

INGENIERIA DE SOFTWARE EN VIDEOJUEGOS

Ingeniería de Software en Videojuegos

TEMARIO

- 1. Introducción a la Ingeniería de Software**
2. Ingeniería de Software en Videojuegos
3. Consejos para el desarrollo de videojuegos
4. Distribución de Videojuegos 2D

1. Introducción a la Ingeniería de Software

Ingeniería de Software: Es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, la aplicación de la ingeniería al software (IEEE).

1. Introducción a la Ingeniería de Software

Un **modelo de proceso** para desarrollo de software es el conjunto de actividades necesarias para transformar los requerimientos del usuario en un sistema de software.

Cada actividad tiene entradas y salidas que pueden ser documentación, código, binarios, etc.

1. Introducción a la Ingeniería de Software

¿Porqué aplicar Ingeniería de Software?

Para intentar mitigar problemas comunes en el desarrollo de software:

- Cumplir con los plazos.
- Ajustarse al presupuesto.
- Aumentar la calidad del software generado.
- Asegurar que el software cumple con su especificación.
- Asegurar que el software cumple con el propósito para el que fue creado.
- Llegar a un producto fácil de mantener y modificar.

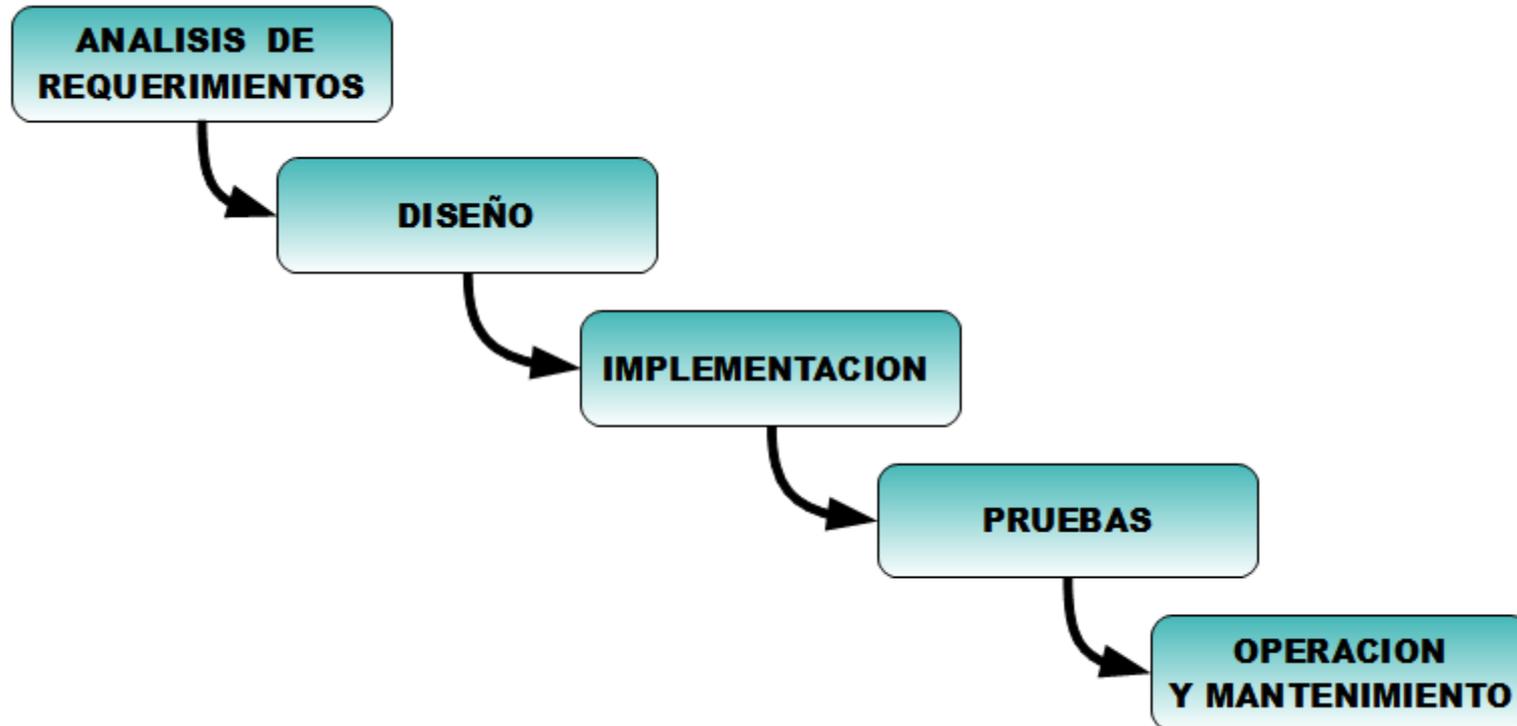
1. Introducción a la Ingeniería de Software

Pasos del desarrollo de software

- Análisis y Definición de Requerimientos.
- Diseño del sistema.
- Diseño del programa.
- Codificación.
- Pruebas/Verificación.
- Liberación Mantenimiento.

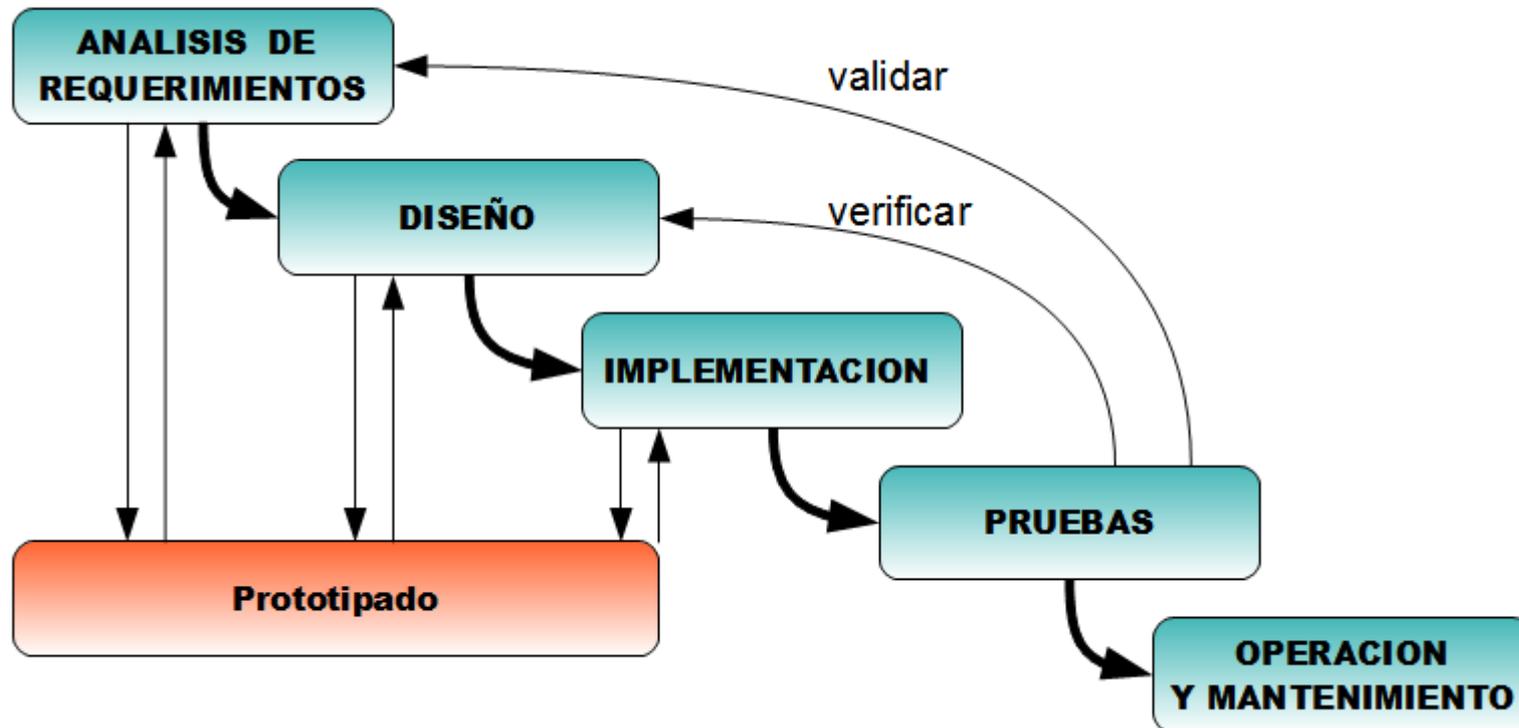
1.1 Modelos de Proceso de Desarrollo

Cascada



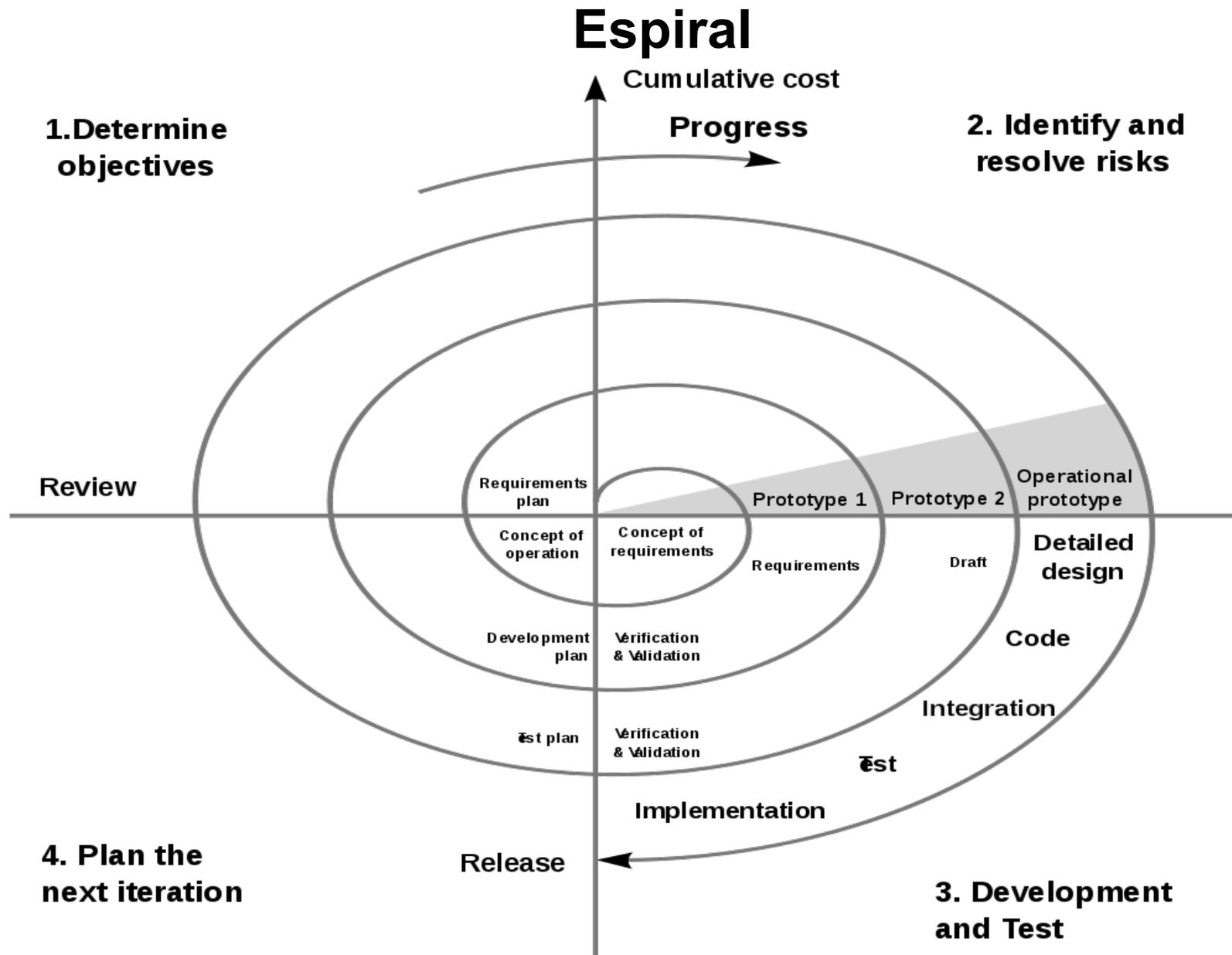
1.1 Modelos de Proceso de Desarrollo

Cascada con Prototipos

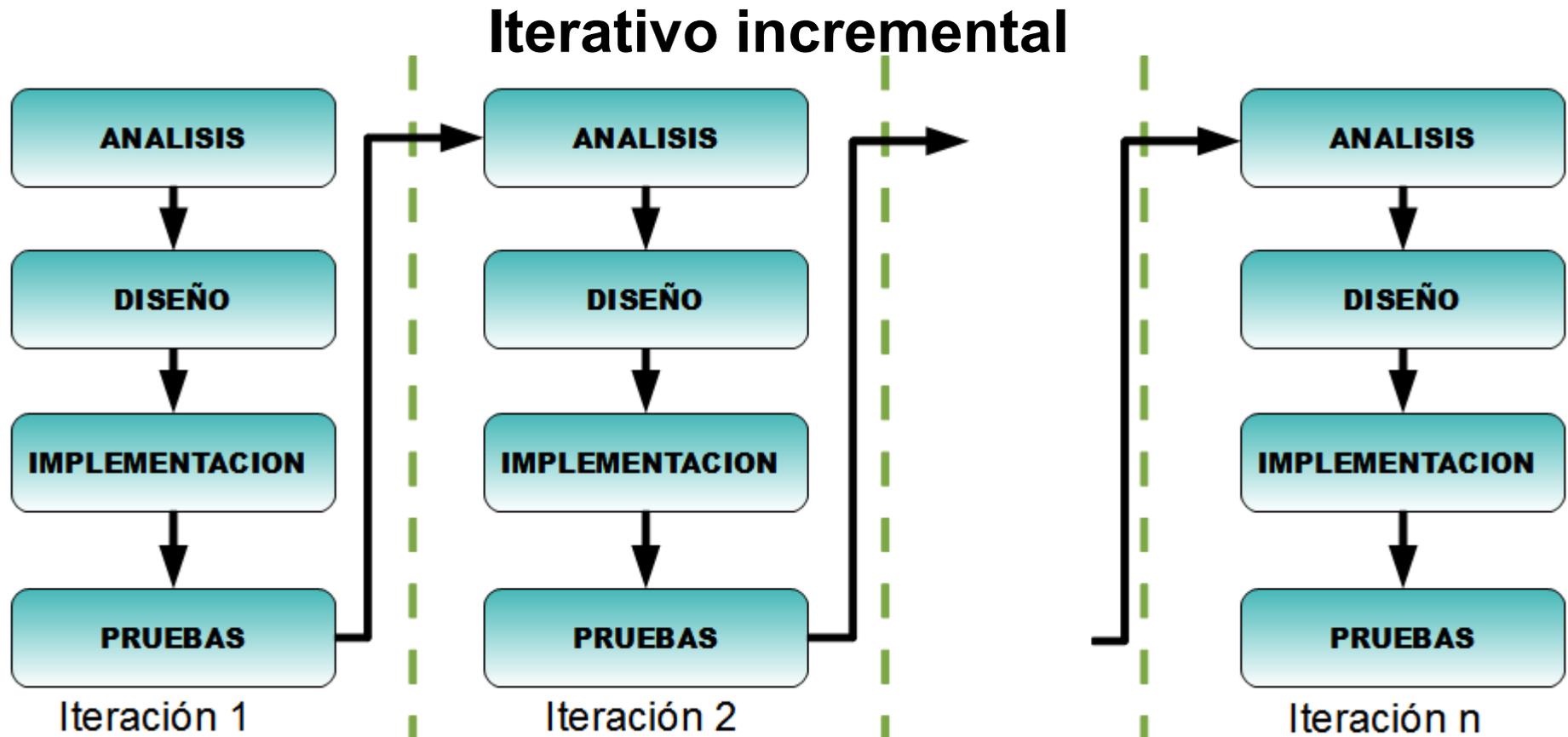


Prototipos: Permiten validar aspectos del producto de forma temprana con usuarios, investigar aspectos tecnológicos o comprobar que es posible realizar el sistema con las restricciones dadas.

1.1 Modelos de Proceso de Desarrollo



1.1 Modelos de Proceso de Desarrollo



En cada iteración se agregan funcionalidades o componentes al producto.

1.1 Modelos de Proceso de Desarrollo

Hay más, pueden adaptarse y combinarse.

El modelo más adecuado al sistema a desarrollar depende del tipo de sistema, sus características y tamaño.

Los procesos de desarrollo de Software son la implementación de esos modelos: Rational Unified Process (RUP), Structured systems analysis and design method (SSADM), XP, Scrum, etc.

1.2 Disciplinas

Requerimientos

- Especificar requerimientos.
- Definir alcance del sistema.
- Especificar funcionalidades.
- Priorizar funcionalidades.
- Validar con el cliente.

1.2 Disciplinas

Diseño

- Describir la arquitectura.
- Diseñar prototipo.
- Diseñar funcionalidades.
- Comunicar el diseño a los implementadores.

1.2 Disciplinas

Implementación

- Implementar prototipos.
- Implementar el producto.
- Integrar componentes del sistema.
- Verificación unitaria de módulos.
- Documentación técnica.

1.2 Disciplinas

Verificación

- Especificar los casos de prueba.
- Ejecutar las pruebas.

1.2 Disciplinas

Implantación

- Planificar la implantación.
- Documentación de usuario.
- Producir la versión del producto a liberar.
- Puesta en producción.
- Pruebas de aceptación.
- Verificar la versión del producto a liberar.

1.2 Disciplinas

Calidad (*Software Quality Assurance*)

- Identificar las propiedades de calidad.
- Revisión técnica formal.
- Revisar las entregas.
- Revisar el ajuste del proceso.
- Evaluar la calidad de los productos.

1.2 Disciplinas

Gestión de configuración y control de cambios (*Software Configuration Management*)

- Planificar la configuración del SCM.
- Definir la línea base del proyecto.
- Seguimiento de la línea base.
- Definir el ambiente controlado.
- Control de cambios.
- Describir la versión.

1.2 Disciplinas

Gestión de proyecto

- Planificar el proyecto.
- Seguimiento del proyecto.
- Estimaciones y mediciones.
- Gestión de riesgos.
- Ajustar y controlar el desarrollo.
- Evaluar y ajustar el plan de proyecto.

1.3 Roles

- Director del proyecto.
- Administrador.
- Arquitecto.
- Coordinador de desarrollo.
- Analistas.
- Diseñadores.
- Especialistas técnicos.
- Implementadores.
- Documentador de usuario.
- Responsable de integración.
- Responsable de SCM.
- Responsable de SQA.
- Responsable de verificación y verificadores.

1.3 Roles

Más Roles:

- Auditor: recaban datos para mejorar el proceso y realizar seguimiento.
- Consultor: brindan apoyo a los distintos roles en las tareas del proyecto.
- Cliente.

1.4 Metodologías Ágiles

Alternativas a metodologías más formales consideradas demasiado pesadas. Se caracterizan por:

- Desarrollo iterativo e incremental.
- Grupos auto organizados y multidisciplinarios.
- Iteraciones cortas donde se agregan pocas funcionalidades. Cada una incluye planificación, análisis, diseño, codificación y verificación.
- Menor carga de documentación y más comunicación directa.

1.4 Metodologías Ágiles

Manifiesto por el Desarrollo Ágil de Software:

Individuos e interacciones sobre procesos y herramientas

Software funcionando sobre documentación extensiva

Colaboración con el cliente sobre negociación contractual

Respuesta ante el cambio sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

<http://agilemanifesto.org/iso/es/>

1.4 Metodologías Ágiles

Ejemplos:

- Agile Unified Process (AUP).
- Open Unified Process (OpenUP).
- eXtreme Programming (XP).
- Scrum.

1.4 Metodologías Ágiles

eXtreme Programming

Tiene por objetivo disminuir la curva exponencial del costo del cambio a lo largo del proyecto.

- Cliente in-situ. Comunicación frecuente el cliente.
- Diseñar la solución más simple.
- Integración continua y pruebas unitarias.
- Refactorización de código (Refactoring).
- Programación en parejas.
- Propiedad colectiva del código.
- Estándares de programación, buenas prácticas.

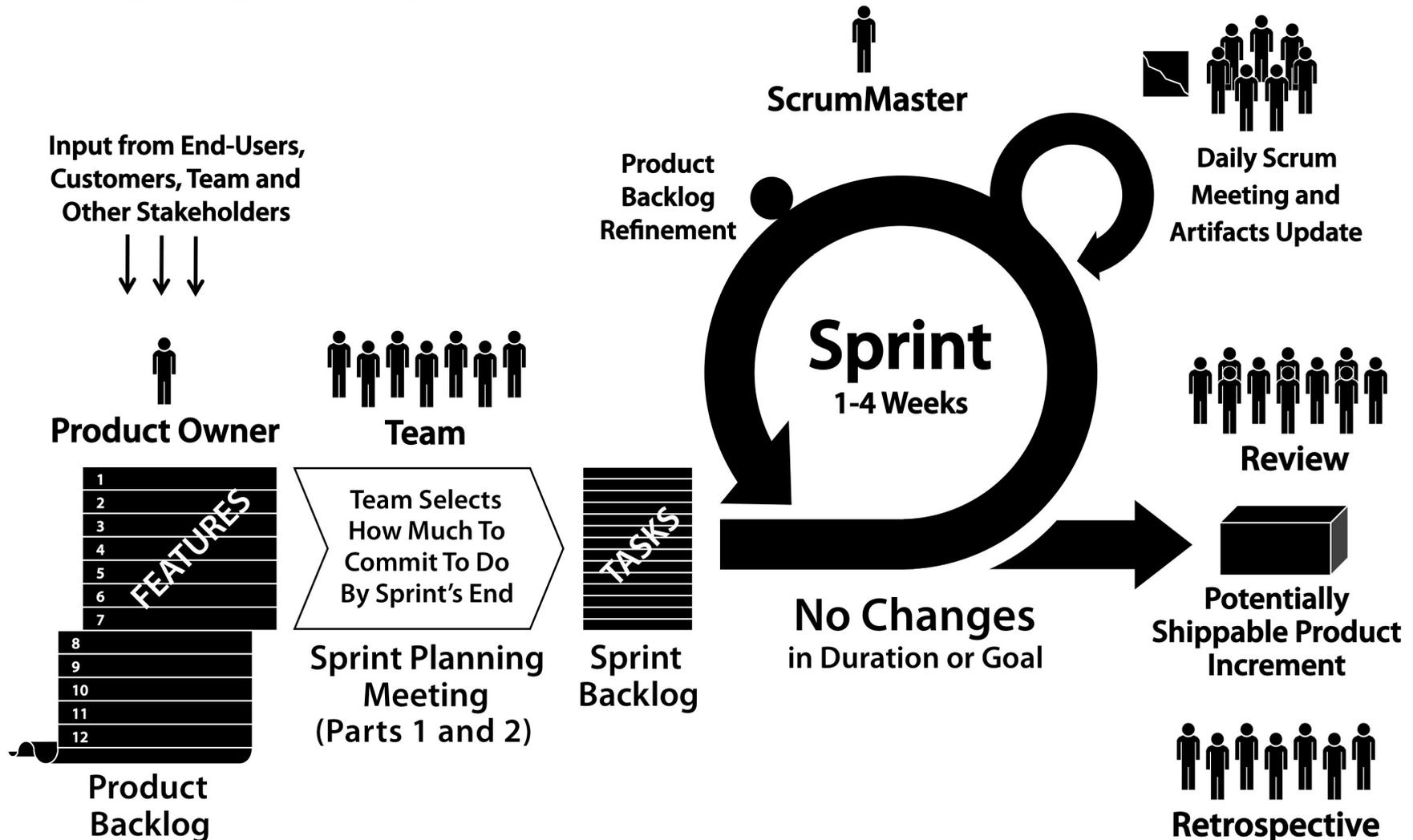
1.4 Metodologías Ágiles

Scrum

- Roles: Product Owner (voz del cliente), ScrumMaster, Team.
- Sprint: tiempo en el que se realiza el trabajo.
- Product Backlog: trabajo pendiente del producto.
- Sprint Backlog: trabajo pendiente del sprint.
- Increment: lo que se añade al producto.
- Reuniones: Daily Scrum, Scrum de Scrums, Sprint Planning Meeting, Sprint Review Meeting, Sprint Retrospective. De duración fija y objetivos claros.

1.4 Metodologías Ágiles

SCRUM



Ingeniería de Software en Videojuegos

TEMARIO

1. Introducción a la Ingeniería de Software
- 2. Ingeniería de Software en Videojuegos**
3. Consejos
4. Distribución de Videojuegos 2D

2. Ingeniería de Software en Videojuegos

A grandes rasgos mismas disciplinas, roles y tareas que los antes descritos.

2.1 Disciplinas y Roles

Dirección

- Director del juego: mantiene la visión y dirige el equipo intentando no intervenir en las creatividades individuales.
- Jefes de departamento: Programación, Diseño, Arte, etc. Dirigen cada departamento, responden al director.

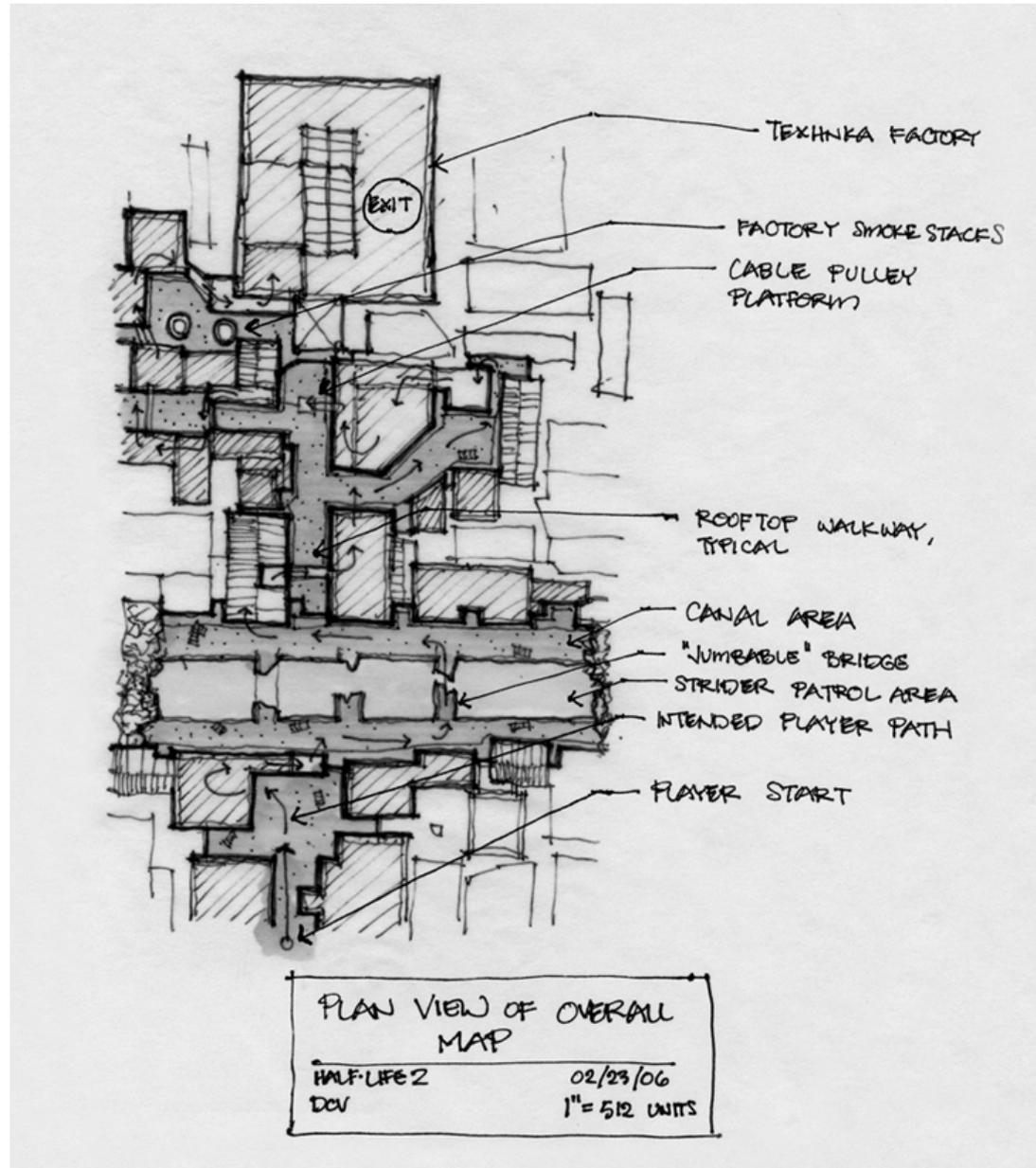
2.1 Disciplinas y Roles

Diseño

- Diseñador del juego: Diseño conceptual (a que se juega). Define reglas y mecánica, diseño para producción.
- Escritor del guión.
- Diseñador de niveles.

2.1 Disciplinas y Roles

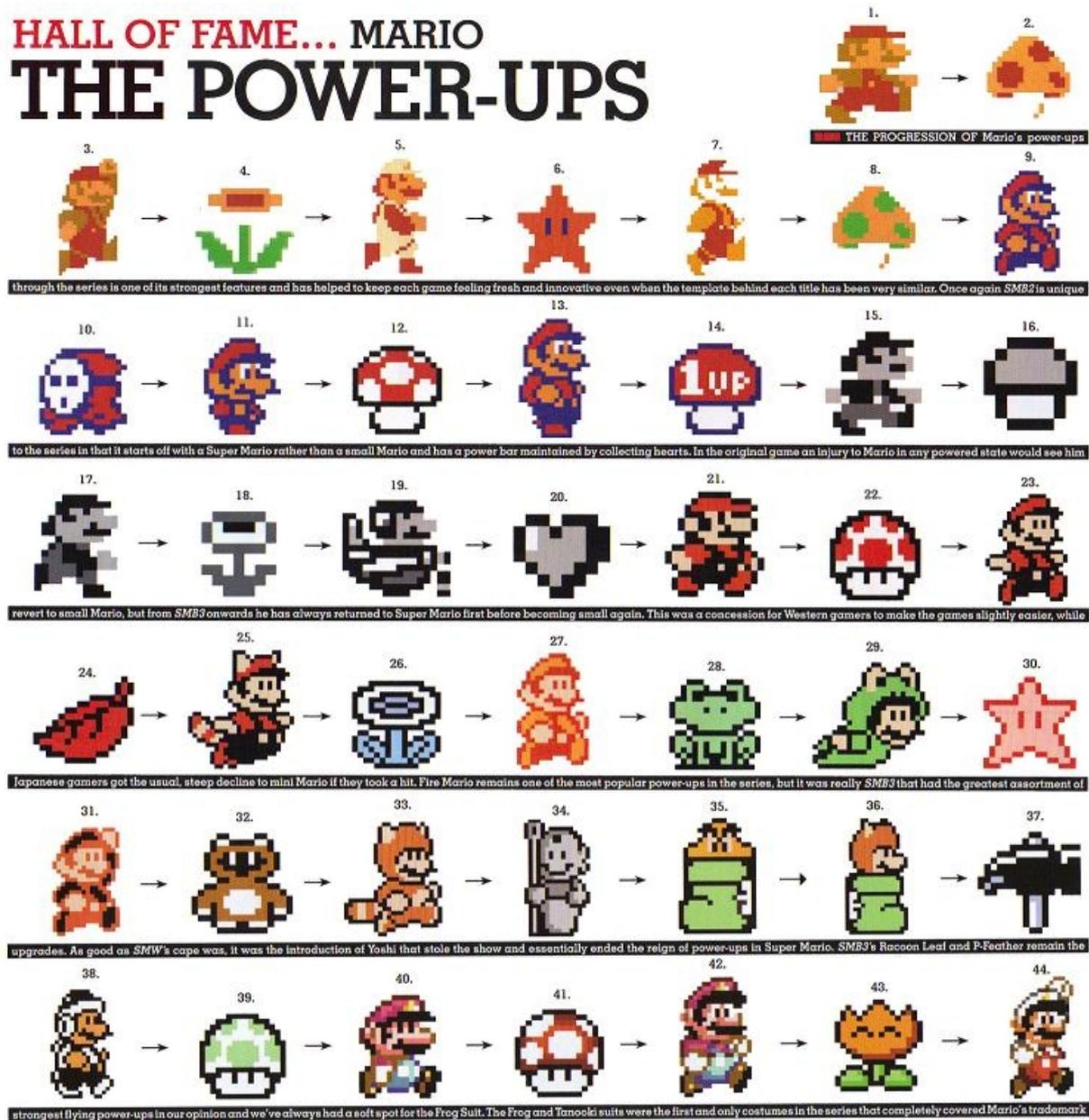
Diseño



2.1 Disciplinas y Roles

Diseño

HALL OF FAME... MARIO THE POWER-UPS



2.1 Disciplinas y Roles

Programación

Programadores:

- De lógica del juego.
- De herramientas/motor (infraestructura).
- Gráfico (efectos).
- Programador/ingeniero de audio.
- De inteligencia artificial.
- De redes.

2.1 Disciplinas y Roles

Artes visuales

Director de arte y animación.

Artistas:

- Constructor de objetos.
- De concepto.
- De texturas.
- Constructor de personajes.
- Animador de personajes.
- Constructor de niveles.

2.1 Disciplinas y Roles

Artes visuales



TDIVJ2D – Tecnólogo Informático

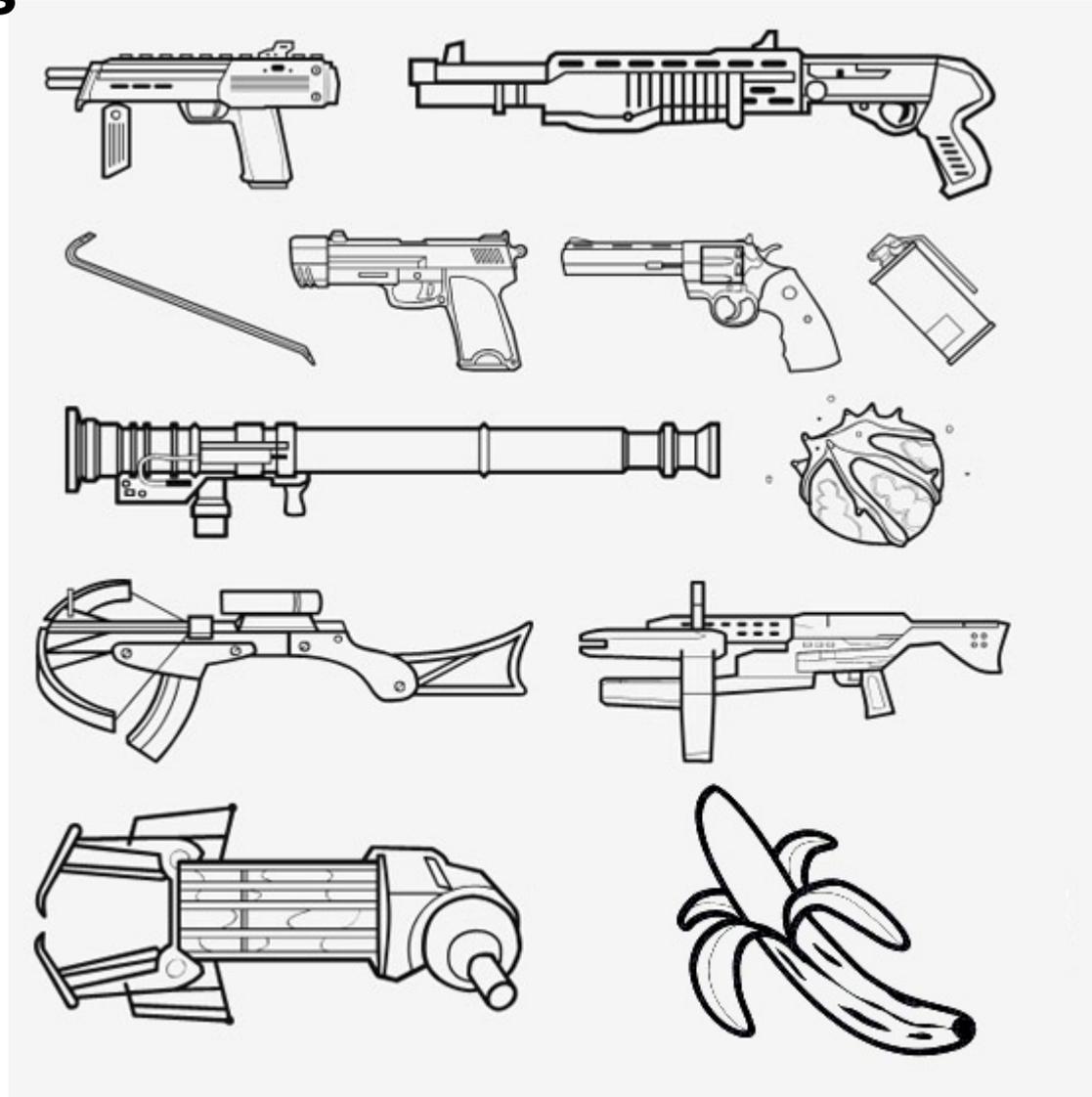
2.1 Disciplinas y Roles

Artes visuales



2.1 Disciplinas y Roles

Artes visuales



2.1 Disciplinas y Roles

Audio

- Ingeniero/diseñador de sonidos: Diseño del estilo y ambiente sonoro. Composición y creación de efectos sonoros. Masterización y edición.
- Compositor: Composición, dirección y creación musical.
- Locuciones de voz: casting y dirección de locutores.

2.1 Disciplinas y Roles

Equipo de Verificación y QA

Verifican que el contenido y funcionalidad del juego son correctos y están completos. Evalúan la jugabilidad.

- SQA.
- Líder de verificación.
- Testers.

2.1 Disciplinas y Roles

Verificación y QA

Funcionalidad: ¿hace lo que debe?

Compatibilidad: ¿en todas las máquinas?

Versiones localizadas: ¿en todos los idiomas?

Jugabilidad: ¿es divertido?

Progresión: ¿la dificultad crece adecuadamente?

Usabilidad: ¿resulta claro lo que hay que hacer?

Equilibrado: ¿muy difícil? ¿muy fácil?

Legal: ¿Tiene aspectos que pueden provocar enfrentamientos o pleitos?

2.1 Disciplinas y Roles

Marketing

Comunican el atractivo del juego al público, al publisher y a los medios.

Aportan al equipo un conocimiento profundo del mercado.
Para tener claro qué quiere el público de cada juego.

2.2 Etapas del desarrollo de Videojuegos

Concepto: la idea fundamental.

Pre-producción: desarrollar la base e implicaciones de esa idea.

Producción: construir la funcionalidad y contenidos.

QA y cierre: asegurar la estabilidad y solidez del juego.

Post-release: parches, actualizaciones.



2.2 Etapas del desarrollo de Videojuegos / Concepto

Concepto

Es cuando se define el concepto del juego. Se crea el arte de concepto para mostrar como se verá el juego y su historia.

Participan Diseñador, Líder técnico (Programador), Artista de concepto, Productor.

Es común realizar una tormenta de ideas.

2.2 Etapas del desarrollo de Videojuegos / Concepto

Artefactos generados en etapa de concepto:

- **Concepto de alto nivel:** descripción en pocas oraciones de lo que es el juego. Incluye las características que lo destacan y lo diferencian de los demás juegos. Muestra la visión del juego.
- **Propuesta del juego:** documento que se utiliza para realizar la presentación del juego en busca de financiamiento. El documento de concepto, explicando de que se trata el juego, por que será exitoso, como será desarrollado y como generará dinero.

2.2 Etapas del desarrollo de Videojuegos / Concepto

Artefactos generados en etapa de concepto:

- **Documento de concepto:** Describe el concepto en forma más detallada. Puede incluir las siguientes secciones:
 - Concepto de alto nivel
 - Género
 - Gameplay
 - Características
 - Ambientación (incluye arte de concepto)
 - Audiencia objetivo
 - Plataformas de hardware
 - Cronograma estimado (*)
 - Presupuesto (*)
 - Equipo de desarrollo
 - Análisis de riesgos (*)

(*) Estos puntos se pueden terminar de completar al comienzo de la etapa de Pre-Producción

2.2 Etapas del desarrollo de Videojuegos / Pre-Producción

Pre-Producción

En esta etapa se demuestra que vale la pena construir el juego y que el equipo está capacitado para hacerlo.

Se desarrollan todas las piezas fundamentales del juego: motor gráfico, editores, exportadores, interfaz del juego, lógica del juego, soporte de red, IA, menús, sonidos, efectos básicos.

El resultado es un prototipo jugable que muestra el estilo y aspecto del juego.

2.2 Etapas del desarrollo de Videojuegos / Pre-Producción

Artefactos de Pre-Producción

- **Plan de proyecto:** Es una guía que indica como va a ser construido el juego. Se divide en varios documentos independientes que son plan de personal, plan de recursos, seguimiento de proyecto, presupuesto, análisis de pérdidas y ganancias, cronograma de desarrollo y definición de hitos del proyecto.
- **Documento de diseño:** Define el juego describiéndolo en forma clara y detallada la mecánica, gameplay, vistas, niveles, personajes, las distintas pantallas, interfaz de usuario, historias, elementos, etc.

2.2 Etapas del desarrollo de Videojuegos / Pre-Producción

Artefactos de Pre-Producción

- **Documento de diseño técnico:** Es un documento de diseño de software. Describe las herramientas que serán utilizadas. Lista hardware y software que debe ser comprado. Establece un cronograma de implementación.
- **Plan de producción de arte:** Indica la estética del juego y el cronograma de creación de arte. Incluye una Biblia de arte que describe los objetos a ser creados
- **Prototipos:** Programas de software que captura la esencia del juego. Serán la mayor influencia para determinar si el proyecto continúa la visión y demuestra que se puede desarrollar el juego en forma efectiva y razonable. Mitiga riesgos tecnológicos.

2.2 Etapas del desarrollo de Videojuegos / Producción

Producción

Se comienza la construcción del juego.

Se escribe el código.

Se crea el arte gráfico.

Se crean los sonidos.

Se crean niveles del juego

Verificadores comienzan a trabajar cuando algo puede ser jugado

2.2 Etapas del desarrollo de Videojuegos / Producción

En Producción hay poca innovación?

Deberían haber pocas innovaciones porque ya se construyeron prototipos, pero la realidad es diferente.

El juego evoluciona o cambia:

- Cambios en requerimientos. Aparecen características nuevas, otras se quitan.
- Surgen imprevistos: Se desechan ideas que podrían ser buenas.
- Se modifican las herramientas y la tecnología a medida que surgen problemas a solucionar.
- Se deben completar y mantener actualizados los documentos ya generados.

2.2 Etapas del desarrollo de Videojuegos / Producción

Hitos

Se deben definir un conjunto de características o componentes a ser implementadas para una fecha determinada.

Los hitos se marcan en el cronograma del proyecto. Deben estar definidos en forma clara, medible y concisa. Mala definición : Tener un nivel jugable.

2.2 Etapas del desarrollo de Videojuegos / Producción

ALFA

Es el punto en el que el juego puede ser jugado de principio a fin.

Quedan detalles por agregar o arreglar, pero el motor, la interfaz de usuario y demás subsistemas están completos.

En este punto el foco cambia de:

- Construir a terminar.
- Crear a arreglar.
- Se comienza la verificación en busca de errores

2.2 Etapas del desarrollo de Videojuegos / Producción

BETA

En este punto se culminó con el desarrollo del juego, todas sus partes están implementadas e integradas.

Se tiene una versión del juego completo, instalable y totalmente jugable en todas sus variantes.

De aquí en más lo único que se hace es arreglar errores.

El objetivo en esta etapa es estabilizar el proyecto y eliminar la mayor cantidad de errores posible antes de liberar el juego.

2.2 Etapas del desarrollo de Videojuegos / QA y Cierre

QA y Cierre

Entre decenas y cientos de personas empiezan a jugar intentando ganar, perder, hacer que falle o se cuelgue, deje de funcionar en condiciones razonables y no razonables.

El objetivo es encontrar errores, inconsistencias y erratas en el juego, textos, interfaz, etc.

En juegos de mediano a gran porte es común encontrar miles de errores.

2.2 Etapas del desarrollo de Videojuegos / QA y Cierre

Congelación de código: Una vez que se solucionaron los errores encontrados en Beta (o por lo menos los más críticos) se congela el código y queda pendiente de aprobación para pasar a ser la versión final.

Liberación: El juego esta completo, verificado y listo para ser enviado a los canales de distribución.

2.2 Etapas del desarrollo de Videojuego / Post-Release

Post-Release

Parches: Comunes en juegos para PC.

Si se encuentra un problema generalizado se deberá crear un Parche.

Actualizaciones: La actualización agrega contenido adicional al Videojuego. La actualización es un proyecto y necesita ser manejado como tal, con verificación, hitos, y el resto de los elementos del proceso de desarrollo.

2.2 Etapas del desarrollo de Videojuegos

Dificultades en el desarrollo de Videojuegos

- La rápida y constante evolución de las tecnologías.
- Los requisitos cambiantes y difíciles de evaluar (diversión, atractivo gráfico, etc.).
- La comunicación se dificulta por tratarse de equipos de desarrollo multidisciplinarios (artistas, diseñadores, sonidistas, etc.).
- Búsqueda de perfección lleva a retrasos en los plazos planificados (mejores armas, mejores escenarios, etc.).

Ingeniería de Software en Videojuegos

TEMARIO

1. Introducción a la Ingeniería de Software
2. Ingeniería de Software en Videojuegos
- 3. Consejos**
4. Distribución de Videojuegos 2D

3. Consejos

Consejos (para cualquier proyecto)

- Definir el proceso, los roles con sus responsabilidades, sus tareas, entradas y salidas, pasos a seguir para las tareas.
- Cumplir con el proceso.
- Estimar avance, esfuerzo, costos. Detectar de forma temprana riesgos y cuellos de botella. Luego replanificar, evaluar, hacer cambios al proceso o tareas si son necesarios.
- Documentar de forma clara el diseño, componentes y decisiones técnicas.

3. Consejos

Consejos (para cualquier proyecto)

- No inventar la rueda, usar bibliotecas, motores, editores y demás herramientas que aceleren el desarrollo.
- Buen manejo de SCM.
- Prototipar y validar.
- Buena comunicación entre roles y disciplinas.
- Verificar, verificar, verificar, verificar.....
- Mantener buen ambiente, compromiso e interés en el trabajo.

3. Consejos

Consejos (para el desarrollo de videojuegos)

- La formación académica no es suficiente, es necesario realizar “autoestudio”, hay que desarrollar videojuegos para generar experiencia.
- Vocación y Creatividad.
- Jugar y saber de juegos.
- Trabajar en grupo.
- Inglés.
- Matemática.
- Sentido común.

Ingeniería de Software en Videojuegos

TEMARIO

1. Introducción a la Ingeniería de Software
2. Ingeniería de Software en Videojuegos
3. Consejos
- 4. Distribución de Videojuegos 2D**

4. Distribución de Videojuegos 2D

Según el modelo de negocios:

- Advergaming.
- Celulares.
- Desarrollos por Encargo.
- Pago por Descarga.
- Mixtos.

4. Distribución de Videojuegos 2D

Advergaming

- Un mes de desarrollo (en la región).
- 5K-10K USD, costo por mes.
- 2-3 Personas por proyecto.
- Generalmente Web (comúnmente Flash pero también Unity, Silverlight, etc).

datos de BATOVI Games Studio 2008

4. Distribución de Videojuegos 2D



4. Distribución de Videojuegos 2D



4. Distribución de Videojuegos 2D

Celulares

- 5-6 meses de desarrollo.
- 30K-100K USD.
- 4-5 Personas por proyecto.
- J2ME, BREW, Windows Mobile, iPhone, Android, etc.
- Certificación.

datos de BATOVI Games Studio 2008

4. Distribución de Videojuegos 2D



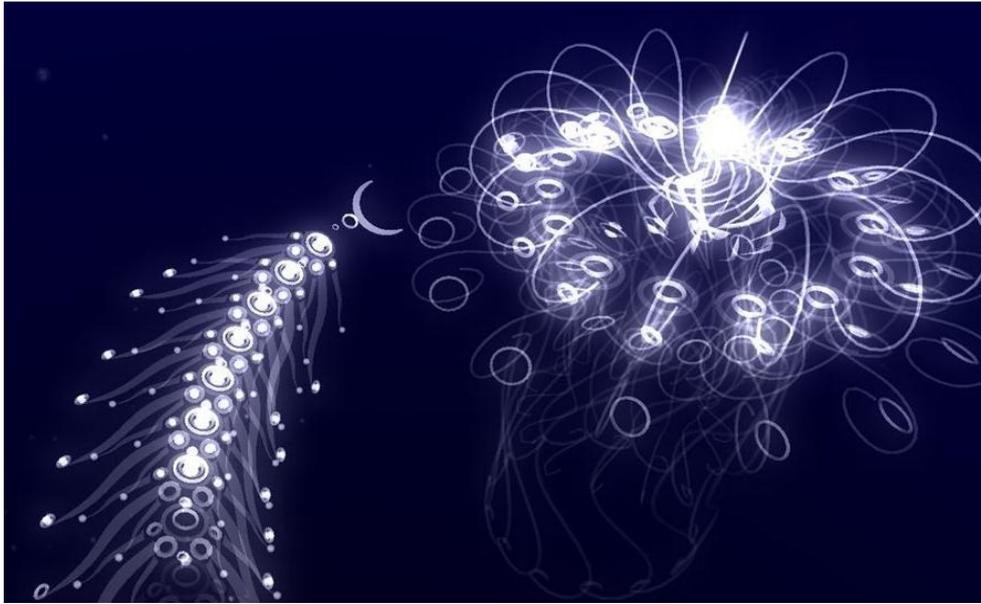
4. Distribución de Videojuegos 2D

Casual Games

- 7-10 meses de desarrollo.
- 60K-150K USD.
- 5-6 Personas involucradas por proyecto.
- Windows & Mac.
- C++, DirectX, OpenGL, Flash, etc.

datos de BATOVI Games Studio 2008

4. Distribución de Videojuegos 2D



Referencias

- **Curso Introducción a la Ingeniería de Software. Facultad de Ingeniería. Universidad de la República. <http://www.fing.edu.uy/inco/cursos/ingsoft/>**
- **Seminario de Tecnologías Interactivas y Videojuegos 2008. Facultad de Ingeniería. Universidad de la República. <http://www.fing.edu.uy/inco/cursos/svti/>**
- **Manifiesto for Agile Software Development. <http://agilemanifesto.org/iso/es/>**
- **Extreme programming and agile processes in software engineering. 5th International Conference, Garmisch-Partenkirchen, Germany. 10/6/2004**
- **Agile Game Development with Scrum. Clinton Keith. Addison-Wesley 2010.**
- **DevMaster - <http://www.devmaster.net>**
- **Gamasutra - <http://www.gamasutra.com/>**
- **Batovi Games Studio - Desarrollo de Videojuegos - Presentación 16/10/2008.**
-

MARIO
024750

× 16

WORLD
1-1

TIME
000



5000

