

# Una Metodología para Desarrollo de Videojuegos

Nicolás Acerenza, Ariel Coppes, Gustavo Mesa, Alejandro Viera  
Eduardo Fernández, Tomás Laurenzo, and 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

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. 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. En promedio, cada proyecto lo realizan de tres a cuatro personas que cubren los roles de productor, programador, artista gráfico, diseñador de juego y artista sonoro. Las tareas de este último habitualmente son realizadas por empresas externas especializadas.

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]. Entre las empresas con casos de éxito documentados se encuentran *Large Animal Games* [Tob08], *Crytek* [Cry08], *DICE* [Nut08] y *Nokia* [Gam08] que utilizan *Scrum* [SB01], *Titus Interactive Studios* [Dem08] que utiliza *XP* [BA04] y *High Moon Studios* [Kei08] que utiliza ambas. A pesar de los beneficios que reportan, ninguna de estas adaptaciones está especificada formal y públicamente.

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.

En la sección 2 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 3 se presentan las conclusiones y el trabajo que se lleva a cabo actualmente para evaluar la metodología.

## 2 Metodología SUM para Videojuegos

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.

## 2.1 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. Éstos 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. 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*.

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.

## 2.2 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 la 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 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

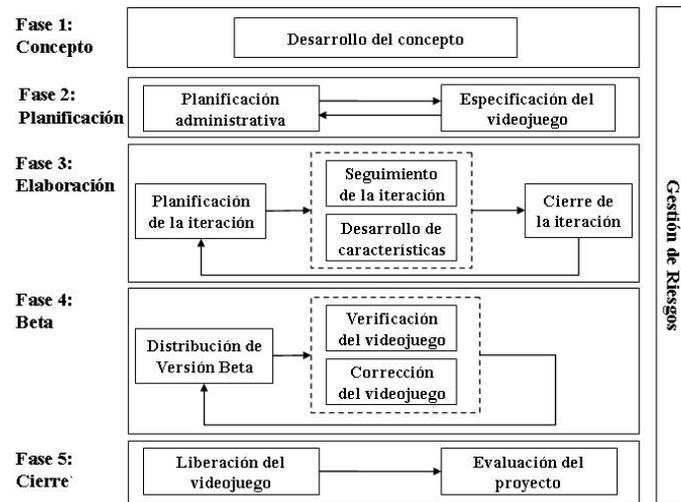


Fig. 1. Fases del proceso

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.

- **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 las características. Esto último consiste en describir, estimar y priorizar cada una de las características funcionales y no funcionales que definen el videojuego. Una característica funcional representa una funcionalidad del videojuego desde el punto de vista del usuario final, mientras que, una característica no funcional representa una propiedad o cualidad que el videojuego debe presentar. La planificación que se obtiene en esta fase es flexible ya que en cada iteración de la fase de elaboración se puede modificar para adaptarse a los cambios y reflejar la situación actual del proyecto.
- **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. Éstas se dividen en tres etapas, en la primera se planifican los objetivos a cumplir, las métricas a utilizar en el seguimiento, las características a implementar y las tareas necesarias para ello. En la segunda se desarrollan las características planificadas a través de la ejecución de las tareas que la componen. Al mismo tiempo se realiza el seguimiento para mantener la visión y el control de la iteración

en base a los objetivos planteados. La tercera y última 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. Con esta forma de trabajo se puede evaluar el avance del proyecto, lo cual permite realizar cambios a tiempo y tomar decisiones para cumplir con los plazos planificados. Además, la experiencia adquirida permite mejorar la forma de trabajo en cada iteración y aumentar la productividad.

- **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. Mientras esta 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.
- **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.
- **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.

### 2.3 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.

## 3 Conclusiones

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. A partir del conocimiento que se obtiene del uso de metodologías ágiles en la industria de videojuegos a nivel mundial y las características de la industria uruguaya de videojuegos, se propone una metodología para desarrollo de videojuegos basada en los principios de las metodologías ágiles que se adapta a la realidad local. 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. El proyecto lo desarrollan cuatro estudiantes con experiencia en tecnologías de información pero sin experiencia en el desarrollo de videojuegos, artes visuales ni sonidos. Las decisiones sobre el videojuego son tomadas por los integrantes del equipo, 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

- [BA04] Kent Beck and Cynthia Andres. *Extreme Programming Explained: Embrace Change (2nd Edition)*. Addison-Wesley Professional, 2004.
- [Bat08] Batovi Games Studio. Online, Mayo 2008. <http://www.batovi.com>.
- [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](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/](http://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](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/>.
- [Kei08] Clinton Keith. An agile retrospective. 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>.
- [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>.