



# Diagnóstico del proceso ingenieril al iniciar la etapa de implementación en el desarrollo de videojuegos

## Diagnostic of the engineering process at the beginning of the implementation stage in the development of videogames

Andy Hernández Paez

Yaimí Trujillo Casañola

Omar Correa Madrigal

Arturo Orellana García

Universidad de las Ciencias Informáticas. La Habana. Cuba

Universidad de las Ciencias Informáticas. La Habana. Cuba

Universidad de las Ciencias Informáticas. La Habana. Cuba

Universidad de las Ciencias Informáticas. La Habana. Cuba

### Resumen

La ingeniería de software es una disciplina de las ciencias informáticas que aplica actividades sistemáticas, técnicas y procedimientos a lo largo de los procesos de desarrollo y mantenimiento de software. Esta área del conocimiento es un pilar fundamental para lograr el éxito en un proyecto de software. La selección y aplicación de buenas prácticas ingenieriles para la obtención de un producto depende del dominio de aplicación a desarrollar. En la actualidad el desarrollo de software dinámico e interactivo ha tomado protagonismo en las diferentes esferas de la sociedad, tales como: salud, educación, deporte y recreación. Los videojuegos se corresponden a este dominio de aplicación de software dinámico e interactivo. Teniendo en cuenta el alto grado de aceptación de los videojuegos en el mercado, su desarrollo ha requerido que se dedique total atención a las funcionalidades desde su concepción y hasta su aceptación. Para esto es necesario aplicar buenas prácticas ingenieriles que conlleven a un desarrollo de videojuegos persiguiendo un enfoque de calidad. La presente investigación tiene como objetivo diagnosticar el proceso ingenieril al iniciar la etapa de implementación en el desarrollo de videojuegos. Este diagnóstico conduce a identificar los factores críticos de éxito que influyen negativamente en la ingeniería de software y los motivos por los



cuales no se utiliza como referencia en la implementación de videojuegos. El diagnóstico se realiza mediante la aplicación de los métodos científicos: análisis documental, entrevistas y encuestas; utilizándose como instrumento principal el cuestionario mediante preguntas cerradas y abiertas.

**Palabras clave:** diagnóstico, factores críticos de éxito, proceso ingenieril, videojuegos

## Abstract

*Software engineering is a discipline of computer science that applies systematic activities, techniques and procedures throughout the software development and maintenance processes. This area of knowledge is a fundamental pillar to achieve success in a software project. The selection and application of good engineering practices to obtain a product depends on the application domain to be developed. At present, the development of dynamic and interactive software has taken center stage in different spheres of society, such as: health, education, sports and recreation. Videogames correspond to this domain of dynamic and interactive software application. Taking into account the high degree of acceptance of videogames in the market, its development has required that full attention be paid to the functionalities from its conception and until its acceptance. For this it is necessary to apply good engineering practices that lead to a development of videogames pursuing a quality approach. The present investigation aims to diagnose the engineering process at the beginning of the implementation stage in the development of videogames. This diagnostic leads to identify the critical factors of success that negatively influence software engineering and the reasons why it is not used as a reference in the implementation of videogames. The diagnostic is made through the application of scientific methods: documentary analysis, interviews and surveys; using the questionnaire as a main instrument by means of closed and open questions.*

**Keywords:** critical success factors, diagnostic, engineering process, videogame

## Introducción

En la actualidad el desarrollo de sistemas informáticos se ha visto condicionado por las necesidades de mejorar los procesos sustanciales que complementan las diferentes esferas productivas, científicas y tecnológicas de la sociedad. Las tendencias en la implementación de estos sistemas se basan en la aplicación de un proceso de desarrollo organizado y estructurado en busca de la calidad del producto final. Roger Pressman plantea, que es la ingeniería de software la disciplina que guía un proceso de desarrollo y que constituye una tecnología multicapas que integra métodos, procesos y herramientas, con la premisa de garantizar un enfoque de calidad (Pressman, 2010).

La ingeniería de software propone un conjunto de principios para la especificación y representación de productos de software. Estas buenas prácticas son propuestas por metodologías, modelos de procesos, y/o marcos de trabajo, que su utilización en algunos casos puede estar condicionado por el dominio de aplicación a desarrollar. Los videojuegos son productos de software que en su concepción ingenieril resultan atípicos a los sistemas de gestión de información. Las tendencias actuales para encapsular las funcionalidades en productos de este tipo van más allá de concebir requisitos funcionales, de diseño, información y validación, también deben considerarse las estructuras inteligentes que desencadenan acciones a dife-



rentes niveles que conducen a mecanismos (E. Adams, 2009) (J. D. E. Adams, 2012) (Fabricatore, 2013).

En la literatura consultada las buenas prácticas para cumplir con las expectativas finales de un proceso de desarrollo de software son propuestas por métodos de mejora y modelos de calidad, tales como: IDEAL, PCM, CMMI / ISO / Moprosoft (Trujillo, 2014). Estos métodos y modelos indican qué hacer, pero no cómo y cuándo hacer. La calidad del producto debe medirse sistemáticamente, se manifiesta que no debe esperarse al final en una etapa de pruebas, debido a que se pueden incurrir en errores técnicos introducidos por los especialistas que conlleven a que el producto de software no esté en correspondencia con las necesidades del cliente (CMMI, 2010). Los procedimientos realizados para iniciar la etapa de implementación en un proceso de desarrollo son aplicados mediante buenas prácticas de la ingeniería de software, la cual debe garantizar la base de qué debe hacerse para programar las funcionalidades del producto.

Teniendo en cuenta que el proceso ingenieril en el desarrollo de videojuegos posee características atípicas es recomendable utilizar un método de ingeniería de software que propicie buenas prácticas para la especificación y representación de sus operaciones orientadas a mecanismos (Hernández, 2017). La aplicación de estas buenas prácticas debe tomar en consideración algunos factores para garantizar el éxito del proceso ingenieril en el desarrollo de videojuegos. De ahí que, un factor crítico de éxito (FCE) es un punto clave que, cuando está bien ejecutado, define y garantiza el desarrollo de un proyecto de software, logrando sus objetivos (González, 2007). Entre los principales aspectos a considerar para aplicar prácticas de ingeniería de software adecuadamente se encuentran: el/los método(s) ingenieril(es) a utilizar, los recursos humanos y materiales. El objetivo de esta investigación es diagnosticar la ejecución del proceso ingenieril al iniciar la etapa de implementación en el desarrollo de videojuegos, considerando el comportamiento de factores críticos de éxito.

## Materiales y métodos

Como métodos teóricos se utilizan, análisis-histórico lógico e hipotético deductivo. El primero de los mencionados se utiliza para realizar un análisis relacionado con las tendencias actuales del proceso ingenieril en el desarrollo de videojuegos. Permite establecer una caracterización del proceso ingenieril en el ciclo de vida de un videojuego y de los factores críticos que condicionan el éxito de la ingeniería de software en el desarrollo de este tipo de software. Se analizan las características de diseño y estructura de los videojuegos en diferentes contextos. Además, la presente investigación sigue un método hipotético deductivo, a partir de la identificación de buenas prácticas ingenieriles para aplicar en el desarrollo de videojuegos que permitan disminuir la influencia negativa de los factores críticos de éxito.

Por otra parte, se utilizan métodos empíricos, producto a que estos describen y explican las características fenomenológicas a tener en cuenta en la ejecución actual del proceso ingenieril en desarrollo de videojuegos. Estos representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional, específicamente mediante las entrevistas, encuestas y análisis documental. El empleo de estos métodos constituye un medio para el conocimiento cualitativo de los fenómenos reales de la situación existente. Los mismos se emplean principalmente con el objetivo de obtener información sobre las principales buenas prácticas de ingeniería de software para el desarrollo de videojuegos y su comprensión.



## **Metodologías, marcos de trabajo y modelos más utilizados para guiar el proceso ingenieril en el desarrollo de videojuegos**

Una metodología de desarrollo de software en ingeniería de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo de software (CMS, 2008). A continuación, se describen las metodologías y/o modelos más utilizados(as) para el desarrollo de videojuegos (Govardhan, 2010 ) y (Nicolás, 2015):

Waterfall o Desarrollo en Cascada:

- Es considerada una metodología o modelo para el desarrollo de videojuegos.
- Plantea que el diseño tiene que ser perfecto y estar completo antes de comenzar la implementación.
- Es el más utilizado en la industria hoy en día en las empresas grandes.
- Ejecuta de forma lineal durante el proceso de desarrollo de software las siguientes actividades: Requisitos, Diseño, Implementación, Verificación y Mantenimiento.

SCRUM:

- Es una metodología ágil que sirve para administrar y controlar el desarrollo de un producto. Se caracteriza por cinco valores: compromiso, enfoque, sinceridad, respeto y coraje. Estos valores son implementados mediante varias prácticas (sprints, SCRUM master, SCRUM team, backlog de producto).
- Resulta difícil definir la descripción de las funcionalidades del videojuego en una granularidad adecuada, que no sea demasiado específica ni muy vasta.
- Las actividades que complementan el proceso de desarrollo con SCRUM son: Planificación iteración, Ejecución iteración, Inspección y Adaptación.

Programación Extrema (XP, por sus siglas en inglés) (Fernández, 2002):

- La interacción frecuente entre los clientes y los diseñadores permite adaptar el desarrollo del videojuego a las necesidades de ambos y a lograr el resultado esperado.
- Persigue un enfoque iterativo y se centra en productos de trabajo, tales como: Historias de usuario, tarjetas CRC y diseño de casos de prueba de Aceptación.
- Las actividades estándares a ejecutar durante el proceso de desarrollo son: Exploración, Planificación de la entrega, Iteración, Producción, Mantenimiento y Muerte del proyecto.
- Otras actividades a aplicar durante el proceso de desarrollo de forma simplificada con XP son: Planificación, Diseño, Desarrollo y Pruebas.

SUM:

- La metodología SUM para videojuegos tiene como objetivos desarrollar videojuegos de calidad en tiempo y costo, así como la mejora continua del proceso.
- SUM adapta para videojuegos la estructura y roles de SCRUM y XP.



- El proceso de desarrollo se divide en cinco 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.
- Sus productos de trabajo esenciales son: Plan de proyecto, Documento de concepto, Documento de diseño, Especificación de características y Lista de riesgos (Nicolás; Acerenza, 2009).

Huddle (Gerardo, 2010):

- Huddle es un proceso específico para desarrollo de videojuegos con las siguientes características: ágil, óptimo para equipos multidisciplinarios de 5 a 10 personas, iterativo, incremental y evolutivo. Huddle, sin embargo, puede utilizarse en equipos de menos de cinco elementos.
- Divide al proceso de desarrollo de software en 3 actividades fundamentales: Pre-producción (define el juego, la concepción general con sus aspectos más relevantes y los términos en los que se llevará a cabo su materialización), Producción (es una fase dificultosa por la confluencia de actividades diversas. En ella participan multitud de profesionales de ámbitos muy específicos, con clara especialización en función de las sub-fases en las que se ubiquen (diseño del juego, artístico, mecánico y técnico, implementación, pruebas alpha y beta)), Postmostem (se analizan los aspectos positivos y negativos del proyecto de videojuego).

Marco de trabajo para el desarrollo de videojuegos educativos para la asociación civil Edulibre (Cabrera, 2015):

- Basado en las metodologías de desarrollo ágil XP, SCRUM y SUM principalmente donde se toma su estructura, se modifica y adapta a lo que se está buscando.
- Las seis fases secuenciales son: Diseño educativo, Concepto del videojuego, Planificación, Desarrollo, Pruebas beta y Finalización.
- Para equipos multidisciplinarios pequeños de no más de siete integrantes y proyectos con máximo de duración de seis meses.

Marco de trabajo ingenieril para el proceso de desarrollo de videojuegos (Framework engineering) (Hernández, 2017):

- Método de ingeniería de software orientado a encapsular las funcionalidades mediante mecanismos.
- Está compuesto por 20 actividades.
- Comprende 5 etapas que complementan el proceso de desarrollo de videojuegos: Conceptualización, Diseño, Implementación, Prueba y Mantenimiento.
- Incorpora prácticas de calidad de software basadas en procesos de CMMI for Development versión 1.3.
- Los productos de trabajos a generar son: Diseño del videojuego, Especificación de mecanismos, Criterios para validar mecanismos, Modelo de diseño, Arquitectura de software, Modelo de implementación, Guía de jugabilidad, Registro de defectos y Reporte de postmortem.

Teniendo en cuenta los métodos ingenieriles anteriormente caracterizados, se considera que los más apropiados a utilizar para guiar el proceso de desarrollo de un videojuego son: Huddle y el Framework engineering, producto a que estructuran su proceso de forma iterativa y este último se centra en mecanismos de acuerdo a las tendencias actuales.

### **Encapsulamiento de las funcionalidades en videojuegos**

Para el desarrollo de un videojuego es importante tener en cuenta la estructura que este presenta a partir de un esquema básico que se aleja en cierta forma de cualquier otro tipo de software. Dicha estructura suele presentar un bucle principal que se repite hasta que termina el juego. En este bucle se realizan una serie de tareas a nivel de usuario que dan paso a las interacciones con entidades del mundo virtual que modifican sus estados y la inteligencia artificial que involucran estos. A este bucle se le denomina game loop (Muñoz, 2011).

Esta estructura es manejada por motores de juegos que implementan las diferentes mecánicas que componen a un videojuego. Existen múltiples motores de juegos diseñados para trabajar en consolas de videojuegos y sistemas operativos. En general, la funcionalidad típica de estos motores es proveer un motor de renderizado para los gráficos 2D y 3D, motor físico o detector de colisiones, sonidos, animación, inteligencia artificial, redes, streaming, scripting, administración de memoria, escenario gráfico, entre otros elementos (Alegsa, 2010).

Las mecánicas que componen a un videojuego son desarrolladas a partir de las funcionalidades que proveen los motores de juegos. Las mismas se relacionan con los puntos, las medallas, los rankings, los logros, los niveles o la cuenta atrás, pero lo cierto es que estos no son mecánicas de juego, sino componentes o elementos para crear las mismas. Por tanto, una mecánica es una regla de juego con una entrada y una salida que produce cambios en el sistema (J. D. E. Adams, 2012). Existen diversos motores de juegos que permiten la reutilización de mecánicas o mecanismos, algunos de estos motores son: Unreal Development Kit (UDK), Unity 3D, Ardor3D, Axioma Engine, Cry Engine, Fox Engine, Stingray, Frostbite, entre otros (Technologies, 2011).

En los videojuegos la definición de requisitos funcionales no se trata como en el desarrollo de sistemas de gestión de diferentes dominios, sino que estos se especifican en términos de mecanismos. Estos últimos engloban los elementos que conforman las partes de un videojuego como sistema a nivel de escena y por detrás de esta. Los elementos básicos que incluye un mecanismo son: objetos, propiedades, comportamientos y relaciones, donde, los objetos constituyen un bloque básico de construcción que interactúan entre sí. Estos últimos pueden ser físicos y abstractos, ejemplos: avatar, pieza, nivel. Las propiedades representan a los atributos que definen tanto los aspectos físicos como conceptuales de los objetos, ejemplos: color, posición, dinero. Sin embargo, los comportamientos: constituyen potenciales acciones que un objeto puede realizar y las relaciones, la asociación existente entre los objetos (Fulletron, 2008). Se le llama entonces mecanismo, a los “juguetes” creados que relacionan objetos a través de sus propiedades y sus comportamientos con un fin en específico (J. D. E. Adams, 2012). Algunos ejemplos de mecanismos son: mecanismo de control de vidas, mecanismo de locomoción. Además, los mecanismos en un videojuego constituyen reglas basadas en un sistema de simulaciones que facilitan y fomentan en un usuario la necesidad de aprender y explorar propiedades en un ambiente virtual (Cook, 2006).





## Encuesta aplicada a involucrados en el proceso de desarrollo de videojuegos

Para diagnosticar el proceso ingenieril en el desarrollo de videojuegos se aplicó una encuesta que consideró las siguientes interrogantes:

1. ¿El/Los proceso(s) de desarrollo de videojuego(s) en el/los que usted ha participado se ha guiado por algún método de ingeniería de software de tipo: metodología, modelo y/o marco de trabajo?
  - a. En caso de ser positiva la respuesta de la pregunta uno (1) marque con una (X) el/los método(s) de ingeniería de software utilizado(s): SCRUM, XP, RUP, AUP, Huddle, SUM, Otros: ¿Cuál(es)?
2. ¿Considera usted que los artefactos ingenieriles generados según método(s) de apoyo a la ingeniería de software, fueron utilizados como referencias en la etapa de implementación?
  - a. En caso de ser negativa la respuesta de la pregunta dos (2) marque con una (X) los criterios que usted considera que condujeron a la no utilización de artefactos ingenieriles como base en la implementación: No aportaron nada, Fue más fácil programar a partir de la conceptualización, Falta de experiencia del personal, Falta de comunicación, Falta de entendimiento, Poca cultura de trabajo en equipo, Poca motivación del personal, Poca utilización de herramientas de apoyo ingenieriles.
3. ¿Ha desarrollado un videojuego sin concebir artefactos de ingeniería de software?
4. ¿Considera usted que se puede programar un videojuego sin haber aplicado prácticas de ingeniería de software?
5. ¿Considera usted que los métodos de ingeniería de software tradicionales satisfacen el proceso ingenieril para el desarrollo de videojuegos?
6. ¿Ha utilizado usted herramientas de apoyo a la ingeniería de software?
  - a. En caso de ser positiva la respuesta de la pregunta seis (6) marque con una (X) la(s) herramienta(s) de apoyo utilizada(s) en la obtención de artefactos ingenieriles: Visual Paradigm, Open Source Requirements Management Tool, Rational Requisite Pro, Microsoft Visio Profesional, Enterprise Architect, Integral Requisite Analyzer, Otras: ¿Cuál(es)?
7. ¿Considera usted que para la adecuada obtención de artefactos ingenieriles en un proceso de desarrollo de videojuego deben aplicarse actividades de calidad?
8. ¿Ha utilizado prácticas de procesos y productos orientadas a la calidad según modelos de referencias para la ingeniería de software en un proceso de desarrollo de videojuego?
  - a. En caso de ser positiva la respuesta de la pregunta ocho (8) marque con una (X) el/los modelo(s) de referencia por el/los cual(es) se ha guiado para garantizar la calidad en las actividades de ingeniería de software ejecutadas: CMMI For Dev, ISO/ISEC 25010, SPI-CE, MPS.Br, MoProSoft, MCDAI, Otros: ¿Cuál(es)?



9. ¿Considera usted relevante utilizar estrategias para evaluar el contexto ingenieril en el desarrollo de videojuegos y contribuir a la toma de decisiones antes de proceder con la etapa de implementación?
10. ¿Considera usted pertinente utilizar Tecnologías de la información, para evaluar el contexto de la ingeniería de software en un proceso de desarrollo atípico, como resulta un videojuego?

## Resultados y discusión

La encuesta fue realizada por 35 participantes en el marco del Global Game Jam (Prensa Latina, 2018). Este evento tiene como objetivo fomentar la innovación y la experimentación en el campo del software recreativo correspondiente a los videojuegos. Estos participantes poseían entre 3 y 8 años de experiencia en el desarrollo de videojuegos. A continuación, se muestran en la Tabla 1 los roles involucrados en la realización de la encuesta:

Tabla 1. Cantidad de participantes por roles en la encuesta aplicada

| Rol           | Cantidad  | Por ciento  |
|---------------|-----------|-------------|
| Analistas     | 5         | 16%         |
| Diseñadores   | 11        | 31%         |
| Arquitectos   | 3         | 8%          |
| Programadores | 16        | 45%         |
| <b>Total</b>  | <b>35</b> | <b>100%</b> |

Los encuestados pertenecían a varias instituciones de Cuba, dentro de las que se encuentran: UCI (Vertex, FORTES, TLM, CIGED, CESOL, CESIM, DATEC, GEYSED, CEIGE (UCI, 2016), ICAIC, ICRT, ISDI, Universidad de Cienfuegos, Universidad de la Habana, Joven Clubs, Cuentapropistas.

En los resultados obtenidos de la encuesta realizada se identificaron algunas deficiencias relacionadas con el proceso ingenieril en el desarrollo de videojuegos. Dentro de las insuficiencias identificadas se pueden mencionar que:

- El 43% de los encuestados, para el proceso de desarrollo de videojuegos no ha tenido en cuenta la utilización de métodos de ingeniería de software del tipo: metodología, modelo y/o marco de trabajo.
- El 57% que utiliza métodos de ingeniería de software en el proceso de desarrollo de videojuegos, especifican que los usados por ellos son: SCRUM, XP, Huddle y SUM. Sin embargo, estos métodos



no son flexibles a las nuevas tendencias para encapsular las funcionalidades basadas en mecanismos para software de este tipo.

- A pesar que el 57% declara que utilizan métodos de ingeniería de software para guiar su proceso de desarrollo, el 78% no utiliza como referencia los artefactos ingenieriles generados para iniciar la etapa de implementación.
- El 78% que relaciona la no utilización de artefactos ingenieriles como referencias en la etapa de implementación, especifica que esto se debe a que: los productos de trabajo no aportaron nada, fue más fácil programar a partir de la conceptualización, alto grado de abstracción con poca especificación, falta de entendimiento y experiencia, poca cultura de trabajo en equipo y escasa utilización de herramientas de apoyo ingenieriles.
- El 63% manifiesta que ha desarrollado videojuegos sin concebir la utilización de artefactos ingenieriles. Además, el 49% considera que se puede programar productos de software de este tipo sin haber aplicado prácticas de ingeniería de software.
- El 89% considera que los métodos de ingeniería de software tradicionales no satisfacen el proceso ingenieril para el desarrollo de videojuegos.
- El 69% declara que no ha utilizado herramientas de apoyo al proceso ingenieril. Sin embargo, el 31% que manifiesta lo contrario, especifica que las herramientas utilizadas por ellos son: Visual Paradigm, Rational Requisite Pro y Enterprise Architect.
- Teniendo en cuenta que la calidad del producto final viene dada por la sistematicidad de las acciones aplicadas, el 64% considera que para la adecuada obtención de artefactos ingenieriles en un proceso de desarrollo de videojuego deben aplicarse actividades de calidad.
- El 80% especifica que no ha utilizado prácticas de procesos y productos orientadas a la calidad según modelos de referencias para la ingeniería de software en un proceso de desarrollo de videojuego. Sin embargo, el 20% que manifiesta lo contrario, especifica que ha utilizado CMMI For Development versión 1.3.
- El 92% considera factible la definición de buenas prácticas para la toma de decisiones relacionadas con la ingeniería de software en videojuegos antes de proceder con la etapa de implementación. Mientras que el 89% manifiesta pertinente la utilización de Tecnologías de la información como mecanismo de apoyo a la ingeniería de software en el proceso de desarrollo de videojuegos.

A partir del análisis realizado se concluye que las siguientes deficiencias no contribuyen al éxito del proceso ingenieril en el desarrollo de videojuegos, para lo cual es necesario realizar un compendio de los principales factores críticos de éxito:

- El empleo de métodos ingenieriles tradicionales para la representación de las operaciones de un videojuego, provoca que las funcionalidades a encapsular a diferentes niveles no sean comprendidas por el equipo de desarrollo.
- La no utilización de artefactos ingenieriles como referencias en la etapa de implementación, provoca que el videojuego desarrollado no se corresponda de acuerdo a lo especificado y representado como parte de la ingeniería de software.
- Las especificaciones y representaciones abstractas generadas de un videojuego no se corresponden



con las funcionalidades que se implementan, lo cual provoca puntos de vistas diferentes entre la ingeniería y la programación afectándose de esta manera la trazabilidad funcional.

- No se evalúan las buenas prácticas de conceptualización y diseño aplicadas durante el proceso ingenieril con vistas al desarrollo del videojuego, lo cual atenta contra la calidad del producto final.
- La poca utilización de buenas prácticas orientadas a la calidad del proceso y del producto, provoca que el videojuego no sea evaluado objetivamente y por tanto se vea afectado su lanzamiento al mercado.
- El poco empleo de herramientas de apoyo al proceso ingenieril en el desarrollo de videojuegos, provoca un aumento del esfuerzo dedicado a la ingeniería de software sobre este dominio de aplicación.

### Factores críticos que condicionan el éxito del proceso ingenieril

Cuando se tratan los factores críticos de éxito en el proceso ingenieril se toman como referencia los que fundamentan la ejecución de proyectos de ingeniería de software. El Standish Group ha considerado tres causas de fracasos a la hora de analizar los proyectos de ingeniería de software. La primera causa se encuentra relacionada a las demoras en las entregas de los resultados de los proyectos. La segunda causa está vinculada a los problemas relacionados al presupuesto asignado al proyecto. La tercera y última causa proviene de la disconformidad de la organización con los productos resultantes del proyecto (Acosta, 2016).

Los FCE son oportunos directamente a la consecución de la visión, son viables en el periodo establecido por la agenda institucional. Todos los FCE son importantes, pero son especialmente críticos aquellos que cumplen con las dos condiciones anteriores (priorización). Un FCE es un elemento que interrelacionado con otros afines define un área de trabajo de la entidad y debe ser logrado en el período de planificación. Los FCE, al ser agrupados por afinidad, constituyen las dimensiones o áreas de éxito de la institución (Albarrán, 2010).

A continuación, se muestran en la Tabla 2 los principales FCE que condicionan la ejecución del proceso ingenieril y los autores que consideran los mismos:

Tabla 2. Factores críticos de éxito que condicionan el proceso ingenieril

| Factores críticos de éxito | Autor  |
|----------------------------|--|
| Experiencia del personal   | (Guido, 2008), (Hartmann, 2002), (Caralli, 2004), (Barbar M.A, 2008), (Habib, 2009), (Allison, 2010), (Steven, 2017) |
| Compromiso del personal    | (Martina, 2001), (Steven, 2017), (Sulayman, 2012), (Habib, 2009), (Montoni, 2010), (Dyba, 2005), (Mathiassen, 2005)  |



|  |  |
|--|--|
| Motivación del personal  | (Sulayman, 2012), (Steven, 2017), (Allison, 2010), (Montoni, 2010), (Dyba, 2005), (Niazi, 2010), (Mathiasen, 2005) |
| Formación del personal   | (Newton, 1995), (Kassicieh, 1998), (Sohal, 2000), (Antony, 2002), (Babar, 2009), (Habib, 2009), (Allison, 2010)    |
| Comunicación   | (Guido, 2008), (Martina, 2001), (Steven, 2017), (Sulayman, 2012)   |
| Cultura de trabajo en equipo   | (Guido, 2008), (Antony, 2002), (Steven, 2017), (Allison, 2010), (Montoni, 2010)                                    |
| Participación de involucrados relevantes   | (Guido, 2008), (Martina, 2001), (Steven, 2017), (Dyba, 2005)   |
| Toma de decisiones lógicas   | (Bermúdez, 2003), (Bañales, 2007)  |
| Conocimientos técnicos apropiados  | (Banuelas, 2002), (Steven, 2017), (Guido, 2008)  |
| Análisis conceptual y procedural<br>Capacidad creativa<br>Capacidad de análisis<br>Capacidad de abstracción lógica<br>Diseño estandarizado | Otros  |

### Buenas prácticas ingenieriles que contribuyen al éxito en el desarrollo de videojuegos

Las buenas prácticas en el desarrollo de software contribuyen a la realización efectiva de todos sus procesos, tanto los del grupo de gestión de proyectos, como los de ingeniería y calidad de software (CMMI, 2010). Es crucial en todo dominio de aplicación conocer cuáles son los rasgos característicos que permiten llegar al éxito en su contexto, desde la ingeniería de software y hasta su calidad. A continuación, se listan buenas prácticas ejemplos que para el objeto de estudio deben ser consideradas:

- Seleccionar un método de ingeniería de software que encapsule las funcionalidades basado mecanismos.
- Contribuir a un ambiente de trabajo en equipo donde prime la confianza profesional y personal.
- Conceptualizar el videojuego desde etapas tempranas de proceso de desarrollo.
- Involucrar al cliente y usuarios finales en el proceso de desarrollo.
- Especificar las funcionalidades a partir de: objetos, propiedades, comportamientos y relaciones.
- Estructurar las funcionalidades a partir de la arquitectura de software desde la etapa de concep-



tualización.

- Realizar un diseño coherente desde la conceptualización y hasta la aceptación del videojuego.
- Representar los estados por los cuales transitan los objetos que construyen al videojuego.
- Diseñar con un enfoque de componentes las partes del videojuego que encapsulan la implementación.
- Describir con claridad los elementos de conceptualización, diseño e implementación.
- Emplear un nivel de abstracción entendible por todos los involucrados del equipo de desarrollo.
- Dominar el funcionamiento de las herramientas y tecnologías de implementación.
- Utilizar un modelo de proceso iterativo e incremental.
- Aplicar pruebas estáticas y dinámicas en contextos variables con la presencia o no de los usuarios finales.
- No condicionar el tiempo de pruebas para este tipo de producto de software por ningún otro proceso.

## Conclusiones

Con el análisis realizado sobre el estado actual del proceso ingenieril en el desarrollo de videojuegos se identificaron las principales insuficiencias que no contribuyen al éxito de la ingeniería de software sobre este dominio de aplicación. Los métodos científicos: análisis documental, encuesta y entrevistas permitieron diagnosticar el estado actual de la ingeniería de software en el proceso de desarrollo de videojuegos, demostrando que se le brinda poca importancia o no es utilizada como referencia en muchos casos antes de iniciar la etapa de implementación. Además, los criterios evaluados por los participantes en el desarrollo de videojuegos a nivel nacional permitieron contrastar la necesidad de concebir un proceso ingenieril bien organizado y estructurado para la industria de videojuegos en Cuba. Con la aplicación tanto de la gestión de proyectos como del aseguramiento de la calidad, el proceso ingenieril en el desarrollo de videojuegos debe emplearse también según las tendencias actuales y evaluarse sistemáticamente antes de iniciar la etapa de implementación, esto contribuye a disminuir la influencia de factores críticos de éxito. Estos factores críticos de éxito identificados en la literatura permiten conducir positivamente al proceso ingenieril en el desarrollo de videojuegos. Por otro lado, los métodos ingenieriles más utilizados para representar y especificar videojuegos permitieron identificar las buenas prácticas de ingeniería de software aplicables al desarrollo de este tipo de software.

## Referencias

- Acosta, P., Carlos Alberto; Vilallonga, Gabriel; Riesco, Daniel; Dusso, Juan Pablo; Zurita Perea, Franco. (2016). Marco de trabajo para mejorar el aprovechamiento de factores críticos de éxito en proyectos de ingeniería de software. Departamento de Informática de la Facultad de Tecnología y Ciencias Aplicadas de la Universidad Nacional de Catamarca.
- Adams, E. (2009). *Fundamentals of Game Design*. New Riders Publishing, Thousand Oaks, CA, USA, 2nd edición, 2009. ISBN 0321643372, 9780321643377.



- Adams, J. D. E. (2012). Game Mechanics. Advanced Game Design.
- Albarrán, T., Silvia E. (2010). Factores críticos de éxito en los proyectos. Facultad de Ingeniería UAEM.
- Alegsa, L. (2010). Diccionario de Informática y Tecnología. Definición de motor de videojuego.
- Allison, I. (2010). Organizational Factors Shaping Software Process Improvement in Small-Medium Sized Software Teams: A Multi-Case Analysis. Actas del Proceedings of the 2010 Seventh International Conference on the Quality of Information and Communications Technology.
- Antony, J., Leung Kevin, Knowles Graeme. (2002). Critical success factors of TQM implementation in Hong Kong industries. UK. International Journal of Quality & Reliability Management, Vol. 19 No. 5, pp 551-566. .
- Babar M.A, M. N. (2008). Implementing Software Process Improvement Initiatives: An Analysis of Vietnamese Practitioner 's Views. Actas del Proceedings of the 2008 IEEE International Conference on Global Software Engineering. Celebrado en Bangalore, India.
- Babar, M. A. a. H. Z. (2009). Systematic literature reviews in software engineering: Preliminary results from interviews with researchers. . International Symposium on Empirical Software Engineering and Measurement. Celebrado en Lake Buena Vista, Florida, USA (2009), IEEE Computer Society. p. 346-355. ISBN
- Banuelas, C. R. a. A. J. (2002). Critical success factors for the successful implementation of six sigma projects in organisations. UK. The TQM Magazine, Vol. 14 No. 2, pp 92-99.
- Bañales, D. L. G. A., M. R. (2007). Factores críticos de éxito de la industria del software y su relación con la orientación estratégica de negocio: un estudio empírico exploratorio.
- Bermúdez, A. L. (2003). Factores críticos de éxito en la administración de proyectos de tecnologías de la información qu involucran equipos virtuales. Tesis de maestría, Monterrey.
- Cabrera, J. D. P. (2015). Propuesta de un marco de trabajo para el desarrollo de videojuegos educativos para la asociación civil Edulibre. Guatemala.
- Caralli, R. A. (2004). The critical success factor method· Establising a foundarion for enterprise security management.
- CMMI, E. d. P. (2010). CMMI® para Desarrollo, Versión 1.3. Mejora de los procesos para el desarrollo de mejores productos y servicios. CMU/SEI-2010-TR-033 ESC-TR-2010-033.
- CMS, C. f. M. M. S. (2008). Selecting a Development Approach.
- Cook, D. (2006). What are game mechanics? What are game mechanics? . (10 de mayo de 2018). Recuperado de [http://lostgarden.com/2006\\_10\\_01\\_archive.html](http://lostgarden.com/2006_10_01_archive.html)
- Dyba, T. (2005). An Empirical Investigation of the Key Factors for Success in Software Process Improvement. IEEE Trans. Softw. Eng. Vol 31, núm 5, p. 410-424. ISSN 0098-5589.
- Fabricatore, C. y. L., X. (2013). Fostering Creativity Through Educational Video Game Development Projects: A Study of Contextual and Task Characteristics' Creativity Research Journal. 418-425.
- Fernández, G. (2002). Introducción a Extreme Programming. Ingeniería de Software II.
- Fulletron, T. (2008). Game Design WorkShop: A Playcentric Approach to Creating Innovative Games.
- Gerardo, A. M. U. N. L., C. E.; Fernández Martínez, L. F. y Rey Corral, M. A. (2010). Procesos de desarrollo de Videojuegos. . Instituto de Ingeniería y Tecnología. Universidad Autónoma de Ciudad



Juárez.

- González, B. D. L. (2007). Factores críticos de éxito en la industria del software y su relación con la orientación estratégica de negocio: un estudio empírico *Revista de Gestão da Tecnologia e Sistemas de Informação*, 7, 47-70.
- Govardhan, N. M. A. M. y. A. (2010 ). A Comparison Between Five Models Of Software Engineering. Jawahrlal Nehru Technological University.
- Guido, J. C., J. P (2008). Administración exitosa de proyectos. México: Cengage Learning Edirores., 3a. Ed.
- Habib, Z. (2009). The Critical Success Factors in implementation of Software Process Improvement Efforts: CSFs, Motivators & Obstacles. Tesis de maestría, Tutor F. Agahi, desarrollada en Department of Applied Information Technology. Universidad de Goteborg y Universidad Tecnológica de Chalmers: Gotemburgo.
- Hartmann, F. A., R. A. (2002). Project management in the information systems and information technologies industries. *Project Management journal*, 33(3), S-1 S.
- Hernández, P. A. (2017). Marco de trabajo ingenieril para el proceso de desarrollo de videojuegos. *Revista Antioqueña de las Ciencias Computacionales y la Ingeniería de Software*, 7, 13-26.
- Kassieh, S. K., Yourstone Steven A. (1998). Training, performance evaluation, rewards, and TQM implementation success. NM, USA. *Journal of Quality Management*, Vol. 3 No. 1, pp 25-38.
- Martina, U. L. K. (2001). Determinación de los factores críticos de éxito de outsourcing de sistema de información. San Nicolás de los Garza.
- Mathiassen, L., O.K. (2005). Ngwenyama, and I. Aaen, Managing Change in Software Process Improvement. *IEEE Softw.*, Vol 22, núm 6, p. 84-91. ISSN 0740-7459.
- Montoni, M. A. a. A. R. R. (2010). Applying Grounded Theory to Understand Software Process Improvement Implementation. Seventh International Conference on the Quality of Information and Communications Technology. Celebrado en Porto. IEEE Computer Society. p. 25-34. ISBN 9781424485390.
- Muñoz, F. (2011). Ingeniero de Software especializado en el desarrollo de aplicaciones web. (17 de mayo de 2018). Recuperado de <http://fmcdev.es/blog/2011/02/07/estructura-basica-de-un-videojuego-programacion/>
- Newton, R., Wilkinson Michael. (1995). Critical Success in Management Development. *Management Development Review*. Vol. 8 No. 1, pp 16-24.
- Niazi, M., M.A. Babar, and J.M. Verner. (2010). Software Process Improvement barriers: A cross-cultural comparison. *Information and Software Technology*. Vol 52, núm 11, p. 1204-1216. ISSN 0950-5849.
- Nicolás, A., A. C. (2015). Producción y desarrollo de Videojuegos.
- Nicolás; Acerenza, A. C. M., G Viera A Fernández, E Lorenzo, T Vallespir, D. (2009). Una metodología para el desarrollo de videojuegos: versión extendida. Montevideo. Uruguay.
- Prensa Latina. (2018). Cuba Global Game Jam. Prensa Latina. (22 de mayo de 2018). Recuperado de <http://www.cuba.cu/ciencia-y-tecnologia/2018-01-18/cuba-participara-en-el-evento-de-videojuegos-global-game-jam/39734>
- Pressman, R. (2010). *Ingeniería de Software: Un enfoque práctico*.





- Sohal, A. S., Terziovski M. (2000). TQM in Australian manufacturing: factors critical to success. Australia. International Journal of Quality & Reliability Management. Vol. 17 No. 2, pp 158-167.
- Steven, R. M., Ph.D. (2017). Technology Portfolio Management for Project Managers. PMI Tysons Corner Chapter.
- Sulayman, M., et al. (2012). Software process improvement success factors for small and medium Web companies: A qualitative study. Information and Software Technology. Vol 54, núm 5, p. 479-500. ISSN 0950-5849.
- Technologies, U. (2011). Motores Gráficos. (22 de mayo de 2018). Recuperado de [http://www.mat.ub.edu/futurs\\_ub/activitats/Matefest/2011/triptics/motoresgraficos.pdf](http://www.mat.ub.edu/futurs_ub/activitats/Matefest/2011/triptics/motoresgraficos.pdf)
- Trujillo, C. Y. (2014). Modelo Si.MPS.CU para valorar las organizaciones al iniciar la mejora de proceso de software. Revista Cubana de Ciencias Informáticas, 8.
- UCI. (2016). Centros de Desarrollo de Software. (22 de mayo de 2018). Recuperado de <http://www.uci.cu/investigacion-y-desarrollo/centros-de-desarrollo>

