

PEDECIBA Informática
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

Reporte Técnico RT 09-13

Una metodología para desarrollo de videojuegos: versión extendida

Nicolás Acerenza Ariel Coppes Gustavo Mesa
Alejandro Viera Eduardo Fernández
Tomás Laurenzo Diego Vallespir

2009

Acerenza, Nicolás; Coppes, Ariel; Viera, Alejandro; Fernández, Eduardo; Laurenzo, Tomás; Vallespir, Diego.
Una metodología para desarrollo de videojuegos: versión extendida

ISSN 0797-6410

Reporte Técnico RT 09-13

PEDECIBA

Instituto de Computación – Facultad de Ingeniería

Universidad de la República

Montevideo, Uruguay, 2009

Una Metodología para Desarrollo de Videojuegos

Versión Extendida

Nicolás Acerenza, Ariel Coppes, Gustavo Mesa, Alejandro Viera
Eduardo Fernández, Tomás Laurenzo, y Diego Vallespir

Instituto de Computación - Facultad de Ingeniería
Universidad de la República, Uruguay
{nicoace, ariel.coppes, gmhaisburu, aleviera6}@gmail.com
{eduardof, laurenzo, dvallesp}@fing.edu.uy

Resumen. Tras relevar las empresas que desarrollan videojuegos en Uruguay, se detecta que son pequeñas en infraestructura, que abarcan generalmente proyectos de corta duración con equipos reducidos y que no cuentan con una metodología para desarrollo formalizada. Las metodologías que utilizan siguen principios de las metodologías ágiles que se adaptan con éxito para el desarrollo de videojuegos a nivel mundial y aplican a realidades similares. En particular se registran casos de éxito con adaptaciones de *Scrum* y *XP*, aunque éstas tampoco se encuentran formalizadas. Este artículo define y especifica SUM, una metodología para el desarrollo de videojuegos que se adapta a las características de la industria en Uruguay y sigue los principios ágiles, utilizando *Scrum* y *XP* como base de la propuesta.

Palabras clave: procesos de desarrollo de software, ingeniería de software empírica, videojuegos, metodologías ágiles.

1 Introducción

Dos de los problemas que atentan contra el crecimiento de la industria de videojuegos en Uruguay son la falta de recursos económicos y de una formación específica en el área. Este último aspecto es mayoritariamente informal y se basa en la experiencia personal de los miembros de la industria. Como consecuencia existen dificultades para transmitir estos conocimientos, tanto entre pares como a nuevos desarrolladores.

El presente trabajo tiene como objetivo definir una metodología para el desarrollo de videojuegos -que llamaremos SUM- la cuál busca adaptarse a la realidad del Uruguay y hacer un aporte al desarrollo de su industria. En particular se toman *Scrum* y *XP* como base de SUM por la existencia de casos de éxito y los beneficios que reportan para desarrollo de videojuegos. Además, la actual utilización de algunos de sus principios en la industria local facilitan su adopción.

El trabajo se estructura de la siguiente forma: en la sección 2 se presenta el estado del arte de las metodologías ágiles para el desarrollo de videojuegos, y se hace foco en *Scrum* y *XP* para los cuales se describen sus características y las adaptaciones en la industria. En la sección 3 se expone el relevamiento de la industria de videojuegos en Uruguay con una breve descripción de ella y el resumen de las metodologías seguidas. En la sección 4 se resumen los principales aspectos de la metodología, sus roles y ciclo de vida. El detalle completo de SUM se encuentra publicado en [SUM09]. En la sección 5 se presentan las conclusiones y el trabajo que se lleva a cabo actualmente para evaluar la metodología.

2 Metodologías Ágiles para Videojuegos

La tendencia a utilizar metodologías ágiles para videojuegos tomó fuerza en los últimos años por existir varios casos de empresas en la industria que logran adaptar estas metodologías y además ser un tema actual en uno de los eventos principales como es la *Game Developer Conference* (GDC) [Kei09]. A pesar de esto, ninguna de estas adaptaciones está especificada formal y públicamente.

Estos son los principales beneficios que reportan los casos de éxito al utilizar metodologías ágiles:

1. Al ser metodologías iterativas e incrementales se obtienen versiones jugables del producto en intervalos regulares de tiempo. Esto facilita una visión temprana del resultado final del juego, lo cual reduce la probabilidad de cambios de requerimientos en forma tardía y brinda una mayor retroalimentación del cliente.
2. Permiten tener una mayor visión y control del avance del proyecto, tanto al cliente como a los desarrolladores. Esto se debe a que se pueden determinar nuevas estrategias, iteración por iteración, para lograr llegar en tiempo y forma a los plazos requeridos.
3. Involucran a todo el equipo en las decisiones, lo que logra compromiso y motivación.

2.1 Scrum

Scrum es una metodología ágil para gerenciar y controlar el desarrollo de software de un producto en forma iterativa e incremental. Una de sus características es que no indica prácticas específicas a seguir durante el desarrollo [ASR02], lo que brinda flexibilidad y permite ajustar el proceso a la realidad y forma de trabajo de cada proyecto, así como a los diferentes requerimientos de los clientes. Existen casos de empresas en la industria que logran adaptar esta metodología para videojuegos y les reporta beneficios, por ejemplo *High Moon Studios* [Kei07], *Large Animal Games* [Tob08], *Crytek* [Cry08], *Relic* [Rel08], *DICE* [Nut08] y *Nokia* [Gam08].

Según la descripción que realiza Ken Schwaber[SB01], *Scrum* se estructura en tres fases denominadas *pre-game*, *game* y *post-game*. Durante el *pre-game* se

define el producto basado en las características conocidas, estimando su tiempo y costo. También se analiza el sistema a construir, se define la arquitectura y se realiza un diseño de alto nivel de la solución. La fase de *game* consta de iteraciones, que duran de dos a seis semanas, donde se desarrollan las características del producto. Al comienzo de cada una se realiza su planificación, donde se describen, priorizan y estiman las características que se van a desarrollar y al concluir se evalúa su resultado. El *post-game* es el cierre del proyecto, donde se prepara la liberación del producto, se verifican las versiones a entregar y se realiza la documentación final.

La metodología define tres roles entre los cuales se dividen todas las responsabilidades de un proyecto: *Product owner*, *Scrum master* y *Scrum team*. El *Product owner* está a cargo del proyecto y es quien maneja y prioriza las características a desarrollar. El *Scrum master* es el responsable de que los miembros del equipo sigan el proceso como es debido y de remover los impedimentos que surjan en el transcurso de este. El *Scrum team* es un equipo multidisciplinario y auto organizado, y su cometido principal es construir el producto que el *Product owner* especifica.

2.2 Extreme Programming

Extreme programming(*XP*) es un proceso de desarrollo ágil que puede ser usado por equipos de tamaño pequeño a mediano para desarrollar software de alta calidad en un tiempo previsible y con una sobrecarga de trabajo mínima [BA04].

En resumen, *XP* es una colección de valores, derechos y buenas prácticas, las cuales han sido utilizadas durante años en la industria de desarrollo de software. *XP* las identifica y las agrupa, ya que, usándolas en conjunto, es cuando realmente se obtiene el mayor beneficio.

En la industria de videojuegos, la empresa *High Moon Studios* [Kei08] reporta la utilización exitosa de algunas de las prácticas de *XP*. A su vez *Titus Interactive Studios* plantea una propuesta de adaptación de *XP* para el desarrollo de videojuegos llamada *Extreme Game Development*(*XGD*) [Dem08] en donde incorporan las prácticas de *XP* a las diferentes disciplinas del desarrollo de videojuegos. No hay resultados publicados acerca de esta propuesta, ya que, la empresa cerró antes de finalizar los proyectos que la seguían.

3 Relevamiento de la Industria Uruguaya

Con la motivación de conocer la industria uruguaya de videojuegos se realizan entrevistas entre marzo y abril de 2008 a cuatro empresas referentes en este rubro. El relevamiento hace foco en las metodologías de desarrollo que utilizan e incluye otros aspectos de las empresas como infraestructura, clientes, tipos de proyectos y estrategias de negocio. Las empresas relevadas fueron *Batovi Game Studio* [Bat08], *Mystery Studios* [Mys08], *Powerful Robot Games* [Pow08] y *Kef Sensei* [Kef08].

En resumen, la industria se caracteriza por ser joven (han transcurrido siete años desde la fundación de la primer empresa), y por estar formadas por empresas pequeñas en infraestructura y en cantidad de personal (entre tres y quince personas por empresa). La mayoría de los proyectos que se realizan se acotan a videojuegos de tipo casual o *advergaming* (videojuegos que publicitan una marca o producto) para las plataformas PC y web, cuyo desarrollo demanda entre dos y doce meses. Esto se debe a los recursos disponibles, tanto económicos como de personal con la capacitación y experiencia necesaria. Sin embargo, la industria busca crecer económicamente, para ello la estrategia que plantean las empresas es desarrollar videojuegos por su propia cuenta o con financiamiento en etapas avanzadas del desarrollo como forma de mejorar los ingresos.

Los equipos de desarrollo se conforman de tres a cuatro integrantes promedio, cubriendo los roles de productor, programador, artista gráfico y diseñador de juego. Los contenidos de audio son realizados por empresas externas especializadas, ya que no cuentan con personas capacitadas o económicamente no lo consideran redituable. El productor se responsabiliza del seguimiento del proyecto y la comunicación con el cliente, generalmente es una única persona y participa en varios proyectos a la vez. El diseño del juego es llevado a cabo en algunos casos por el integrante de mayor experiencia y en otros por todo el equipo.

El proceso general de desarrollo comienza por definir y acordar la idea del videojuego a realizar. Luego, se especifican sus características y se planifican los plazos de entrega. Para la elaboración del videojuego se relevan dos formas de trabajo, de las cuales la primera es la que se utiliza en la mayoría de las empresas y la segunda solo en una. Esta es iterativa e incremental con ciclos de corta duración, donde en cada ciclo se diseña, implementa y verifican un subconjunto de las características del videojuego. Al final del ciclo se muestra el progreso logrado para evaluar el videojuego y realizar cambios sobre su especificación. La segunda es secuencial, donde primero se realiza el diseño completo de la solución para luego implementar y posteriormente verificar. Una vez terminada la elaboración se realiza una verificación funcional externa al equipo de desarrollo para detectar errores y evaluar el videojuego. A partir de los errores y evaluaciones que se reportan, se corrige el videojuego hasta alcanzar la versión final, la cual se distribuye de acuerdo al modelo de negocio determinado.

La metodología utilizada se ve influenciada por la forma de financiar el proyecto, se basan en su experiencia y no están formalmente definidas. Cuando la financiación es externa, quien financia, impone plazos, prácticas y entregables a generar durante el desarrollo. Esto hace que el proceso sea más ordenado y apunte a cumplir en tiempo y forma con los plazos impuestos. Quien financia se encarga además de la verificación funcional externa, así como del marketing y la distribución del videojuego. Esta modalidad de trabajo tiene como desventajas la pérdida de autonomía en cuanto a decisiones sobre aspectos del videojuego y la disminución de las ganancias al obtener un menor porcentaje sobre las ventas. Como ventajas, permite generar experiencia, hacer conocida la empresa en el mercado y reducir riesgos económicos. Todas las empresas adoptan esta modal-

idad, ya que, les permite financiar sus propios proyectos de forma paralela o a futuro.

Cuando la propia empresa financia el proyecto, se cuenta con mayor flexibilidad a la hora de decidir las características y los plazos. Esta flexibilidad tiene como ventaja un mayor tiempo para crear elementos divertidos que hagan atractivo al juego, pero en contrapartida suponen el riesgo de invertir demasiado tiempo en busca de la perfección. La verificación funcional externa es menos formal ya que solamente se distribuye el videojuego entre conocidos, además existe la posibilidad de negociar con más de un distribuidor. Esta modalidad permite a la empresa obtener mayores ingresos pero supone cargar con los riesgos de la inversión.

Las metodologías utilizadas para el desarrollo de videojuegos siguen principios ágiles por ser iterativas e incrementales, tener interacción frecuente con el cliente y ser flexibles ante los requerimientos cambiantes. Otra característica es que las decisiones se toman en base a la experiencia, sin existir un proceso definido ni técnicas específicas a seguir. Para ello algunas empresas utilizan varias de las prácticas de metodologías ágiles conocidas como *Scrum* y *XP*.

4 Metodología SUM para Videojuegos

Dado que no existe una metodología ágil formalmente especificada para el desarrollo de videojuegos se realiza una propuesta como aporte a la industria. La misma sigue los principios de las metodologías ágiles y adapta la estructura y roles de *Scrum*. Esta adaptación busca contemplar a la realidad relevada en la industria en Uruguay y resumir la experiencia de los casos que adaptan con éxito estas metodologías para obtener sus beneficios.

La metodología SUM para videojuegos tiene como objetivo desarrollar videojuegos de calidad en tiempo y costo, así como la mejora continua del proceso para incrementar su eficacia y eficiencia. Pretende obtener resultados predecibles, administrar eficientemente los recursos y riesgos del proyecto, y lograr una alta productividad del equipo de desarrollo. SUM fue concebida para que se adapte a equipos multidisciplinarios pequeños (de tres a siete integrantes que trabajan en un mismo lugar físico o estén distribuidos), y para proyectos cortos (menores a un año de duración) con alto grado de participación del cliente.

La definición de la metodología se basa en el *Software and Systems Process Engineering Metamodel Specification*(SPEM) 2.0 [Gro07], un meta-modelo para describir procesos y metodología desarrollado por el *Object Management Group* (OMG). Una ventaja de utilizar SPEM es que su estructura permite especificar el proceso de desarrollo de videojuegos sin mencionar prácticas específicas, lo que lo hace flexible y adaptable a cada realidad. Para especificar la metodología se utiliza *Eclipse Process Framework* (EPF) [Fou08] ya que provee un marco de trabajo extensible basado en los conceptos de SPEM 2.0 para definir y manejar procesos de desarrollo de software.

SUM adapta para videojuegos la estructura y roles de *Scrum* descritas por Ken Schwaber [SB01]. Se utiliza esta metodología ya que brinda flexibilidad para

definir el ciclo de vida y puede ser combinado fácilmente con otras metodologías para adaptarse a distintas realidades.

4.1 Roles

La metodología define cuatro roles: equipo de desarrollo, productor interno, cliente y verificador beta. El productor interno y el cliente se corresponden en forma directa con los roles de *Scrum Master* y *Product Owner* de *Scrum* respectivamente.

El equipo de desarrollo tiene las características del *Scrum team*, pero a diferencia de *Scrum* se definen subroles dentro del equipo. Es necesario esta definición ya que se requiere una alta especialización para satisfacer las distintas disciplinas que involucra del desarrollo de videojuegos, aspecto no contemplado en *Scrum*. Estos se corresponden con los que se utilizan habitualmente en la industria local y son los de programador, artista gráfico, artista sonoro y diseñador de juego. El programador define la arquitectura, realiza el diseño, implementación y verificación de los componentes de software e integra el contenido audiovisual del videojuego. Los subroles de artista gráfico y artista sonoro se encargan de la creación del contenido audiovisual del videojuego. El artista gráfico realiza el arte de concepto, el arte 2D, el modelado 3D y la creación de animaciones y texturas. El artista sonoro se encarga de la creación, grabación, mezcla y edición de los efectos de sonido y música del juego. Por último el diseñador de juego es el encargado de diseñar el *gameplay*, la historia, el ambiente, los personajes y todos los elementos que hacen a la experiencia del jugador. Además, diseña los niveles, misiones y los desafíos que enfrenta el jugador.

El rol de verificador beta no está presente en *Scrum* pero sí se detecta su existencia en el relevamiento de la realidad local y en la industria del videojuego en general. Su responsabilidad es la de realizar la verificación funcional del videojuego y comunicar su resultado. Sin embargo puede no poseer experiencia ni ser jugador frecuente y participar igualmente de la verificación, por ejemplo, al formar parte de un *focus group* del videojuego.

4.2 Ciclo de Vida

El ciclo de vida se divide en fases iterativas e incrementales que se ejecutan en forma secuencial con excepción de la fase de gestión de riesgos que se realiza durante todo el proyecto. Las cinco fases secuenciales son: concepto, planificación, elaboración, beta y cierre, como se aprecia en la Fig.1. Las fases de concepto, planificación y cierre se realizan en una única iteración, mientras que elaboración y beta constan de múltiples iteraciones.

Las fases surgen como adaptación al desarrollo de videojuegos de las fases *pre-game*, *game* y *post-game* que presenta *Scrum*, donde las dos primeras coinciden con las fases de planificación y elaboración, mientras que la tercera se corresponde con las fases de beta y cierre. Esta división se realiza ya que la fase beta tiene características especiales en la industria de videojuegos. La fase de concepto no se corresponde con ninguna etapa de *Scrum* y se agrega ya que

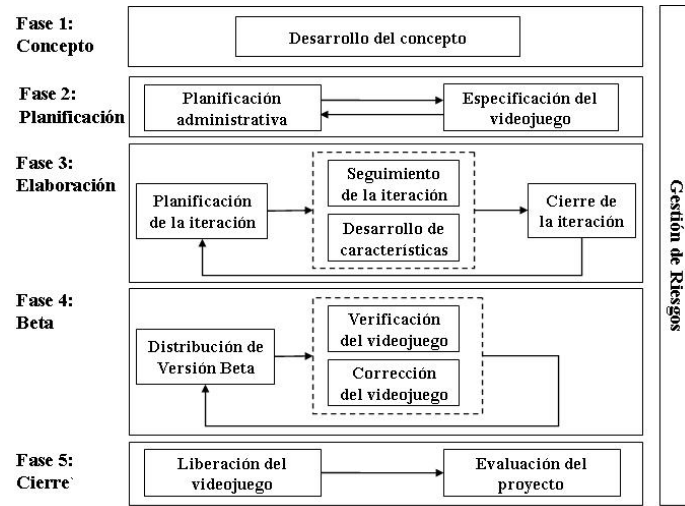


Fig. 1. Fases del proceso

cubre necesidades específicas para el desarrollo de videojuegos y se identifica su uso en la realidad local y en la industria mundial.

Los objetivos principales de cada fase son los siguientes:

- **Concepto:** Tiene como objetivo principal definir el concepto del videojuego lo que implica definir aspectos de negocio (público objetivo, modelo de negocio), de elementos de juego (principales características, *gameplay*, personajes e historia entre otros) y técnicos (lenguajes y herramientas para el desarrollo). El concepto del videojuego se construye a partir de ideas y propuestas de cada rol involucrado sobre los aspectos a definir. Las propuestas se refinan a través de reuniones y se analiza su factibilidad con pruebas de concepto. Esta fase finaliza cuando se tiene el concepto validado entre todas las partes involucradas. No es necesario que el concepto esté definido en forma completa para pasar de fase, ya que hay aspectos que se pueden determinar posteriormente.
- **Planificación:** La fase tiene como objetivo principal planificar las restantes fases del proyecto. Para ello es necesario definir el cronograma del proyecto junto con sus principales hitos, conformar el equipo para la fase de elaboración de acuerdo a las necesidades técnicas del proyecto, determinar y tercerizar las tareas que el equipo no pueda cumplir, definir el presupuesto y especificar el videojuego.

El cronograma del proyecto determina la cantidad de iteraciones y su duración en la fase de elaboración junto con las fechas en las que se planea realizar el pasaje a las etapas beta y cierre. Pueden existir hitos intermedios de avance para cumplir con requerimientos del cliente, algo que es común por causa de los contratos que se realizan en la industria de videojuegos [Bat03].

Se conforma el equipo para el resto de las etapas del proyecto de acuerdo a las necesidades técnicas y artísticas que se identifican. Esta definición puede implicar cambios en el equipo de la fase anterior para cumplir con los requerimientos. En caso de que existan necesidades que las personas que integran el equipo no pueden cubrir, éstas deben ser cubiertas por contratistas externos. La selección y la contratación de estos también es parte de esta tarea.

Definir el presupuesto consiste en determinar cuáles son y cómo obtener los recursos económicos necesarios para realizar el proyecto. Dos de los componentes principales del presupuesto son los salarios del equipo y los costos externos, como por ejemplo el hardware necesario para desarrollar o el pago a contratistas externos.

Estos aspectos componen la planificación administrativa del proyecto, y es el productor interno el responsable de la actividad. Se apoya en el equipo para detectar las necesidades del proyecto y elaborar el cronograma. El cliente también participa, ya que debe dar el aval al cronograma y al presupuesto.

Especificar el videojuego consisten en describir, estimar y priorizar cada una de las características que definen el videojuego. Una característica representa, en forma similar a una *User Story* de *Extreme Programming (XP)* [Bec04], una funcionalidad del videojuego desde el punto de vista del usuario final. La descripción de cada característica es breve pero permite suficiente detalle para poder estimar el tiempo necesario para realizarla. Al ser definidas desde el punto de vista del usuario final, las características son una excelente herramienta que tiene el cliente para comunicar al equipo los requerimientos del videojuego y medir el avance durante todo el proyecto. El proceso para especificar las características consta de tres pasos. En el primero el equipo junto con el cliente determinan y describen, a partir del concepto del juego, cuáles son las características funcionales y no funcionales del videojuego. La descripción incluye los criterios de aceptación que sirven como herramienta para verificar la característica y para eliminar ambigüedades en la definición de la misma. En segunda instancia el cliente, con el apoyo del equipo, prioriza estas características de acuerdo a su importancia, y por último el equipo estima cuanto tiempo requiere realizar cada una. La especificación que se obtiene en esta fase es flexible ya que a lo largo del proyecto se pueden agregar, modificar y eliminar características, mientras que la prioridad y la estimación de cada característica se actualiza en cada iteración de la fase de elaboración.

- **Elaboración:** El objetivo de esta fase es implementar el videojuego. Para ello se trabaja en forma iterativa e incremental para lograr una versión ejecutable del videojuego al finalizar cada iteración. El proceso sigue la secuencia de actividades que se muestra en la Fig.1 y que se detallan a continuación.

1. Planificar iteración:

En esta actividad se planifican los objetivos a cumplir, las métricas a utilizar en el seguimiento, las características a implementar y las tareas necesarias para ello. Los objetivos describen que se pretende lograr al finalizar la iteración y se utilizan para evaluar el éxito de la misma. Sirven también de guía para la toma de decisiones en el transcurso de

la iteración. La selección de las características se realiza en base a su prioridad y a los objetivos de la iteración. La suma de los tiempos estimados de las características seleccionadas no debe superar la duración de la iteración. Existen diversas técnicas para llevar a cabo esta tarea, las cuales se brindan como guías del proceso. Cada característica elegida, se divide en tareas de menor duración lo cual hace más sencillo estimarlas, asignarlas a un miembro del equipo, identificar desviaciones, verificarlas y evaluar su completitud.

El cliente y el equipo son los responsables de definir los objetivos y las características a implementar. El equipo además determina las tareas necesarias para realizar las características.

2. Seguimiento de la iteración: su propósito es mantener la visión y el control de la iteración en base a los objetivos planteados. Para ello es necesario definir métricas, registrar medidas y comunicar sus resultados. En caso que ocurran problemas se deben identificar soluciones posibles de acuerdo a su impacto en los objetivos de la de iteración y del proyecto. Posibles soluciones pueden ser, ingresar nuevas tareas a realizar en la iteración o cambiar el plan de la iteración en caso de desviaciones críticas. El productor interno realiza el seguimiento y mantiene informado al cliente y al equipo del avance. Las soluciones a los problemas son acordadas entre las personas involucradas.
3. Desarrollar características: se desarrollan las características planificadas a través de la ejecución de las tareas que la componen. Una vez que se completan todas las tareas pendientes de una característica, esta se verifica de acuerdo a los criterios de aceptación establecidos. En caso de que no cumpla con alguno de los criterios se debe corregir hasta que lo haga. El proceso para llevar a cabo una tarea se ilustra en la Fig.2. Los miembros del equipo seleccionan las tareas de acuerdo a sus capacidades, y una vez que el equipo aprueba su elección, son responsables por el correcto cumplimiento de estas. Al ejecutar una tarea se pueden identificar nuevas tareas necesarias para completarla, en ese caso se ingresan como nuevas tareas de la iteración. Una vez que se completa la tarea esta es verificada y en caso de encontrar errores se reportan para ser corregidos.
4. Cierre de la iteración: Esta actividad implica la evaluación del estado del videojuego y de lo ocurrido en el transcurso de la iteración para actualizar el plan de proyecto respecto a la situación actual.

A partir de los criterios de aceptación el cliente puede obtener una medida del estado de cada característica planificada para la iteración. El equipo y el productor interno son los encargados de presentarle la versión actual del videojuego con las características construidas. Con esta evaluación se actualiza el plan de proyecto de acuerdo a la situación actual y se pueden agregar, cambiar o eliminar características del videojuego, así como modificar la prioridad y tiempo estimado de cada una de ellas. Estos cambios los realizan el cliente y el equipo, mientras que el productor interno es responsable de actualizar el plan de proyecto.

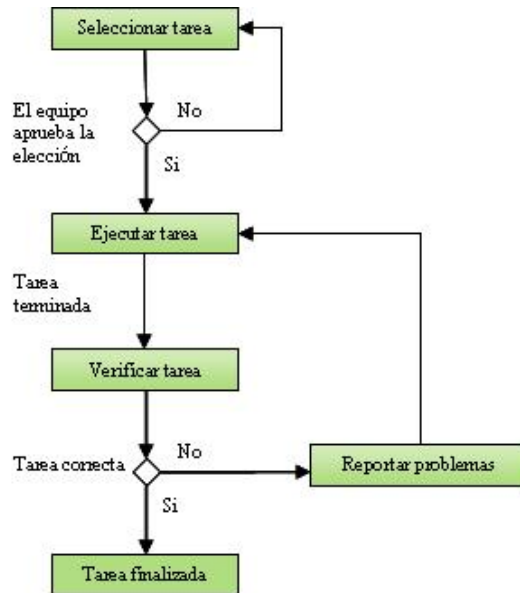


Fig. 2. Proceso para desarrollo de tareas

La evaluación de la iteración consiste en identificar problemas y dificultades que ocurrieron durante la iteración y determinar soluciones para estos. Los responsables de esta actividad son el equipo y el productor interno, en forma opcional puede participar el cliente.

– **Beta:**

La fase tiene como objetivos evaluar y ajustar distintos aspectos del videojuego como por ejemplo *gameplay*, diversión, curva de aprendizaje y curva de dificultad, además de eliminar la mayor cantidad de errores detectados. Se trabaja en forma iterativa liberando distintas versiones del videojuego para verificar. Para ello primero se distribuye la versión beta del videojuego a verificar y se determinan los aspectos a evaluar y la forma de comunicación. Mientras la versión se verifica, se envían reportes con los errores o evaluaciones realizadas. Estos reportes son analizados para ver la necesidad de realizar ajustes al videojuego. Se puede optar por liberar una nueva versión del videojuego para verificar una vez que se realizan los ajustes. El ciclo termina cuando se alcanza el criterio de finalización establecido en el plan del proyecto.

El productor interno y cliente seleccionan a los verificadores beta, proporcionan la versión a probar y establecen los mecanismos de comunicación. Los verificadores beta reportan los errores encontrados y sus reacciones sobre los aspectos mencionados, mientras el equipo de desarrollo es quién corrige el videojuego.

- **Cierre:** Esta fase tiene como objetivos entregar la versión final del videojuego al cliente según las formas establecidas y evaluar el desarrollo del proyecto.
Para la evaluación se estudian los problemas ocurridos, los éxitos conseguidos, las soluciones halladas, el cumplimiento de objetivos y la certeza de las estimaciones. Con las conclusiones extraídas se registran las lecciones aprendidas y se plantean mejoras a la metodología. En la evaluación es recomendable que participen todas las personas que han estado involucradas en el proyecto.
- **Gestión de riesgos:** Esta fase se realiza durante todo el proyecto con el objetivo de minimizar la ocurrencia y el impacto de problemas. Esto se debe a que distintos riesgos pueden ocurrir en cualquiera de las fases, por lo cual siempre debe existir un seguimiento de los mismos. Para cada uno de los riesgos que se identifican se debe establecer la probabilidad y el impacto de ocurrencia, mecanismos de monitoreo, estrategia de mitigación y plan de contingencia.

4.3 Guías

Las guías son sugerencias, pautas y herramientas para llevar a cabo en forma efectiva y eficaz las actividades que componen el proceso. A través de ellas, se incorporan a la metodología prácticas aplicadas con éxito para el desarrollo de videojuegos, además de las lecciones aprendidas con el desarrollo de cada proyecto. Actualmente SUM incluye las prácticas y herramientas de *Scrum* y *XP*, y además, artículos publicados sobre la aplicación de metodologías ágiles en el desarrollo de videojuegos.

5 Conclusiones

Se presenta el uso de metodologías ágiles en la industria de videojuegos a nivel mundial y las características de la industria uruguaya de videojuegos. Se detecta, mediante entrevistas a las empresas de desarrollo de videojuegos más relevantes a nivel nacional, las distintas carencias existentes. Entre ellas se encuentra la falta de formalización de una metodología de desarrollo.

Con el conocimiento que se obtiene se crea una metodología para desarrollo de videojuegos que se adapta a la realidad local. Se basa en los principios de las metodologías ágiles, para obtener los beneficios que estas reportan. Esta metodología se especifica con la herramienta EPF cumpliendo con el estándar SPEM, lo que permite comunicar el proceso en forma efectiva y extenderlo de forma simple.

Actualmente la metodología está siendo evaluada en un caso de estudio que consiste en el desarrollo de un videojuego 3d de acción, multijugador distribuido, utilizando el lenguaje de programación Java. Se cuenta con cuatro integrantes con experiencia en tecnologías de información pero sin experiencia en el desarrollo de videojuegos, artes visuales ni sonidos. El rol de equipo de desarrollo lo

constituyen tres de los integrantes del grupo, mientras que el cuarto interpreta el rol de productor interno. Las decisiones sobre el videojuego son realizadas por los integrantes del grupo, contando con la opinión de potenciales usuarios e interesados en el desarrollo. Este caso de estudio permitirá mejorar y realizar ajustes a la metodología propuesta.

Referencias

- [ASR02] Pekka Abrahamsson, Outi Salo, and Jussi Ronkainen. *Agile Software Development Methods*. VTT Publications, 2002.
- [BA04] Kent Beck and Cynthia Andres. *Extreme Programming Explained: Embrace Change (2nd Edition)*. Addison-Wesley Professional, 2004.
- [Bat03] Bob Bates. *Game Developer's Market Guide*, chapter 1. Premier Press, 2003.
- [Bat08] Batovi Games Studio. Online, Mayo 2008. <http://www.batovi.com>.
- [Bec04] Kent Beck. *User Stories Applied*. Addison-Wesley Professional, 2004.
- [Cry08] Crytek. Transition to scrum midway through a aaa development cycle: Lessons learned. In *Game Developer Conference*, Marzo 2008.
- [Dem08] Thomas Demachy. Extreme game development. Online, Mayo 2008. http://www.gamasutra.com/resource_guide/20030714/demachy_01.shtml.
- [Fou08] Eclipse Foundation. Eclipse process framework project homepage. online, Noviembre 2008. www.eclipse.org/epf/.
- [Gam08] Gamasutra. Interview: Nokia's scott foe - a member of the reset generation. Online, Mayo 2008. http://www.gamasutra.com/php-bin/news_index.php?story=19210.
- [Gro07] Object Management Group. Software and systems process engineering meta-model specification, version 2.0, 2007.
- [Kef08] Kef Sensei. Online, Mayo 2008. <http://www.kefsensei.com/>.
- [Kei07] Clinton Keith. Scrum rising. *Game Developer Magazine*, pages 22–26, Febrero 2007.
- [Kei08] Clinton Keith. An agile restrospective. In *Game Developer Conference*, Febrero 2008.
- [Kei09] Clinton Keith. Advanced scrum and agile development. In *Game Developer Conference*, Marzo 2009.
- [Mys08] Mystery Studio. Computer Games and Games Download. Online, Mayo 2008. <http://www.mysterystudio.com/index.php>.
- [Nut08] Christian Nutt. Living on the edge: Dice's owen o'brien speaks. Online, Mayo 2008. <http://www.gamasutra.com/view/feature/3684/living>.
- [Pow08] Powerful Robot Games. Online, Mayo 2008. <http://www.powerfulrobot.com>.
- [Rel08] Relic. About relic. Online, Mayo 2008. <http://www.relic.com/about/>.
- [SB01] Ken Schwaber and Mike Beedle. *Agile Software Development with Scrum*. Prentice Hall PTR, 2001.
- [SUM09] SUM. Online, 2009. <http://www.gemserk.com/sum>.
- [Tob08] Bliksem Tobey. Introducing scrum at large animal games: a look back at the first year of agile development. Online, Mayo 2008. <http://www.gamasutra.com/view/feature/3677/introducing>.