

QualDev Process: Procesos Adaptables de Desarrollo de Software para Proyectos Ágiles

Rubby Casallas Gutiérrez*
Hugo Fernando Arboleda Jiménez**

RESUMEN

Los Procesos que deben ser utilizados durante el desarrollo de software dependen altamente de las características del proyecto. Entre estas características se encuentran las del grupo de desarrollo y las del contexto dentro del cual se trabajará (los clientes, la tecnología, las restricciones, etc.). Denominamos proyecto ágil aquel en el que sus características de alto riesgo (alta rotación de personal, tecnología nueva, dispersión geográfica, requerimientos cambiantes, etc.) dificulta el prescribir las prácticas de software que deben ser seguidas. En este artículo presentamos una propuesta de procesos de software para este tipo de proyectos denominada *QualDev Process*. Los procesos sugeridos son adecuados en términos de mejorar

* Doctora en Informática, Especialista en Sistemas de Información en la Organización - Profesora - Universidad de los Andes, Bogotá, Colombia.
e-mail:rcasalla@uniandes.edu.co

Magíster Ingeniería de Sistemas y Computación, Ingeniero **de Sistemas - Profesor, Facultad de Ingeniería, Universidad de San Buenaventura, Santiago de Cali, Colombia.
e-mail:huarbole@usb.edu.co

Fecha de recepción: Septiembre 30 de 2004
Fecha de aprobación: Diciembre 17 de 2004

la calidad del producto, la productividad del equipo y la facilidad para realizar mantenimiento a los productos de software construidos. *QualDev Process* está basado en la idea de adaptabilidad de los procesos y de mejoramiento continuo. El resultado final es una mezcla de las propuestas de los procesos livianos (*lightweight process*) [7] y de procesos más definidos como TSP [5] o CMMI [1]. La propuesta, además de incluir el qué hacer incluye guías metodológicas de cómo llevar a cabo las prácticas. Este proceso se ha estado aplicando en los últimos dos años en algunos desarrollos de proyectos ágiles, lo que nos ha permitido validar las prácticas.

Palabras Clave: Proceso de software, Calidad, Adaptación de procesos, Contextos de desarrollo, Modelo de ciclo de vida de desarrollo, Plan de implantación, Mejora continua, Procesos ágiles.

ABSTRACT

The processes to be used during a software development project highly depend on the characteristics of the project. Among the characteristics we have those of the team project and those of the context in which the project is going to be carried out (clients, technology, constraints, etc.) We named agile project a high risk project with characteristics like an elevated labor turnover, new technology, a team geographically disperse, changing requirements, etc. On these kind of project is very difficult to prescribe the software engineering practices to be followed. In this paper we present a proposal of software processes called *QualDev Process*. The suggested processes in the proposal are suitable to improve the product quality, team productivity and to facilitate the maintenance of the product. The main ideas of the *QualDev Process* is the adaptability of the process and the continual improvement. The final result is a mixed of the called *lightweight* [7] process and those more defined like TSP [5] or CMMI [1]. The proposal includes besides of the practices to take into account to follow, a methodological way to put

them in practice. This process has been used during the last two years in some software developments in agile projects, it has allowed us to validate the proposed practices.

Key Words: Software Process, Quality, Process Adaptability, Life Cycle Models, Improvement Plan, Agile process.

1. INTRODUCCIÓN

A nivel mundial se manejan presupuestos elevados para desarrollo de software. Gran parte de estos presupuestos se gasta en desfases, cancelación de proyectos y mantenimiento del software. Requerimientos mal implementados, retrasos de cronogramas, baja calidad del producto y difícil mantenimiento del software son problemas recurrentes en los proyectos de desarrollo. Entre otras causas, se debe a la mala o ninguna administración del proceso de desarrollo, que lleva a ignorar las características del proyecto en el momento de escoger las mejores prácticas que se llevarán a cabo en cada etapa. Dentro de estas características se encuentran por ejemplo la tecnología de desarrollo y el recurso humano.

Algunas respuestas a estos problemas son iniciativa del área de Procesos de Software. Es así como han surgido propuestas como el CMM [11], CMMI [1], SPICE [12], y TSP [5] entre otros. Estas propuestas tienen en común la idea fundamental de *Mejoramiento Continuo*. Sin embargo, para ciertos tipos de organizaciones y ciertos tipos de proyectos, no es viable implantar estos modelos debido a su complejidad, a la cantidad de prácticas sugeridas, a la rigidez de las exigencias, a los costos de acompañamiento, etc.

De otro lado, las propuestas de proyectos ágiles (*lightweight process*) como las presentadas en [2] [3] [9] [10] [13], entre otras, hacen énfasis en la adaptabilidad del proceso. De manera general, promueven principios que en ocasiones incluyen prácticas *paperless* y/o sin

administración de métricas. Dado que no se espera que un proceso sea predecible, se trabaja más en la preparación para adaptarse al cambio, que en la oportunidad de capitalizar lo aprendido para que el impacto del cambio sea mínimo.

Cualquiera que sea la iniciativa de proceso definido que un equipo de desarrollo siga, de manera natural, una organización madura gradualmente al ritmo de sus propias necesidades. La maduración de los procesos incluye la inclusión de prácticas al ciclo de vida de desarrollo. Si las prácticas resultan efectivas y los procesos le son útiles, los equipos las institucionalizan y mejoran.

En este artículo presentamos *QualDev Process*, una propuesta de procesos de desarrollo de software para proyectos que denominamos ágiles. Las características de estos proyectos son principalmente el tamaño pequeño de los grupos de desarrollo, no mucha experiencia con la tecnología, rotación alta de personal y dispersión geográfica de todo o parte del grupo de trabajo.

Nuestra propuesta también está basada en la idea del mejoramiento continuo. No predefine un orden estricto sobre los procesos, ni espera que el resultado de ellos sea completamente predecible. Por el contrario, a medida que el grupo o la organización va madurando, se van adaptando los procesos y generando habilidad para responder al cambio. El mejoramiento continuo, aunque consiste concretamente en la adaptabilidad de los procesos (principio cardinal de *lightweight process*), se fundamenta en la administración de métricas de desarrollo que permiten el aprendizaje y maduración de los procesos.

La organización del artículo es la siguiente. En la segunda parte presentamos una descripción del contexto de los proyectos y de los grupos a los que les queremos dar soporte. En la tercera parte presentamos el marco de referencia de los procesos, los principios, la organización de los

procesos sugeridos y la metodología para la documentación de estos. De igual forma, discutimos brevemente algunos puntos de implementación de los procesos propuestos (el cómo hacerlo). Por último, algunos resultados alcanzados hasta hoy con un grupo de desarrollo particular, conclusiones y trabajos futuros.

2. CONTEXTO DE PROYECTOS ÁGILES

En principio consideramos que no es posible esperar tener un único modelo de desarrollo adecuado para todos los contextos de desarrollo. Los proyectos de software pueden presentar entre sus características, equipos de trabajo no homogéneos, alta rotación del personal, poca disponibilidad de los recursos (dispersión geográfica), tecnología nueva, tamaño variable del equipo de desarrollo, diferencia entre los tamaños y naturaleza del software que se va a desarrollar o a mantener, entre otras.

Hemos definido un contexto de desarrollo para proyectos ágiles cuyas características son presentadas en la Tabla 1.

Tabla 1. Características de proyectos ágiles

Característica		Nivel
Experiencia del equipo	En la tecnología de desarrollo	Baja/Media
	En la lógica de negocio de desarrollo	Baja/Media
	En investigación y desarrollo	Baja
	En procesos de desarrollo	Baja
Métricas de velocidad de desarrollo conocidas y existencia de datos históricos		Baja
Rotación del equipo		Alta
Condiciones apropiadas para la comunicación del equipo		Baja

Las características de los proyectos ágiles corresponden a compañías de desarrollo de software jóvenes o a grupos que se conforman para desarrollar algún proyecto específico. Por compañías jóvenes nos referimos a compañías con menos de un año de permanencia en la industria, con menos de 2 proyectos de

desarrollo, con trabajo en tecnologías de punta y con procesos inmaduros de desarrollo de software.

Para el caso de los proyectos ágiles, se requieren prácticas concretas que en otros contextos pueden no tener tanta importancia. A continuación se presenta unos casos de ejemplo.

Experiencia en tecnología de desarrollo:

Ante la baja experiencia en tecnología, los equipos requieren un mayor esfuerzo en el área de investigación y desarrollo. Los resultados se deben ver reflejados en prototipos o soluciones *spike* que apoyen la labor de aprendizaje en la tecnología. De manera paralela se requiere invertir en el proceso de administración de conocimiento, de manera que la curva de aprendizaje de todo el equipo se minimice.

Experiencia en lógica del negocio:

Ingresar en el mercado de nuevos negocios es fundamental para el crecimiento económico de un equipo de desarrollo con proyectos ágiles. La pérdida de oportunidades de desarrollo es crítica cuando un equipo de desarrollo está empezando a ingresar en el mercado. Cuando no se tiene experiencia en la lógica de negocio para la cual se desarrolla, es necesario hacer énfasis en la validación del análisis de requerimientos y en general en las pruebas de validación. Esto requiere un contacto mucho más cercano con el cliente. Si al contrario se tiene amplia experiencia en la lógica del negocio, el contacto con el cliente puede ser menor, minimizando el riesgo de realizar una inadecuada especificación de requerimientos.

Experiencia en procesos y conocimiento de métricas de desarrollo:

La experiencia para seguir y adaptar procesos de desarrollo dentro de un proyecto es fundamental para maximizar la productividad de un equipo. Liderar un proceso de comprensión y adaptación de un modelo de ciclo de vida de desarrollo es indispensable dentro de un equipo sin mucha

experiencia. Por lo mismo, la no existencia de datos históricos conlleva a mala estimación y por consiguiente, repercute en atrasos dentro de los proyectos. En el contexto de desarrollo que se define, el equipo debe estar dispuesto a obtener rápidamente datos que permitan una mejor estimación. Si la experiencia en la tecnología de desarrollo es alta, no implica que se cuente con datos históricos formales. En cambio, si no se tiene experiencia es segura la ausencia de estos datos.

Rotación del equipo:

La alta rotación del equipo se puede dar a varios niveles. Compañías con proyectos de desarrollo grandes pierden muchos de sus desarrolladores en el transcurso de sus proyectos. Compañías con desarrollos pequeños pierden parte de sus desarrolladores al terminar sus proyectos. Esto implica la búsqueda de nuevos desarrolladores cada vez que se inicia un proyecto. No importa cuál sea la causa, la rotación del equipo es una característica determinante para el éxito de un proyecto de desarrollo. Los equipos que presentan esta característica deben implementar procesos adecuados de administración de conocimiento y, en general, mantenimiento de documentación para soportar la pérdida continua de sus integrantes.

Comunicación del equipo (dispersión física de sus integrantes):

La comunicación de un equipo de desarrollo influye directamente en el éxito de los proyectos. Ante una pobre comunicación o sincronización de ésta, el equipo se puede encontrar ante una crisis de trabajo aislado y particular por parte de cada integrante, sin lograr conseguir una unidad de grupo que apoye la labor de conseguir un último fin común. Un equipo de desarrollo sin condiciones apropiadas de comunicación debe implantar procesos que cambien esta condición. La administración de configuración y el seguimiento al proceso de documentación y comunicación son los procesos encargados de trabajar en esta área.

Como se puede ver, prácticas que pueden ser innecesarias en su rigurosidad, se hacen fundamentales en un contexto definido como el de proyectos ágiles. En un proyecto ágil, muchas veces se cree que el proceso de software en vez de ser una herramienta de apoyo al desarrollo, es una carga extra para los participantes. Esto puede ser cierto si el enfoque para incluirlo es prescriptivo e incluye muchas prácticas a la vez. Con esto sólo se logra que las prácticas se dejen de lado rápidamente ya que no se ve su utilidad y los desarrolladores sienten que por seguir el proceso, no desarrollan el producto.

Nosotros creemos que aún en contextos difíciles de implantación de procesos como los proyectos ágiles, la tarea se puede lograr si se utiliza un enfoque adaptativo y se deja que la misma organización vaya haciendo gradualmente sus ajustes e incluyendo más prácticas a medida que madura. Incluir sistemáticamente prácticas de proceso de las cuales no se obtendrá un beneficio a corto o mediano plazo, impone resistencia en el interior del grupo. Las prácticas se deben incluir en la medida que el grupo las discuta y considere que ofrecen un valor agregado. Así, luego de la inclusión de cada práctica, ésta es adaptada por el equipo en la medida de la necesidad vista. Cada práctica puede madurar de manera independiente de las demás.

A diferencia de propuestas como la presentada por eXtreme Programming (XP) [3], nosotros pensamos que ciertas prácticas de proceso no son adecuadas para todos los contextos, incluso después de que son adaptadas. Un ejemplo puede ser la programación en parejas o el trabajo en grupos. Dentro del contexto de proyectos ágiles los desarrolladores se encuentran dispersos la mayor parte del tiempo así que realizar reuniones de seguimiento con la presencia física de todos los integrantes del equipo ya es un logro importante. Para este caso particular, se deben buscar prácticas diferentes que apoyen la calidad del producto, sin imponer condiciones que por naturaleza del equipo son

difíciles de lograr. Procesos como TSP [5] pretenden ser utilizados en cualquier contexto de desarrollo. De acuerdo a su propuesta, las prácticas deben ser incluidas indistintamente del criterio del equipo. Aunque luego promueven su adaptación, algunos equipos no requieren ciertas prácticas por su contexto de desarrollo. Peor aún, algunos equipos que sí las requieren encuentran imposible incluirlas de acuerdo a la especificación de la propuesta debido a su complejidad y rigurosidad.

Un punto en el que se debe hacer énfasis ya que aplica en cualquier contexto, es la importancia de usar herramientas de apoyo al proceso. Cualquiera que sea la práctica de proceso incluida, se deben buscar herramientas que apoyen el seguimiento de esta práctica, ayuden a su adaptación y así, promuevan la mejora continua de la práctica a la vez que descarguen trabajo operativo manual a los integrantes del equipo.

Estas ideas han sido incluidas en el marco de procesos que hemos denominado *QualDev* y que presentamos en la sección siguiente.

3. *QualDev* PROCESS

QualDev Process es un marco de referencia de procesos de desarrollo para proyectos ágiles en donde el proyecto se desarrolla en ciclos iterativos en los que el producto se construye incrementalmente. *QualDev process* está compuesto por varios elementos. El primer elemento son los principios, el segundo la organización de las prácticas, el tercero la metodología para implementar cada práctica y por último un plan de implantación sugerido. En esta sección estudiamos los dos primeros elementos y brevemente presentamos la estructura propuesta para documentar las prácticas de implementación de los procesos. El plan de implantación está fuera del alcance de este documento.

3.1 Principios

El proceso de software QualDev gira alrededor de cinco principios sobre los que se organizan las prácticas y el mejoramiento continuo. Estos son: (1) Deben existir prácticas de planeación y seguimiento permanentes. El realizar de manera consistente esta práctica es lo que va a permitir que el grupo de desarrollo crezca y pueda adaptar el proceso a sus necesidades. Alrededor del principio de planeación y seguimiento se encuentran tres más: (2) Administración del riesgo, (3) Administración de la calidad, y (4) Administración de conocimiento. El principio de administración de conocimiento es el que potencia que los resultados obtenidos con respecto a las prácticas sean duraderos. El quinto (5) principio hace referencia al uso de herramientas para apoyo de los procesos de desarrollo. La Figura 1 ilustra la relación entre los principios QualDev.

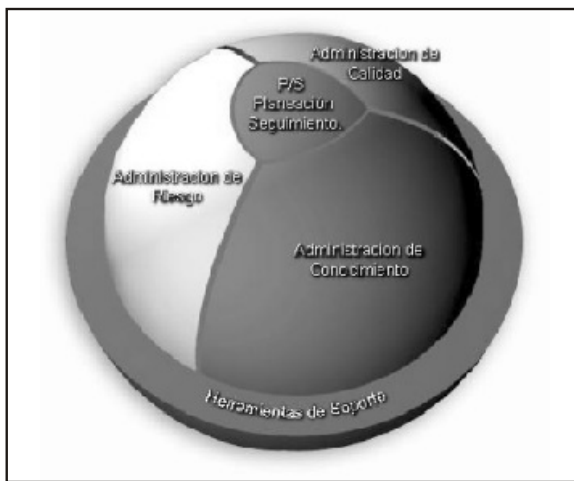


Figura 1. Principios de QualDev y su relación.
1. Planeación y seguimiento.

a. Planeación y Seguimiento: La planeación de un proyecto de software debe realizarse a lo largo de todo el proyecto. Debido a diferentes factores (internos y externos), no es posible predecir los cambios que sufrirá el plan. Sin embargo, se debe realizar planeación como la base para retroalimentar el proceso de recolección de métricas que permitan realizar mejores

estimaciones. Cualquiera que sea el modelo de ciclo de vida de desarrollo escogido, los planes deben ser revisados continuamente de manera que se adapten a las necesidades del proyecto. El seguimiento realizado al proyecto tiene como resultado el establecimiento de prácticas que redundan en la adaptabilidad y el mejoramiento continuo.

En el tema de planeación un factor determinante son los ciclos de desarrollo. Estos ciclos deben ser cortos e iterativos, y al final de cada uno debe estar terminado un sistema funcionalmente completo. No se deben generar prototipos funcionales que no sean con fines concretos de aprendizaje al interior del equipo o como borradores de interfaz de usuario.

Es importante definir al inicio de cada ciclo de desarrollo un conjunto de objetivos. Estos deben estar acompañados de métricas que permitan evaluar el alcance logrado al final del ciclo. De igual forma, todas las actividades planeadas deben tener compromisos definidos y cuantificables.

b. Administración de Riesgos: El riesgo es el componente fundamental en la definición del proceso que se establece. En el proceso de QualDev se establece que realizar las actividades básicas de administración del riesgo, desde antes de iniciar el proyecto, es fundamental para determinar el conjunto de procesos que deben ser seguidos. Listar los riesgos potenciales del proyecto, clasificarlos y priorizarlos permite al equipo pensar en las prácticas más adecuadas para minimizar la ocurrencia de estos, o el impacto ante una ocurrencia. La clasificación de los riesgos se debe realizar con base en una taxonomía definida, por ejemplo la presentada en [4].

Este principio es análogo a un proceso de administración preventiva del riesgo que permite establecer planes que en este caso resultan en procesos. El seguimiento continuo de los

procesos llevados a cabo, planeando la introducción de nuevas prácticas con base en los riesgos que el equipo encuentra como potenciales, permite que los procesos se introduzcan por motivación propia del equipo ante la identificación de la necesidad.

c. Administración del conocimiento: El conocimiento es un recurso intangible sobre el que se deben establecer métricas que permitan administrarlo. Todo el conocimiento traído de fuera, construido o adquirido por el equipo debe ser valorado como uno de los activos más valiosos y administrado de acuerdo a una estrategia establecida. El equipo lo debe administrar de manera que se convierta en un elemento diferenciador a nivel interno que permita mejorar la productividad y calidad del trabajo y a nivel externo como una ventaja competitiva.

Uno de los elementos más importantes que se basan en la administración de conocimientos, es la definición de los roles al interior del equipo de desarrollo. A diferencia de lo propuesto por TSP [5], nosotros creemos que no es suficiente definir roles que de manera general apunten a la parte administrativa del proyecto. Es necesario que al interior del equipo existan líderes claramente definidos e identificados por cada área de conocimiento, en especial las que se han identificado como generadoras de la mayor cantidad de riesgos potenciales. Cada líder no administrativo debe ser el responsable de administrar una base de datos de conocimiento que permita a todo el equipo llegar a ser más productivo. Por ejemplo, puede ser necesario definir un rol de pruebas ante la necesidad de modificar el proceso de pruebas del software o la inclusión de nuevas herramientas de soporte al proceso.

d. Administración de la calidad: La administración de la calidad no puede dejar de ser uno de los principios fundamentales para el desarrollo de software. Se deben definir

estrategias para que a lo largo del ciclo de desarrollo se pueda controlar la calidad de los productos que se construyen. De ninguna manera se debe limitar la administración de calidad a realizar pruebas de unidad, funcionales o del sistema. En cambio de esto, la estrategia de calidad se debe basar en las inspecciones realizadas a los productos intermedios de trabajo. Aunque realizar inspecciones a todos los productos de trabajo sería ideal para garantizar altos niveles de calidad, el costo de esto es tan elevado que no sería posible mantener algún tipo de ventaja competitiva basada en la productividad del equipo. Debe existir un balance en la cantidad de inspecciones que se realicen, la cantidad de estándares e instructivos que se definan y las pruebas realizadas al software construido.

Como se ha mencionado, los cuatro principios presentados son la base para la definición de los procesos de *QualDev*. De igual forma, son la base para realizar un plan de implantación y de adaptación al contexto propio de desarrollo de un equipo de trabajo. De manera adicional, es importante resaltar la importancia de la definición y documentación del cómo debe ser llevado a cabo cada proceso/actividad del ciclo de vida de desarrollo.

e. Soportar los procesos con herramientas: Los procesos de software deben ser soportados sobre herramientas que apoyen su desarrollo. Es posible llevar algunos procesos de manera manual sin un sistema de información que ayude a su administración. Sin embargo, algunos procesos que no se fundamenten en una herramienta son imposibles de llevar a cabo. De igual forma, los que se pueden realizar sin herramientas pueden ser mejor explotados si se define una herramienta para realizar su administración y gestión. Las herramientas de soporte al proceso deben ser introducidas de manera incremental y por iniciativa del equipo. De la misma manera que el proceso, las herramientas deben ser adaptadas a lo largo de los ciclos de desarrollo para cubrir las necesidades del equipo.

3.2 Organización de las prácticas

Los procesos se han dividido en dos grupos: soporte y desarrollo. Los procesos de soporte apoyan el desarrollo a lo largo de todo el ciclo de vida de la construcción del producto. Los procesos de desarrollo se refieren a actividades enfocadas a la producción de artefactos que son parte integral del producto. La Tabla 2 resume los grupos de procesos propuestos.

Tabla 2. Grupos de procesos del marco de referencia para proyectos ágiles

Grupo de procesos	Procesos	Subprocesos
Soporte	1 Planeación	
	2 Administración de la configuración	
	3 Verificación	1. Inspección 2. Pruebas de verificación
	4 Administración de riesgos	
	5 Administración de conocimiento	1. Administración de base de datos de conocimientos 2. Investigación y auto capacitación
	6 Seguimiento al proceso	1. Documentación 2. Comunicación
	7 Estabilización del ambiente de desarrollo	
Desarrollo	1 Lanzamiento	
	2 Estrategia	
	3 Análisis	
	4 Diseño	
	5 Implementación	
	6 Pruebas de validación	
	7 Implantación	
	8 Posmortem	

3.3 La metodología de cada práctica

Las prácticas de los procesos de soporte y desarrollo van más allá de qué hacer y van acompañadas del cómo hacerlo. Esto lo hacemos como guías de implantación y metodología asociada a la actividad. Las guías de implantación incluyen las entradas y salidas de cada actividad, junto con las herramientas y documentos de soporte que deben ser utilizados. La metodología incluye lista de tareas y orden de ejecución de éstas al interior de cada actividad.

La definición y documentación de cada actividad se presenta en el siguiente orden:

1. Descripción del proceso.
 - a. Motivación para llevarlo a cabo
2. Descripción de las actividades asociadas
 - a. Justificación
 - b. Propósito
 - c. Guías de implantación
 - d. Metodología (herramientas y documentos de soporte).

A manera de ilustración presentamos algunos apartes de la metodología para la práctica de planeación y seguimiento. Por razones de espacio no la podemos presentar en su totalidad. Esta práctica incluye una justificación, un proceso propuesto y para cada actividad del proceso propuesto una descripción de cómo llevarla a cabo. La Tabla 3 muestra las actividades del proceso propuesto:

Tabla 3. Actividades asociadas al proceso de planeación

Actividades del proceso propuesto para Planeación y Seguimiento	
1	Estimación de casos de uso.
2	Planeación de ciclos de desarrollo.
3	Reunión de seguimiento de planeación
4	Reporte de tiempo invertido.
5	Consolidación de reportes de tiempo.
6	Mantenimiento de métricas.

Presentamos a continuación la actividad “Reunión de seguimiento de planeación”.

Justificación: Dada la característica de dispersión física del equipo, la reunión de seguimiento de planeación es uno de los pocos espacios que tiene el equipo para reunirse y retroalimentarse. Dentro de la reunión, la comunicación entre todos los integrantes del equipo es directa y en persona. Este tipo de comunicación es importante para generar un ambiente de equipo de trabajo.

La revisión de la planeación dentro de la reunión, con aporte de todo el equipo, ayuda a generar un ambiente en el que todos tienen voto en el plan que se crea. Las reuniones de seguimiento permiten que todos los integrantes del equipo conozcan el resumen de los inconvenientes que se han presentado en el periodo de desarrollo, aportando ideas para solucionarlos. Por otro lado, permite conocer el avance general del proyecto y aportar ideas para lograr cumplir con el objetivo general de desarrollo.

Propósito: Las reuniones de seguimiento de planeación se llevan a cabo al inicio de todo período de desarrollo. Estas reuniones, preferiblemente presenciales, tienen como objetivo hacer un seguimiento de la planeación realizada en la anterior reunión de seguimiento. Dentro de la reunión se define el plan de actividades hasta la siguiente reunión de seguimiento de planeación. Basados en el plan de actividades, los desarrolladores trabajan reportando sus tiempos de desarrollo para llevar registro del tiempo invertido en el ciclo. Al momento de llegar a la reunión, los reportes de tiempo invertido han sido consolidados, lo que permite conocer el estado general de las actividades planeadas en la reunión anterior y planear las actividades hasta la siguiente reunión.

La presencia física de todos los integrantes del equipo en la reunión de seguimiento es importante. Sin

embargo, dadas las características de los proyectos ágiles, se puede presentar la ausencia de algún integrante. El impacto de esta eventualidad se minimiza reportando todos los resultados de la reunión y publicándolos para conocimiento de todo el equipo. Además, se espera que de ser posible, el integrante del equipo que falta a la reunión reporte su ausencia con anterioridad y envíe un documento con los puntos que tiene para exponer en la misma.

Guías de implementación: La Tabla 4 presenta las entradas, salidas y formatos e instructivos asociados a esta actividad. Como anexo a este documento se presenta una ampliación de cómo lograr la transformación.

4. RESULTADOS

El proceso presentado en este artículo está siendo utilizado en varios proyectos de desarrollo

Tabla 4. Documentos asociados a la actividad de reunión de seguimiento de planeación

Entradas	Salidas	Documentos de soporte asociados
1 Documento de plan de trabajo del periodo de desarrollo inmediatamente anterior. En el primer ciclo no existe.	1 Documento de agenda de reunión.	1. Formato de desarrollo de agenda a tratar en la reunión.
2 Documento del consolidado de tiempo invertido por el equipo en el periodo de desarrollo inmediatamente anterior.	2 Documento bitácora de reunión.	2. Instructivo de llenado de formato de desarrollo de agenda a tratar en la reunión.
3 Formatos weekly de cada integrante del equipo.	3 Documento plan de trabajo del periodo. Este documento debe ser digitalizado en la herramienta de soporte para planeación. Se recomienda usar dotProject.	3. Formato del plan del periodo de desarrollo. 4. Instructivo de llenado del plan del periodo de desarrollo.
4 Documento de estimación de casos de uso para el ciclo.	4 Modificaciones necesarias al documento de planeación del ciclo de desarrollo.	5. Formato de reporte de informe de modificación de plan para el cliente.
5 Documento de planeación del ciclo de desarrollo.	5 Documento reporte de informe de modificación de plan para el cliente.	6. Instructivo de reporte de informe de modificación de plan para el cliente.

de software al interior de la Universidad de los Andes en Bogotá, Colombia. Estos proyectos cumplen las características de los proyectos ágiles que presentamos previamente. Uno de estos proyectos se llama *Changeset* y su propósito es construir un software de apoyo a la administración de configuraciones de software. Durante casi dos años hemos hecho parte del equipo más de 20 personas con un promedio de 5 al mismo tiempo. El grupo se conforma con estudiantes de pregrado, de maestría y por profesores, ninguno de ellos trabajando tiempo completo en el proyecto ni compartiendo un espacio de trabajo común. De forma paralela, los productos de software desarrollados por el equipo, incorporan tecnología de punta con alto nivel de calidad. Estos productos son desarrollados para ser puestos en producción en la industria, logrando una transferencia de conocimiento a nivel tecnológico y como herramientas que apoyan el proceso de desarrollo de software.

En general, los resultados de las experiencias recopiladas dentro de *QualDev* han sido la base para estructurar esta propuesta. Igualmente, de manera inversa, sin la propuesta del marco de referencia no habría sido posible el trabajo ordenado y productivo del equipo.

Las experiencias adquiridas y el software construido, tienen como objetivo ser transferidos a organizaciones con contextos similares. De esta forma se inician procesos de adaptación que permiten validar la metodología de desarrollo. De manera paralela, permite proyectar la experticia ganada para que sea útil a organizaciones con contextos de desarrollo diferentes.

5. CONCLUSIONES Y TRABAJOS FUTUROS

No es posible esperar tener un único modelo de desarrollo adecuado para todos los contextos de desarrollo. Por ejemplo, han sido estudiadas por [14] algunas de las limitaciones de los procesos

ágiles y, por supuesto, se entiende que éstos no son la solución para todos los contextos. Por esta razón, antes de proponer las prácticas específicas de desarrollo que se quiere implantar en un equipo, se debe definir su contexto y buscar las prácticas más adecuadas a éste.

Toda propuesta de procesos de desarrollo debe tener además de la definición y documentación de los procesos que la componen (el qué), el conjunto de metodologías (el cómo) deben ser llevadas a cabo. Dentro de la propuesta presentamos un conjunto de procesos de desarrollo definidos y documentados, acompañados de un conjunto de principios que deben ser retomados por equipos de desarrollo con proyectos ágiles para adaptar su modelo de desarrollo.

La adaptación de los procesos de desarrollo es indispensable para lograr el mejoramiento continuo y para que se adecuen al contexto de desarrollo. De igual forma, el uso de herramientas de apoyo a los procesos es fundamental en la inclusión de prácticas de desarrollo que lleva a cabo un equipo.

Consideramos que esta propuesta debe estar acompañada de una guía de implantación del proceso que ayude a las organizaciones a introducir prácticas de procesos en sus ciclos de vida de desarrollo. En este momento estamos validando en varios proyectos un plan de implantación y esperamos tener resultados concretos en el mediano plazo.

6. BIBLIOGRAFÍA

- [1] Chrissis, Mary Beth. Konrad, Mike. Shrum, Sandy. CMMI®: Guidelines for Process Integration and Product Improvement; Addison Wesley Professional; The SEI Series in Software Engineering. 2003. ISBN: 0321154967
- [2] Beck, Kent. Extreme Programming Explained: Embrace Change Addison-Wesley

- Professional. 1999. ISBN: 0201616416
- [3] Beck, Kent. Fowler, Martin. Planning Extreme Programming. Addison-Wesley Professional. 2000. ISBN: 0201710919
- [4] Carr , M. Kondra, S. Monarch, I. Ulrich, F. Walker, C. Taxonomy-Based Risk Identification. Technical Report. CMU/SEI-93-TR-006
- [5] Humphrey, Watts. "Introduction to the team software process". Addison-Wesley Professional. 1999. ISBN: 020147719X
- [6] IEEE Standard for Developing Software Life Cycle Processes Document Number: IEEE 1074-1997. ISBN: 1559379936.
- [7] Manifiesto para desarrollo de software ágil. Descargado de: <http://agilemanifesto.org>. Consultado en mayo 2004.
- [8] Metodologías Crystal. Descargado de: <http://alistair.cockburn.us>. Consultado en: abril 2004
- [9] Palmer, Stephen. Felsing, John. A Practical Guide to Feature-Driven Development (The Coad Series).Prentice Hall. 2002. ISBN: 0130676152
- [10] Schwaber, Ken. Beedle, Mike. Agile Software Development with SCRUM Prentice Hall. 2001. ISBN: 0130676349
- [11] Software Engineering Institute. Carnegie Mellon University. CMM - Capability Maturity Model®. Addison-Wesley Professional; 1995; ISBN: 0201546647
- [12] SPICE - Software Process Improvement and Capability dEtermination. Descargado de: <http://www.sqi.gu.edu.au/spice/>. Consultado en: enero 2004
- [13] Stapleton, Jennifer. Constable, Peter. DSDM. Addison-Wesley Professional.1997. ISBN: 0201178893
- [14] Turk, Dan. France, Robert. Rumpe, Bernhard. Limitations of Agile Software Processes. Technical Report. Colorado State University, 2002.