**Selección de la metodología de desarrollo para el proyecto:**

**Desarrollo de un videojuego haciendo uso del framework “Unity ETH UQ”.**

**Dival Mauricio Hoyos Castro**

**Julián David Serna Echeverri**

2016

Facultad de Ingeniería – Programa de Ingeniería de Sistemas y Computación

Universidad del Quindío

**Contenido**

[**Establecer las guías para el análisis de decisiones** 3](#_Toc460001800)

[**Establecer los criterios de evaluación** 3](#_Toc460001801)

[**Métricas de los criterios** 3](#_Toc460001802)

[**Identificar las soluciones alternativas** 4](#_Toc460001803)

[**Metodología cascada** 4](#_Toc460001804)

[**Metodología SUM** 5](#_Toc460001805)

[**Metodología Huddle** 6](#_Toc460001806)

[**Evaluar las soluciones alternativas** 6](#_Toc460001807)

[**Seleccionar la solución** 7](#_Toc460001808)

[**Referencias Bibliográficas** 8](#_Toc460001809)

# **Establecer las guías para el análisis de decisiones**

Durante los últimos años la industria del software ha sufrido cambios drásticos en la manera en que se llevan a cabo los proyectos. El más relevante de estos cambios es que los procesos de preproducción y producción son cada vez más iterativos. Este proceso cíclico está sometido a mejora constante ya que durante este se conciben nuevas ideas, se documentan, se implementan y se evalúan para su posterior aceptación, refinación o descarte.

Debido a lo anterior, durante los últimos años la mayoría de desarrollos de software y por ende de videojuegos han adoptado metodologías ágiles como lo son Scrum o extreme programming (XP), las cuales son ideales para equipos de desarrollo pequeños e igualmente proyectos de pequeña y mediana magnitud, aunque algunos proyectos aún utilizan metodologías robustas debido a aspectos como su magnitud, las necesidades de la empresa, entre otros aspectos.

# **Establecer los criterios de evaluación**

El presente proyecto cuenta con necesidades muy puntuales, a las cuales se debe adaptar la metodología de desarrollo y por ende estas necesidades se convierten en los principales y más importantes criterios de evaluación a la hora de realizar dicha selección, las cuales son:

* **Tamaño del equipo de trabajo:** Dado que el proyecto es un trabajo de grado, está limitado a un grupo reducido de integrantes, específicamente dos, por lo cual dos de las condiciones ideales para la metodología son que no contenga demasiados roles y que el proceso de documentación no sea muy extenso.
* **Involucrar al cliente en el proceso:** Debido a que un trabajo de grado está bajo constante revisión, donde el director y principal revisor adquiere el rol de cliente, la metodología debe garantizar su constante interacción en el proceso.
* **Magnitud del proyecto:** El proyecto tiene como fin poner a prueba ciertos componentes del framework desarrollado por ethereal game factory en un entorno de desarrollo externo, es por esto que su dimensión es reducida.
* **Entregas:** Las constantes revisiones del proyecto abren la necesidad de mostrar avances sólidos cíclicamente y la mejor manera de hacerlo es mostrarlos sobre el aplicativo funcional, permitiendo de esta manera un crecimiento y evaluación secuencial.

# **Métricas de los criterios**

Los cuatro criterios suman diez puntos, dadas las necesidades anteriormente mencionadas, cada criterio tiene el siguiente peso:

|  |  |
| --- | --- |
| **Criterio** | **Peso** |
| Tamaño del equipo de trabajo | 3 |
| Involucrar al cliente en el proceso | 2 |
| Magnitud del proyecto | 3 |
| Entregas | 2 |

Tabla 1: Peso de los criterios de selección.

El valor de cada criterio para las posibles soluciones se basará en el juicio del equipo de trabajo, teniendo tres posibles valores en cada uno:

|  |  |
| --- | --- |
| **Valor** | **Descripción** |
| Cero | Para indicar que la solución no cumple con las necesidades del equipo en ese criterio. |
| La mitad del peso del criterio | Para indicar que la solución no es óptima para el criterio pero puede adaptarse. |
| La totalidad del peso del criterio | Para indicar que la solución es óptima en dicho criterio. |

Tabla 2: Calificación de los criterios de selección.

# **Identificar las soluciones alternativas**

Hoy en día las metodologías para desarrollo de software son muy diversas y hay una cantidad considerable de estas, pero a la hora de ser implantadas para el desarrollo de videojuegos estas deben ser sometidas a ciertos cambios, es allí donde la diversidad escasea, a continuación se exponen tres de las opciones más aceptadas en la industria y con posibles potenciales para el presente proyecto.

## **Metodología cascada**

Esta metodología se caracteriza por tener unas fases muy claras, las cuales son totalmente secuenciales de tal manera que el inicio de una etapa está ligada a la culminación de la anterior, pero antes de dar por culminada una etapa se debe realizar una evaluación para determinar su aceptación o corrección.

Un aspecto relevante de esta metodología es que se enfoca bastante en la documentación para dejar claro desde un inicio todos los aspectos del proyecto sin dejar opción a cambios durante las demás etapas.

Para el desarrollo de videojuegos esta metodología ha sido levemente modificada ya que a diferencia de los productos software tradicionales los videojuegos incluyen componentes gráficos y sonoros, los cuales son de vital importancia.

Las etapas de esta metodología para el desarrollo de videojuegos según Chaparro García (2015) son:

**Especificación del juego:** Se documenta la especificación del videojuego desde la perspectiva del usuario.

**Biblia del arte:** Los productores y directores artísticos especifican las herramientas, conceptos, historia, guion y diseño.

**Especificación técnica:** Se describen las herramientas para realizar los diagramas, interacciones y código.

**Construcción:** Se lleva a cabo la elaboración del videojuego.

**Aseguramiento de calidad:** Se lleva a cabo la verificación de lo realizado con lo estipulado en los documentos de planeación y diseño.

**Pruebas:** Se realiza el testing de las primeras versiones del videojuego para una posterior corrección de errores. Generalmente esta etapa se subdivide en pruebas internas con el equipo de desarrollo, posteriormente pruebas alfa y beta.

**Liberación:** Esta etapa es cuando se libera la versión estable del videojuego al público.

## **Metodología SUM**

Esta metodología para videojuegos tiene como objetivo primordial el desarrollar videojuegos de calidad en tiempo y costo garantizando la mejora continua para mejorar su eficacia y eficiencia. Además de administrar eficientemente tanto los recursos como los riesgos optimizando así la productividad del equipo de trabajo (Gemserk, 2008).

SUM es una metodología ágil que está basada en SCRUM permitiendo flexibilidad en la elaboración del proyecto, ideal para equipos de desarrollo pequeños conformados de dos a siete integrantes, proyectos cortos de duración inferior a un año, equipos multidisciplinarios y además permite una alta interacción con el cliente (Acerenza et al., 2009).

Las etapas de esta metodología, las cuales fueron plasmadas por Gemserk en un dominio web usando Eclipse EPF son:

**Concepto:** Se definen aspectos de negocio como los objetivos del proyecto y a qué audiencia se apunta, elementos sobre el producto a desarrollar como la historia y los personajes y técnicos como el entorno de desarrollo a utilizar para la elaboración de dicho producto.

**Planificación:** En esta fase se planifican el resto de fases del proyecto y se especifican las características a implementar del videojuego.

**Elaboración:** Se realiza un trabajo de forma iterativa e incremental para presentar versiones funcionales hasta culminar en una versión estable.

**Beta:** Su objetivo es evaluar y ajustar de forma iterativa el videojuego eliminando la mayor cantidad de errores detectados.

**Cierre:** Esta es la etapa final, en la cual se pone el videojuego a disposición del público.

**Gestión de riesgos:** Esta etapa mantiene vigente durante todo el ciclo de vida del proyecto y su función es administrar los riesgos y su solución.

## **Metodología Huddle**

Esta metodología fue propuesta por Gerardo Abraham Morales Urrutia, Claudia Esther Nava López, Luis Felipe Fernández Martínez, y Mirsha Aarón Rey Corral en su publicación Procesos de desarrollo para videojuegos en el 2010.

Se llama Huddle a la reunión que se realiza en el juego antes de cada jugada en el futbol americano; la filosofía es que mediante breves reuniones de planeación a corto plazo, se planee cada “jugada” que se inicie; con el fin de dar un seguimiento más estrecho al avance del proyecto y poder hacer correcciones tempranas.

Es un proceso específico para desarrollo de videojuegos óptimo para equipos multidisciplinarios de 5 a 10 personas, ágil, iterativo, incremental y evolutivo, sin embargo puede usarse en equipos de menos de 5 elementos.

Las etapas de esta metodología según sus creadores son:

**Preproducción:** Tiene como objetivo migrar la idea del diseñador al Feature Log y al Sprint Plan; estos documentos dan la pauta a la planeación y producción del videojuego.

**Producción:** Esta etapa se apoya en las herramientas de Scrum como son los Daily Meetings, los Sprints y Sprint Reviews, y en artefactos como el Sprint Backlog y Burn-down Charts y como su nombre lo indica, en esta etapa se realiza la producción del videojuego.

**Postmortem**: Para realizar esta fase, el equipo debe realizar la última actividad llamada End-game Huddle en la cual se analizarán los aspectos positivos y negativos del proyecto. Del End-game Huddle saldrán sugerencias que deberán ser analizadas y filtradas con la intención de generar un reporte que incluya todas aquellas propuestas de mejora al proceso con el objetivo de que sean incorporadas en el próximo proyecto.

# **Evaluar las soluciones alternativas**

A continuación se realiza una tabla comparativa de las tres posibles soluciones contrastando con los criterios de evaluación con el fin de identificar fortalezas y debilidades para el proyecto.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Tamaño del equipo de trabajo** | **Involucrar al cliente** | **Magnitud del proyecto** | **Entregas** |
| **Cascada** | Es flexible al tamaño del equipo, pero su alto nivel de documentación consume una cantidad de tiempo considerable. | El cliente es involucrado en las etapas iniciales y su interacción se ve reducida hasta que se tiene un producto funcional. | Es flexible al tamaño del proyecto pero debido a su alto nivel de documentación se hace ideal para proyectos de gran escala. | Tiene etapas y entregables estructurados, pero es difícil percibir propuestas de valor hasta una entrega funcional y estable. |
| **SUM** | Ideal para equipo de dos a siete personas. | Es una metodología ágil, por lo cual el cliente es un rol vital para las propuestas de valor del proyecto. | Es una metodología ideal para proyectos cortos, de duración menor a un año, por ende su magnitud no es muy grande. | Al ser una metodología ágil, esta plantea sprints en los cuales se le agrega propuesta de valor a un producto funcional. |
| **Huddle** | Óptimo para equipos de 5 a 10 personas, sin embargo, puede utilizarse en equipos de menos de 5 elementos. | Conserva la interacción con el cliente al igual que en SCRUM ya que está basado en este. | Es flexible al tamaño del proyecto, pero se adapta mejor a proyectos no muy largos. | Sus bases agiles hacen que sea una metodología iterativa incremental y en cada huddle se agrega una propuesta de valor. |

Tabla 3: Características de cada solución en cada criterio.

Basados en la tabla 3 se tiene la siguiente evaluación:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Tamaño del equipo de trabajo** | **Involucrar al cliente** | **Magnitud del proyecto** | **Entregas** | **Total** |
| **Cascada** | 1,5 | 0 | 0 | 0 | 1,5 |
| **SUM** | 3 | 2 | 3 | 2 | 10 |
| **Huddle** | 1,5 | 2 | 3 | 2 | 8,5 |

Tabla 4: Evaluación de las soluciones alternativas.

# **Seleccionar la solución**

Teniendo en cuenta la evaluación de las soluciones alternativas (Tabla 4) y de acuerdo a su adaptación a los criterios más relevantes para el proyecto se llega a la conclusión de que SUM y Huddle son las metodologías más convenientes para el proyecto, pero SUM presenta características mejor adaptadas a grupos y proyectos pequeños, por lo tanto es más conveniente.

# **Referencias Bibliográficas**

Acerenza, N., Coppes, A., Mesa, G., Viera, A., Fernández, E., Laurenzo, T., & Vallespir, D. (2009). Una metodología para desarrollo de videojuegos. Retrieved June 1, 2016, from https://www.fing.edu.uy/sites/default/files/biblio/22811/asse\_2009\_16.pdf

Chaparro García, B. (2015). *Ingeniería de software aplicada al desarollo de videojuegos*.

Gemserk. (2008). SUM. Retrieved February 15, 2016, from http://www.gemserk.com/sum/