porque la aplicación se realiza, gradualmente, en varias versiones, con un ciclo para cada versión. Es incremental porque los programas, de cada versión, se desarrollan en incrementos que se van entregando al cliente gradualmente.

El modelo de procesos del Blue WATCH está inspirado en la metáfora del reloj de pulsera, que consta de un motor que hace avanzar las horas, una vez completado un ciclo de minutos; y hace avanzar los minutos, una vez completado un ciclo de segundos (ver figura 2.a). Cada aplicación se desarrolla mediante un conjunto de ciclos de procesos. La figura 2.b muestra estos ciclos y los productos que cada uno genera. El ciclo de una aplicación contiene uno o más ciclos de versiones y cada uno de estos últimos contienen, a su vez, varios ciclos de iteraciones.

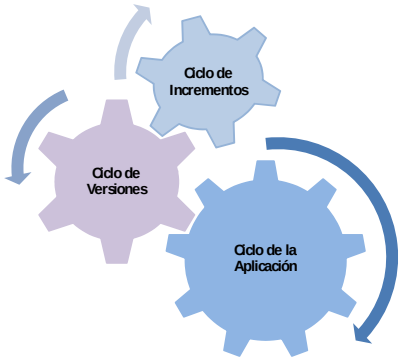

Figura 2.a: Relaciones entre ciclos de desarrollo

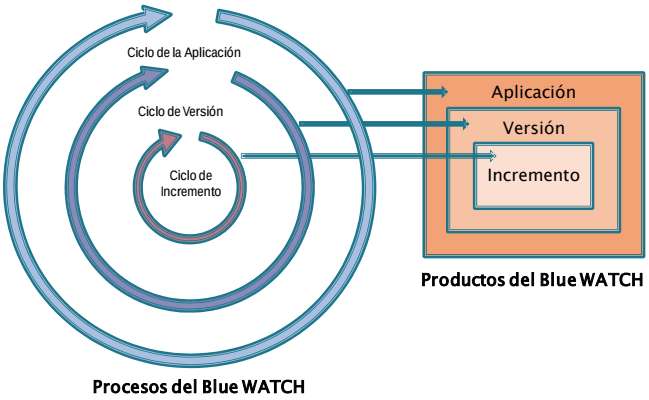

Figura 2.b: Los ciclos de desarrollo y sus productos

Cada ciclo del incremento produce una pieza de código ejecutable denominada incremento. Esta pieza de código se produce en un tiempo muy corto (1 - 3 semanas). Cada incremento implementa un subconjunto de funciones de la aplicación, las cuales, generalmente, se representan mediante casos de uso en la notación UML. Un incremento produce resultados concretos para el usuario, que aunque son parciales, le permiten al usuario validar partes de la aplicación.

Cada ciclo de versión produce una versión completa y operativa de la aplicación. Una versión es un producto acabado que implementa un conjunto completo de funciones y puede ser utilizada inmediatamente, por sus usuarios, una vez que ella ha sido instalada en su plataforma de operación. La siguiente versión de la aplicación agrega a la anterior nuevas funciones o mejora las funciones ya implementadas.

Tal como se plantea en las figuras 2a-b, una aplicación es un producto de software que evoluciona, a lo largo del tiempo, a través de una serie de versiones; cada una de las cuales es producida por un ciclo de versión. La versión

es, también, un producto evolutivo que se genera mediante la agregación progresiva de incrementos. Cada incremento es generado por un ciclo de desarrollo menor.

4.2 OTRAS CARACTERÍSTICAS DEL MARCO METODOLÓGICO BLUE WATCH

- **Es completo:** Cubre todo el ciclo de desarrollo de una aplicación e integra los procesos técnicos del desarrollo de software con los procesos de soporte y gestión del proyecto.
- **Es extensible y adaptable:** Sus componentes se pueden extender para incorporar otros procesos, productos, prácticas o actores propios de la cultura de desarrollo de una empresa o de las características particulares de un proyecto y/o grupo de trabajo.
- **Es reutilizable:** Una vez extendido o adaptado, el marco de trabajo puede reutilizarse como un método estándar para el desarrollo de todos los proyectos de software de una empresa.
- **Produce un conjunto mínimo de documentos**: Con la finalidad de reducir el esfuerzo requerido durante el desarrollo de una aplicación, el método produce un número reducido de documentos que no afectan el proceso de mantenimiento de la aplicación.
- **Es balanceado:** Proporciona un balance entre agilidad y disciplina. Combina prácticas ágiles con procesos y prácticas disciplinadas, particularmente, algunas de las establecidas en el modelo CMMI y los cuerpos de conocimiento SWEBOK y PMBOK.

4.3 PRÁCTICAS USADAS EN EL MARCO METODOLÓGICO BLUE WATCH

La tabla 2 identifica el conjunto de prácticas ágiles (A) y disciplinadas (D) que Blue WATCH recomienda utilizar, en sus diferentes procesos, y que pueden ser aplicadas en cualquier método balanceado.

Tabla 2: Prácticas ágiles y disciplinadas que pueden utilizarse en métodos balanceados

Aplicabilidad	Práctica (A: Ágil - D: Disciplinada)
En todos los procesos del desarrollo de software	Trabajo estimulante (A y D) Control de versiones (D) Confianza (A)
En la Gestón del Proyecto	Visión del Producto (D) Planificación y control del proyecto (D) Reportes periódicos de avance y gestión (A y D) Planificación de iteraciones (A) Planificación de entrega (A) Reuniones diarias (A) Retrospectiva (A) Espacio de trabajo informativo (A) Análisis causa-raíz (A y D)
En los Procesos de Apoyo	Verificación y validación continua (D) Gestión de la configuración del software (D) Aseguramiento de la calidad del software (D) Gestión de riesgos del proyecto (D)
En el Modelado de Negocios	Participación activa de interesados y usuarios (A) Modelado del dominio de la aplicación (D)
En la Ingeniería de Requisitos	Participación activa de interesados y usuarios (A) Requisitos incrementales (A) Gestión de requisitos (D)
En el Diseño Arquitectónico y Detallado	Diseño y arquitectura incremental (A y D) Diseño simple (A)
En la Programación e Integración	Desarrollo guiado por pruebas (A) Integración continua (A) Refactorización del código (A) Uso de estándares de codificación (D)
En las Pruebas de Software	Desarrollo guiado por pruebas (A) Retrospectiva (A)
En la Entrega de la Aplicación	Entrega por versión e incremento (A) Entrega periódica de software operacional (A)

Para seleccionar estas prácticas se hizo, primero, un análisis y comparación de las prácticas usadas en métodos ágiles y disciplinados (ver (Cabral, 2010)). Luego, se seleccionaron aquellas prácticas que cumplen con los siguientes criterios: (1) que sean utilizables por pequeños grupos de trabajo; (2) que sean efectivas y conocidas, para lo cual se usó el estudio realizado por Ambler (2006); (3) que su aplicabilidad en uno o más procesos del desarrollo de software sea efectiva. Una explicación de cada una de estas prácticas escapa del alcance de este artículo. La literatura sobre estas prácticas es muy extensa (ver, por ejemplo, Shore y Wanden (2008)).

5. ESTRUCTURA Y COMPONENTES DEL MARCO METODOLÓGICO BLUE WATCH

Blue WATCH está compuesto por tres modelos fundamentales:

- **El modelo de productos:** Describe los productos intermedios y finales que el marco metodológico propone elaborar durante el desarrollo de una aplicación empresarial.
- **El modelo de procesos:** Describe los procesos técnicos, gerenciales y de soporte que el grupo de trabajo debe seguir durante la ejecución de un proyecto de desarrollo.
- **El modelo de actores:** Identifica los roles necesarios para elaborar una aplicación empresarial y sus respectivas responsabilidades dentro del proceso de desarrollo de la aplicación. Este modelo propone, también, una estructura que sirve para organizar al grupo de trabajo que desarrollará la aplicación.

Cada modelo es un patrón metodológico extensible y adaptable. Las empresas o grupo de desarrollo, que estén interesadas en reutilizar el marco metodológico, deben adaptar cada modelo a sus necesidades y requisitos particulares.

5.1 EL MODELO DE PRODUCTOS

El modelo de productos es el primer componente del Blue WATCH. La importancia de este modelo radica en establecer que es lo que un grupo de trabajo debe producir a lo largo del proceso de desarrollo de una aplicación empresarial. Este modelo identifica, clasifica y describe los diferentes productos que, según el marco metodológico, deben producirse durante el desarrollo de una aplicación empresarial. Estos productos se clasifican en dos categorias: productos entregables y productos intermedios.

El principal producto entregable es, evidemente, la aplicación empresarial. Esta aplicación se entrega al cliente y a sus usuarios bajo la forma de versiones. Una aplicación puede constar de una o más versiones, las cuales se van elaborando y liberando a lo largo del tiempo de desarrollo de la aplicación. Cada versión es un producto acabado que entra en operación una vez entregado. Los usuarios utilizan cada versión para llevar a cabo sus actividades del negocio. Cada nueva versión agrega mayor funcionalidad a la aplicación y, en consecuencia, proporciona un mayor valor a los usuarios. Una versión se construye de manera incremental. Cada incremento es una pieza de software funcional que ejecuta una o más funciones requeridas por los usuarios. Los incrementos se producen en muy corto tiempo (1 – 3 semanas) y sirven para que los usuarios puedan validar partes de la versión en desarrollo.

La Tabla 3 identifica los productos intermedios que el marco Blue WATCH recomienda elaborar durante la ejecución de los procesos técnicos, de gestión y de soporte. Cada documento está compuesto de uno o más modelos que se elaboran usando las técnicas y herramientas que Blue WATCH recomienda para cada proceso técnico, de gestión o de soporte. Para reducir el esfuerzo requerido en la elaboración de estos documentos, muchos de ellos tienen plantillas asociadas y se generan con el apoyo de las herramientas CASE.

Tabla 3. Productos intermedios propuestos por el marco metodológico Blue WATCH

Productos Técnicos	Productos de Gestión	Productos de Soporte
Modelo del Negocio	Visión del Producto	Proceso y estándares de desarrollo
Documento de Requisitos	Plan del Proyecto	Plan de Validación y Verificación
Documento de Diseño	Planes de Iteraciones y Entregas	Plan de Gestión de Configuración

Especificaciones de Pruebas	Plan de Pruebas	Plan de Riesgos

5.2 EL MODELO DE PROCESOS

Para elaborar una aplicación empresarial, Blue WATCH utiliza la cadena de valor presentada en la figura 1. Los procesos de dicha cadena de valor se ejecutan cíclicamente de la manera señalada en la figura 3. Nótese que los procesos técnicos se ubican en anillo externo del ciclo, siguiendo la forma del dial de un reloj de pulsera; mientras que los procesos de gestión y soporte, están ubicados en el centro. Los procesos de gestión dirigen y controlan la ejecución de los procesos técnicos. Los procesos de soporte dan apoyo a la ejecución de los procesos técnicos y de gestión.

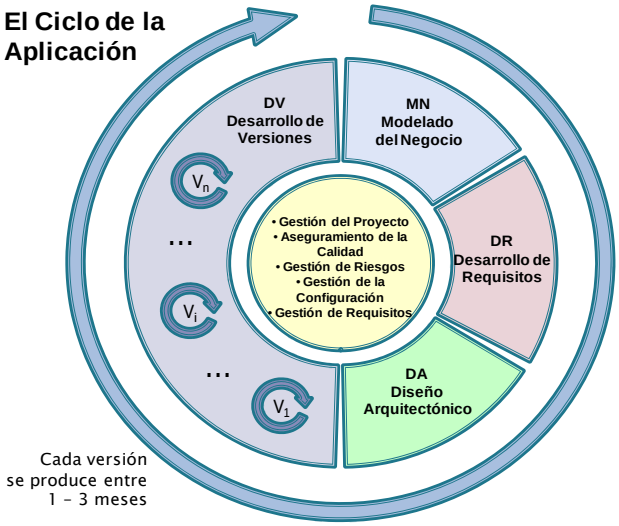

Figura 3. Procesos relacionados con el Ciclo de la Aplicación

El proceso de Desarrollo de Versiones es iterativo y consta de un conjunto de ciclos de versiones que se van desarrollando secuencialmente. El número de versiones depende de las características de cada proyecto y aplicación. Cada ciclo de versión ejecuta secuencialmente el conjunto de procesos técnicos que se señalan en la figura 4.a. Los cuatro primeros procesos refinan los productos elaborados previamente durante el ciclo de la aplicación.

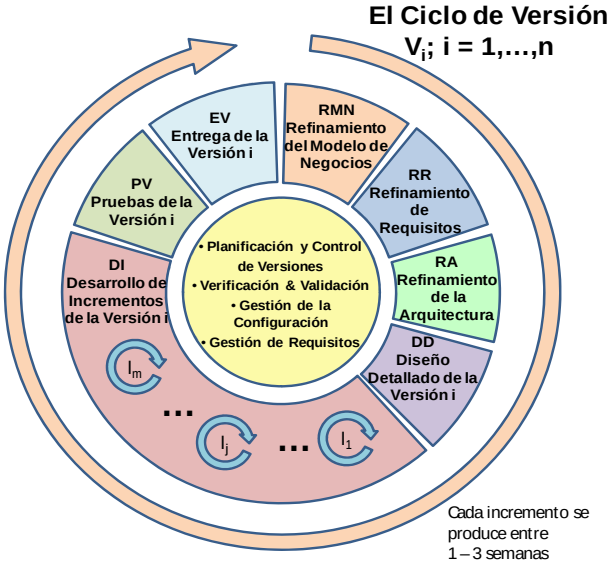

Figura 4.a. Procesos del Ciclo de Versión

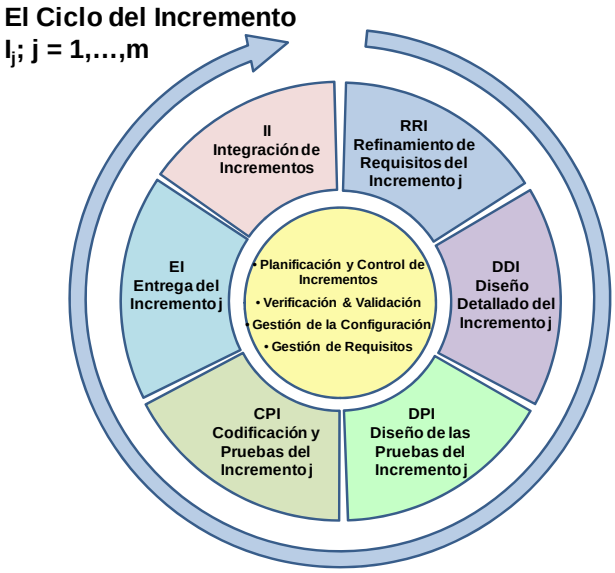

Figura 4.b. Procesos del Ciclo del Incremento

El proceso de Desarrollo de Incrementos, que se realiza en cada ciclo de versión, es también iterativo. Consta de un conjunto de ciclos de incrementos que se pueden desarrollar secuencialmente o en paralelo, dependiendo de la cantidad de programadores disponibles en el proyecto. La figura 4.b señala el orden en el cual se ejecutan los procesos técnicos encargados de elaborar un incremento.

5.3 EL MODELO DE ACTORES

El tercer componente del marco metodológico Blue WATCH plantea una estructura para organizar grupos de trabajo e identifica los roles que son necesarios para ejecutar los procesos descritos en el Modelo de Procesos. La figura 5 muestra los roles que deben integrar la estructura organizacional de un grupo de trabajo. La cantidad de personas que debe integrar la estructura de un grupo de trabajo depende del tamaño del proyecto y del personal disponible.

Los roles son distribuidos entre los miembros del grupo de trabajo. Es importante que esta asignación sea hecha en base a la formación, experiencia y competencias de las personas que forman el grupo. Cada rol tiene asociado un conjunto de responsabilidades vinculadas a las actividades de los diferentes procesos de desarrollo. Un mismo miembro del grupo de trabajo puede ejecutar, a lo largo del proyecto, varios roles. Por ejemplo, en grupos muy pequeños (2-4 personas), el líder del proyecto puede ejercer los roles de Líder del Proyecto, Gestor de Soporte y Analista.

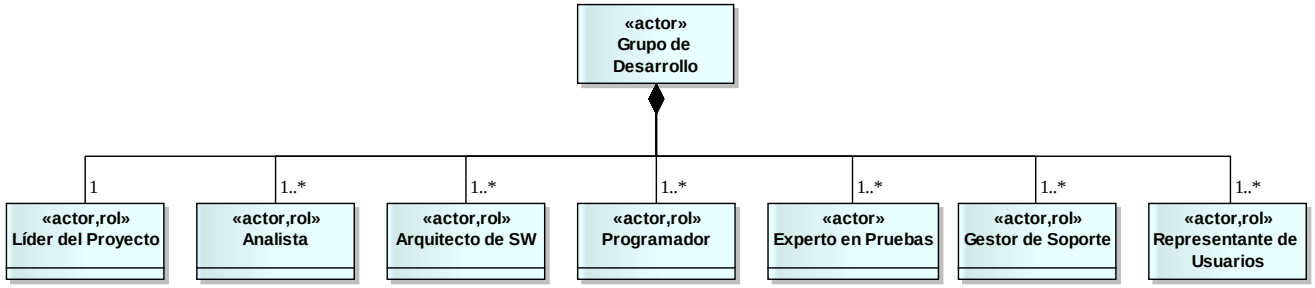

Figura 5. Roles requeridos por el método Blue WATCH

6. CONCLUSIONES

El desarrollo de aplicaciones empresariales es un proceso tecnológico complejo que requiere la participación de profesionales con competencias muy diversas. Por muy pequeña que sea la aplicación, la complejidad de su proceso de desarrollo no puede menospreciarse. Por otro lado, el tamaño de la empresa donde se desarrolla la aplicación influye significativamente en su proceso de desarrollo. En empresas pequeñas (PEDS), este proceso se ve afectado por la baja cantidad de recursos disponible y la presión para finalizar la aplicación a tiempo.

Blue WATCH es un patrón metodológico dirigido a las PEDS. Fue elaborado siguiendo el método de modelado de negocios BMM (*Business Modeling Method*) de Montilva y Barrios (2004). En su diseño, se buscaron los siguientes objetivos: (1) alcanzar un balance apropiado entre agilidad y disciplina; (2) reducir la documentación, que se produce durante el desarrollo de software, sin afectar el mantenimiento; y (3) simplificar e integrar los procesos técnicos, de gestión y de soporte para asegurar que las aplicaciones tengan la calidad establecida, se entreguen a tiempo y no excedan el presupuesto asignado.

Esta primera versión del marco metodológico Blue WATCH ha sido utilizada para elaborar un método de desarrollo de software detallado y adaptado a las características de una PEDS venezolana. La adaptación del marco metodológico y su aplicación en el desarrollo de dos aplicaciones empresariales, se describe en Cabral (2010). Como trabajo futuro está planteada la evaluación formal del marco metodológico. Una explicación detallada del método será incluida en el portal del proyecto Methodius (http://www.methodius.info.ve)

AGRADECIMIENTOS

Los autores agradecen el financiamiento proporcionado, para la realización de este trabajo, por el Fondo Nacional de Ciencia, Tecnología e Investigación (FONACIT) de Venezuela, bajo el proyecto No. 2005000165.

REFERENCIAS

Ambler, S. (2006). Agile Adoption Rate Survey Results: March 2006, http://www.ambysoft.com/surveys/agileMarch2006.html, 03/11/11. (date accessed)

Beck, K. et al. (2001). Manifesto for Agile Software Development, *Agile Alliance*, http://agilemanifesto.org/, 03/12/11. (date accessed)

Boehm, B. (2002). "Get Ready for Agile Methods, with Care". *Computer*, Vol. 35, No.1, pp. 64-69.

Boehm, B., and Turner, R. (2003*). Balancing Agility and Discipline: A Guide for the Perplexed*, Addison-Wesley.

Cabral, M.(2010). "Métodos de desarrollo de aplicaciones Web para PYMES", Tesis de Maestría, Universidad de Los Andes, Postgrado en Computación, Mérida, Venezuela.

Eriksson, H. E. and Penker, M. (2000). *Business Modeling with UML: Business Patterns at Work*, John Wiley & Sons.

Humphrey, W.S. (2006). *TSP – Leading a Development Team*, Addison-Wesley.

Krutchen, Ph. (2000). *The Rational Unified Process*, 2da edition, Addison-Wesley.

Laporte, C.Y., Alexandre, S. and Renault, A. (2008). "The Application of International Software Engineering Standards in Very Small Enterprises". *Software Quality Professional (SQP)*, Vol. 10, No. 3, pp. 4-11.

Mellor, S.J., Clark, A.N, and Futagami, T. (2003)."Model-Driven Development". *IEEE Software*, Vol. 20, No. 5, September/October.

Montilva, J., and Barrios, J. (2004). "BMM: A Business Modeling Method for Information Systems Development". *CLEI Electronic Journal*, Vol. 7, No. 3, December. Available at: http://www.clei.cl/cleiej/.

Montilva, J., Barrios, J., y Rivero, M. (2008). Gray WATCH: Método de desarrollo para aplicaciones empresariales (Proyecto Methodius), Informe técnico disponible en: http://www.methodius.info.ve, 12/03/11. (fecha de acceso)

Pino, F.J., Garcia, F., and Piattini, M. (2008). "Software process improvement in small and medium software enterprises: a systematic review". *Software Quality Journal*, Vol.16, pp. 237-261.

Rivero, D., Montilva, J., Granados, G., Barrios, J., Besembel, I. y Sandia, B. (2007). "La Industria de Software en Venezuela: Una Caracterización de su Recurso Humano", *Actas del X Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes de Software (IDEAS-EVETIS'2007)*. Isla de Margarita, Venezuela, pp. 435-443.

Rivero, D., Montilva, J., Barrios, J., y Murúa, M. (2009). "Un análisis del desarrollo de software en empresas venezolanas", *Actas de la Seventh LACCEI Latin American and Caribbean Conference for Engineering and Technology* (LACCEI'2009). San Cristobal, Venezuela, pp. WE1: 1-10.

Software Engineering Institute (2010). "CMMI for Development, Version 1.3". Carnegie Mellon University, USA, Technical Report No.CMU/SEI-2010-TR-033.

Shore, J., and Warden, S. (2008). *The Art of Agile Development*. O'Reilly, California, USA.

AUTORIZACION Y RENUNCIA DE RESPONSABILIDAD

Los autores autorizan a LACCEI para publicar el artículo en las actas de la conferencia. LACCEI y los editores no son responsables ni del contenido ni de las implicaciones de lo que está expresado en el artículo.