**UNIVERSIDAD CATÓLICA DE SANTIAGO DEL ESTERO**

Departamento Académico Rafaela

Trabajo práctico N° 1

Carrera: Ing. en Informática

Materia: Sistemas de Información I

Profesor: José Boffelli, Silvana Henzenn

Fecha: 25/08/2015

Alumno: Camila Kopech, Wendy Sclerandi

**1)** Ejemplo de sistema

Empresa de videojuegos.

Testers

Programadores

Diseñadores

Ventas

Proveedores

Clientes

*Objetivo*: Mejorar el diseño de los videojuegos, teniendo en cuenta lo que desean los clientes y así lograr un alcance mundial.

*Elementos*: Director, programadores, diseñadores, testers y ventas.

*Entorno*: Proveedores y clientes.

*Relaciones*:

1. Los clientes solicitan un juego con la parte de ventas.
2. Ventas pasa los requisitos del juego a los diseñadores.
3. Los diseñadores le entregan a los programadores el diseño del juego para que lo codifiquen.
4. Los programadores solicitan servicios a los proveedores.
5. Los proveedores prestan servicios a los programadores.
6. Los programadores pasan el juego a los testers para que lo prueben.
7. Los testers dan una devolución del producto a los programadores.
8. Cuando el juego esté listo, los programadores entregan el juego al sector de ventas para comercializarlo.
9. Ventas entrega el juego terminado a los clientes.

**2)** Subsistema del elemento Ventas para la comprobación de las propiedades Ackoff

Atención al cliente

Publicidad

Jefe de ventas

Clientes

Diseñadores

Programadores

*Relaciones*:

1. Los programadores le entregan el producto al jefe de ventas.
2. El jefe de ventas le entrega el producto a los clientes.
3. Si los clientes tienen alguna consulta o problema, se comunican con atención al cliente.
4. El jefe de ventas coordina el sector de publicidad y de atención al cliente.
5. Si en atención al cliente surge algún problema grave con el producto, lo comunica al jefe de ventas.
6. El jefe de ventas le informa a los programadores y diseñadores los problemas que surjan con el producto por parte de los clientes.

**3)** El dato es una forma mínima de representar la información, son elementos recolectados a partir de observaciones o mediciones para un fin específico. En cambio, la información se trata de un conjunto de datos procesados y relacionados de manera funcional para el fin previsto.

Ejemplo 1:

Una empresa de investigación de mercados recaba los siguientes **datos** entre varios documentos de un posible cliente:

*Banmundi, #38, Roberto Gutiérrez Barrios, Sur 73, 25 años, Sinfonatel, 345893457843, 6-12-89-74.*

De este modo, cada dato pasa por un proceso (análisis, ordenamiento, agregado, etc.), el cuál le da sentido, convirtiéndolos en **información**:

*Nombre: Roberto*

*Apellidos: Gutiérrez Barrios*

*Edad: 25 años*

*Dirección (calle y número): Sur 73, #38*

*Colonia: Sinfonatel*

*Teléfono: 6-12-89-74*

*Banco: Banmundi*

*Cuenta: 345893457843*

Ejemplo 2:

*-*Dato*: 25%*

*-*Información*: El 25% de la población es pobre.*

Ejemplo 3:

-Dato: *5413336060*

-Información: *54: corresponde al código del país.  
1: es el código de la ciudad  
3336060: corresponde al número del abonado.*

**4)** Un **sistema de información** es un conjunto integrado de procesos que resultan de las operaciones realizadas sobre un conjunto de datos estructurados de una organización para cubrir una necesidad u objetivo. Dichas operaciones son almacenar, recuperar, procesar y distribuir los datos.

Las **finalidades** de los sistemas de información son procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes, y otras salidas.

Hay tres **actividades** en un sistema de información que producen la información que las organizaciones necesitan para tomar decisiones, controlar operaciones, analizar problemas y crear nuevos productos o servicios. Estas actividades son:

* Entrada: captura o recolecta datos en bruto tanto del interior de la organización como de su entorno externo.
* Procesamiento: convierte la entrada de datos en una forma más significativa.
* Salida: transfiere la información procesada a la gente que la usará o a las actividades para las que se utilizará.

Los sistemas de información también requieren retroalimentación, que es la salida que se devuelve al personal adecuado de la organización para ayudarle a evaluar o corregir la etapa de entrada.

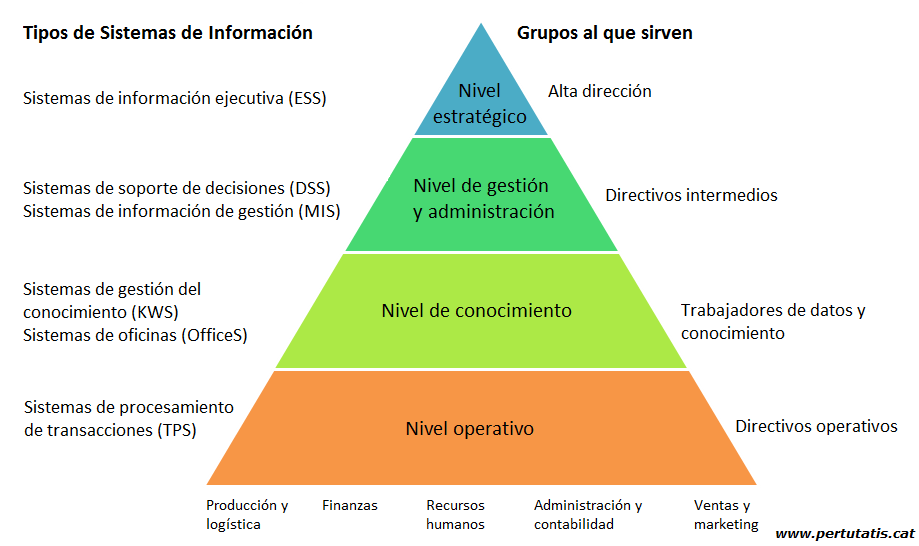
**5)** Un **analista de sistemas** examina y evalúa sistemáticamente los sistemas informáticos de los negocios a través de un examen del procesamiento, de la entrada de datos y la producción de dicha información, con el propósito de mejorar los procesos de una organización.

Las funciones principales son:

1. Planificar el trabajo de análisis y diseño del sistema.
2. Organizar a todos los elementos que intervienen en el proyecto (técnicos de análisis y diseño, programadores, usuarios, equipamiento...). Controlar el trabajo del equipo de diseño para garantizar el cumplimiento de los planes elaborados. Diseñar y utilizar los métodos, técnicas y herramientas más adecuadas para el desarrollo del trabajo colectivo. Estudiar el sistema de dirección y organización e información de la entidad.
3. Diseñar el nuevo sistema informativo, desde un punto de vista funcional, en primera instancia. Representar algorítmicamente los procesos que se realizan en cada tarea funcional del sistema.
4. Diseñar el sistema, descomponiendo el mismo en todos los niveles previstos y con todos los enfoques necesarios. Diseñar la base de datos óptima que utilizará el sistema. Diseñar los documentos o formularios de utilización manual o manual automatizada que requiera el sistema. Diseñar las salidas de la computadora, de la forma más idónea requerida. Elaborar las soluciones a los procedimientos manuales que requiera el sistema. Diseñar los flujos informativos a través de los diferentes elementos del sistema.
5. Proponer y aplicar las medidas de carácter organizativo que se requiera para perfeccionar la actividad de dirección estudiada y para implantar el sistema que se diseña.
6. Estudiar las necesidades y proponer soluciones necesarias para la adquisición del equipamiento computacional requerido y de los medios de transmisión de datos. Estudiar las necesidades del sistema y proponer el software necesario para su aplicación.
7. Calcular los gastos y beneficios del sistema determinando el plazo de recuperación de los mencionados gastos.
8. Elaborar la documentación técnica y de utilización del sistema.
9. Programar, cuando se requiera, la solución automatizada de una determinada tarea, utilizando el software y los métodos requeridos; aplicando las normas de trabajo que estén establecidas dentro del grupo para elaborar y documentar los programas.
10. Revisar los resultados obtenidos por los programas elaborados por los programadores.
11. Elaborar los datos de prueba para comprobar la calidad de los programas, individualmente y en su conjunto. Determinar los métodos más adecuados para implantar y probar el sistema diseñado en la práctica. Entrenar al personal usuario y de operación para demostrar la operación del sistema una vez implantado.
12. Estudiar y aplicar las medidas requeridas para mantener el sistema, acorde a las nuevas necesidades de la entidad. Perfeccionar el mismo.
13. Impartir conferencias, seminarios, etc.; explicando las características del sistema que se diseña y otras cuestiones relativas a la informática y el perfeccionamiento de la dirección.

**6)** Según la función a la que vayan destinados o el tipo de usuario final del mismo, los sistemas de información pueden clasificarse en:

* **Sistema de procesamiento de transacciones** (TPS): Gestiona la información referente a las transacciones producidas en una empresa u organización, también se lo conoce como ‘Sistema de información operativa’. Estos datos son almacenadas en una base de datos que puede servir de apoyo a los otros tipos de sistemas de información.
* **Sistemas de automatización de oficinas** (OAS).- Aplicaciones destinadas a ayudar al trabajo diario del administrativo de una empresa u organización, como procesadores de texto, hojas de cálculo y programas de comunicaciones.
* **Sistemas de información gerencial** (MIS): Orientados a solucionar problemas empresariales en general. Proporciona la información que será empleada en los procesos de decisión administrativos.
* **Sistemas de soporte a decisiones** (DSS): Herramienta para realizar el análisis de las diferentes variables de negocio con la finalidad de apoyar el proceso de toma de decisiones. Estos sistemas permiten a los gerentes obtener respuestas a problemas inesperados y relativamente excepcionales.
* **Sistema experto** (SE): Emulan el comportamiento de un experto en un área específica.
* **Sistemas de información ejecutiva** (EIS): Herramienta orientada a usuarios de nivel gerencial, que permite monitorizar el estado de las variables de un área o unidad de la empresa a partir de información interna y externa a la misma. Es en este nivel cuando los sistemas de información manejan información estratégica para las empresas.
* **Sistema planificación de recursos** (ERP): Su objetivo es la planificación de los recursos de una organización para tener claramente identificado como llegar a los productos finales desde la materia prima. Integran la información y los procesos de una organización en un solo sistema.



**7)** Los participantes en el desarrollo de un sistema de información son los siguientes:

* Usuario: Interactúa con el sistema cuando está finalizado, ya sea para entrar u obtener datos.
* Analista de negocios: Analiza el aspecto comercial del sistema, identificando cómo proveerá el valor comercial, y diseña nuevas políticas y procedimientos al negocio.
* Analista de sistemas: Identifica cómo la tecnología puede mejorar los procesos del negocio, diseña los sistemas de información y los procesos nuevos, y se asegura de cumplir todos los reglamentos y estatutos.
* Programador: Utiliza las especificaciones del diseño de sistemas para escribir y probar códigos de programas e implementa los nuevos sistemas.
* Administrador de base de datos: Define las estructuras de la base de datos, asegura que el diseño sea efectivo y, además, monitorea y hace mejoras al funcionamiento de todas las bases de datos.
* Analista de infraestructura: Identifica e implanta cambios en la infraestructura y asegura el cumplimiento de los estándares.
* Analista de administración de cambios: Desarrolla y ejecuta el plan de administración de cambios y adiestramientos a los usuarios.
* Administrador de proyectos: Trabaja en el proyecto de principio a fin, maneja el equipo de proyectos, desarrolla y monitorea los planes del proyecto, asigna recursos y presupuestos, y actúa como contrato primario para el proyecto completo (enlace entre administración y trabajadores del proyecto).

**8)** Involucrar al usuario a lo largo del proyecto es muy importante para el desarrollo exitoso del sistema de información ya que éste será quien use el sistema finalizado. Incluirlo en el desarrollo tiene la ventaja de que el sistema final que se le entregará será algo que le guste.

En algunas organizaciones es posible que se formen comités de grupos de usuarios, que tienen el control sobre qué se desarrolla y cuándo se implanta. Esto puede traer como desventaja el posible desperdicio de recursos y un mal planeamiento del sistema.

La desventaja está en que el usuario puede tener requisitos que los programadores y analistas no puedan llevar a cabo, por lo que si el usuario no está dispuesto a un cambio de perspectiva, abandonaría el proyecto.

**9)** Los aspectos relacionados a la interfaz de usuario deberían revisarse con éstos durante todo el proceso de creación del sistema ya que son quienes aprueban las especificaciones, implementación, prueba del sistema.

Primeramente, ante el primer diseño del sistema que se cree, para evitar crear algo que al usuario no le guste y por lo tanto, quede en desuso. Luego, cuando el proyecto ya esté bastante encaminado, convendría consultarle al usuario si tiene algún otro requisito que desee agregarle al sistema.

En el caso de que se realicen modificaciones significativas sobre la interfaz de usuario, es conveniente que éste las valide, teniendo en cuenta la opción de realizar un nuevo diseño si no le gusta.

**10)** **El ciclo de vida** se define como la metodología tradicional para desarrollar un sistema de información, el cual divide el proceso de desarrollo del sistema en fases formales para el análisis y diseño del mismo. Estas fases constan en un conjunto de actividades que no necesariamente se tiene que llevar a cabo por separado, sino que varias pueden ocurrir al mismo tiempo y repetirse.



**11)** El ciclo de vida **clásico** de un proyecto consta de los siguientes pasos:

1. **Investigación preliminar**: En esta fase, el analista se encarga de identificar correctamente los problemas, las oportunidades y los objetivos de la empresa, junto con los usuarios y los administradores de sistemas. Consta de 3 partes:
   * *Aclaración de la solicitud*: Examinar las solicitudes para determinar con precisión lo que el solicitante desea.
   * *Estudio de la factibilidad*: Determinar si el trabajo para el proyecto puede realizarse con el equipo y tecnología existente, si se pueden aceptar los costos que implica, y si los usuarios están dispuestos al cambio de sistema.
   * *Aprobación de la solicitud*: Decidir qué proyecto es más importante para llevarlo a cabo, ya que puede haber muchas solicitudes. Luego de aprobarlo, se estiman costos, tiempo para llevarlo a cabo y necesidades de personal.
2. **Determinación de los requerimientos del sistema**: La siguiente fase a la que entra el analista es determinar las necesidades de los usuarios involucrados, mediante el uso de varias herramientas, cuestionarios y entrevistas, para comprender la forma en que interactúan en el contexto laboral con sus sistemas de información actuales y así incluir estas características en el nuevo sistema.
3. **Diseño del sistema**: El diseño lógico de un sistema de información produce los detalles que establecen la forma en la que el sistema cumplirá con los requerimientos identificados durante la fase de análisis. Este diseño indica los datos de entrada, los calculados y los que serán almacenados. Se selecciona la estructura de los archivos y el dispositivo de almacenamiento. Los diseñadores son los responsables de darle a los programadores las especificaciones de software completas y claramente delineadas. La interfaz del usuario se diseña con ayuda de los usuarios para asegurar que el sistema sea perceptible, legible y seguro, así como atractivo y divertido de usar.
4. **Desarrollo del software**: Aquí, el analista trabaja con los programadores para desarrollar el software original requerido y junto con los usuarios una documentación efectiva para el software, incluyendo manuales de procedimientos, ayuda en línea, sitios Web con preguntas frecuentes y archivos Léame. Los programadores diseñan, codifican y eliminan los errores sintácticos de los programas de computadora.
5. **Prueba de sistemas**: Antes de utilizar el sistema de información, se debe probar para asegurarse de que el software no tenga fallas y funciona de acuerdo con las especificaciones. Una parte del procedimiento de prueba es llevado a cabo por los programadores solos; la otra la realizan junto con los analistas de sistemas. El mantenimiento del sistema y la documentación de este mantenimiento empieza en esta fase y se lleva a cabo de manera rutinaria durante toda la vida del sistema de información.
6. **Implantación y evaluación**: En esta última fase del desarrollo, se implanta el sistema, es decir, se verifica y se instala el nuevo equipo, se entrena a los usuarios, se instala la aplicación y se construyen todos los archivos de datos necesarios para utilizarla. Los encargados de desarrollar el sistema deben procurar que el uso inicial del mismo se encuentre libre de errores. Luego de esto, debe poder darse un mantenimiento de las aplicaciones, realizar cambios y modificaciones en el software, archivos o procedimientos para satisfacer las futuras nuevas necesidades de los usuarios.

*Implantación Ascendente:* Se espera que los programadores lleven a cabo primero sus pruebas modulares, luego las pruebas del subsistema, y finalmente las pruebas del sistema mismo.

*Progresión Secuencial:* Consiste en pasar, en orden, de una etapa a la siguiente sólo tras finalizar con éxito las tareas de verificación y validación propias de la etapa. Si resulta necesario, únicamente se da marcha atrás hasta la fase inmediatamente anterior.

Ventajas: Su simple organización hace que resulte sencilla la explicación a usuarios que no están familiarizados con el software. Creación de documentos que ayudan a comprobar cómo avanza el proyecto.

Desventajas: Los proyectos reales raramente siguen el flujo secuencial de actividades que propone este modelo. Normalmente, es difícil para el cliente establecer explícitamente todos los requisitos al comienzo del proyecto. No habrá disponible una versión operativa del sistema hasta llegar a las etapas finales del proyecto, por lo que la rectificación cualquier decisión tomada erróneamente en las etapas iniciales del proyecto supondrá un coste adicional significativo.

**12)** El ciclo de vida **semiestructurado** consiste en:

1. **Encuesta**: Se juntan los requerimientos que el usuario considera necesario para el desarrollo del sistema.
2. **Estudio del hardware**: Creación de los documentos necesarios para llevar a cabo el sistema.
3. **Análisis**: Se analiza la funcionalidad del sistema con herramientas como diagrama de flujo.
4. **Diseño estructurado**: Formación de planes de prueba y diseños de paquetes.
5. **Implantación descendente**: Se codifican y prueban los módulos de alto nivel hacia los de bajo nivel.

La secuencia ascendente de codificación, la prueba de módulos y del sistema se reemplazan por una implantación de arriba hacia abajo, que es un enfoque en el cual los módulos de alto nivel se codifican y prueban primero, seguidos por los de bajo nivel más detallados.

El diseño clásico se reemplaza por el diseño estructurado, que es un enfoque de diseño formal de sistemas.

La implantación descendenteofrece retroalimentación entre el proceso de implementación y el de análisis.

Ventajas: La posibilidad de trabajar en paralelo en algunas etapas. Aporte de herramientas gráficas para la documentación.

Desventajas: Si no se tiene un formato flexible para el intercambio e integración de datos, estas tareas se dificultan. La estructura es muy compleja y genera pérdidas de tiempo en el desarrollo.

**13)** Etapas del ciclo de vida **estructurado**:

1. **Encuesta**: Empieza cuando el usuario solicita que una o más partes de su sistema se automaticen. Se entrevista a los usuarios, posiblemente se desarrolla un diagrama de contexto inicial, se identifica las deficiencias actuales, se establecen metas y objetivos para un nuevo sistema, se determina si es factible automatizar el sistema y se prepara el esquema que se usará para guiar el proyecto.
2. **Análisis de sistemas**: El objetivo es transformar las dos entradas principales, las políticas del usuario y el esquema del proyecto, en una especificación estructurada. Implica modelar el ambiente de usuario.
3. **Diseño**: Asigna el modelo esencial a procesadores adecuados (máquinas o humanos). Para cada tarea se crea una jerarquía de módulos de programas e interfaces para implantar las especificaciones de la etapa de análisis. Implica el desarrollo de un modelo de implantación del usuario, que describe la especificación de la frontera humano – máquina y de la interfaz hombre – máquina, y un modelo de implantación de sistemas para el manejo de los procesos.
4. **Codificación**: Incluye la codificación y la integración de módulos en un esqueleto progresivamente más completo del sistema final.
5. **Generación de pruebas de aceptación**: Se realiza desde el punto de vista del usuario, es decir, desde la especificación estructurada.
6. **Garantía de calidad**: Se aplican las pruebas de aceptación generadas en la etapa anterior al nuevo sistema.
7. **Descripción del procedimiento**: Implica la producción de manuales de usuario.
8. **Conversión de bases de datos**: En general esta actividad requiere como entrada la base de datos actual del usuario.
9. **Instalación**: Las entradas para la Instalación son: los manuales de usuario, la base de datos convertida y el sistema aceptado en el control de calidad.

Se puede implementar en forma *radical* o *conservadora*. La radical es en paralelo y se puede usar cuando es un sistema de investigación, cuando hay una fecha límite de entrega o cuando el usuario interviene mucho en el diseño. La conservadora es en forma secuencial y se utiliza en proyectos grandes o muy costosos, y en proyectos que requieran diseños muy detallados.

Ventajas: Actividades en paralelo. Detección de fallas más rápido.

Desventajas: Si el equipo de desarrollo no está bien organizado, puede que existan problemas de coordinación y se alargue el plazo de desarrollo por la gran cantidad de tapas que posee.

**14)** El ciclo de vida **prototipo** consiste en:

1. Identificar los requerimientos de información que el usuario conoce junto con las características necesarias del sistema.
2. Desarrollar un prototipo que funcione.
3. Utilizar el prototipo anotando las necesidades de cambios y mejoras. Esto expande la lista de los requerimientos de sistemas conocidos.
4. Revisar el prototipo con base en la información obtenida a través de la experiencia del usuario.
5. Repetir los pasos anteriores las veces que sea necesario, hasta obtener un sistema satisfactorio para el usuario.

Una vez el prototipo esté aprobado, se transforma en un prototipo operativo que proporciona las especificaciones finales para la aplicación y se opta por una de las siguientes opciones: el prototipo se convierte en la versión definitiva del sistema deseado, se utiliza la información obtenida con el prototipo operativo para comenzar el desarrollo detallado de un nuevo sistema, se emprende el desarrollo de un nuevo prototipo, o se toma la decisión de abandonar el sistema en su totalidad.

Ventajas: El usuario ve un prototipo del sistema al momento del requerimiento, por lo que sabe cómo va a ser y no se programa sin sentido. Reduce el riesgo de que el proyecto fracase y facilita la especificación de requerimientos de productos que desconocemos.

Desventajas: El cliente puede pensar que el prototipo es el sistema definitivo, ignorando que un prototipo no es un sistema acabado aunque tenga el mismo aspecto externo.

**15)** Las **metodologías ágiles** son una serie de técnicas para la gestión de proyectos que han surgido como contraposición a los métodos clásicos de gestión como [CMMI](http://es.wikipedia.org/wiki/Modelo_de_Capacidad_y_Madurez). Aunque surgieron en el ámbito del desarrollo de software, también han sido exportadas a otro tipo de proyectos.

Todas las metodologías que se consideran ágiles cumplen con el [manifiesto ágil](http://www.agilemanifesto.org/iso/es/) que no es más que una serie de principios que se agrupan en 4 valores:

* Los individuos y su interacción, por encima de los procesos y las herramientas.
* El software que funciona, frente a la documentación exhaustiva.
* La colaboración con el cliente, por encima de la negociación contractual.
* La respuesta al cambio, por encima del seguimiento de un plan.

Características:

* Se apoyan en períodos de producción corto y fijo (sprint) y mucha colaboración dentro del equipo del proyecto.
* El método tiene inherente el cambio. Los requerimientos pueden evolucionar y eso no es un problema significativo.
* El equipo cuenta con poder de decisión.
* Foco en la entrega frecuente (validación con el usuario).
* Participación activa del usuario.
* Desarrollos en forma iterativa.
* Se completa cada función antes de continuar.
* **KANBAN**: Se basa en un sistema de producción que dispara trabajo solo cuando existe capacidad para procesarlo. El disparador de trabajo es representado por tarjetas *Kanban* de las cuales se dispone de una cantidad limitada.

Características:

* Permite comprender el tiempo de tránsito de cada tarea desde que ingresa al sistema hasta que culmina.
* Hay satisfacción del personal para cada miembro que mueve una tarea de una columna a la otra, es decir, completó su tarea.
* Una vez que se ajustan los parámetros de producción, el equipo alcanza un ritmo sustentable.
* El método vincula rápidamente los problemas técnicos del proyecto a los resultados del negocio.

Consta de tres reglas principales:

1. Mostrar el proceso: Consiste en la visualización de todo el proceso de desarrollo, mediante un tablero físico, generalmente, públicamente asequible.
2. Limitar el trabajo en curso: consisten en acordar anticipadamente, la cantidad de ítems que pueden abordarse por cada proceso.
3. Optimizar el flujo de trabajo: Midiendo el tiempo que el ciclo completo de ejecución del proyecto demanda.

Ventajas*:* Produce con calidad y a tiempo lo que genera confianza en los clientes y hace que el equipo mejore constantemente sus procesos y evita demoras.

* **SCRUM**:Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características son:
* El desarrollo de software se realiza mediante iteraciones, denominadas *sprints*, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente.
* Las reuniones a lo largo del proyecto son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

Beneficios:

* Entrega mensual (o quincenal) de resultados lo cual proporciona las siguientes ventajas:

. Gestión regular de las expectativas del cliente y basada en resultados tangibles.

. Resultados anticipados.

. Flexibilidad y adaptación respecto a las necesidades del cliente, cambios en el mercado, etc.

. Gestión sistemática del Retorno de Inversión.

. Mitigación sistemática de los riesgos del proyecto.

* Productividad y calidad.
* Alineamiento entre el cliente y el equipo de desarrollo.
* Equipo motivado.
* **XP (Extreme Programing)**: Está centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Proceso de XP:

1. El clientes define el valor de negocio a implementar
2. El programador estima el esfuerzo necesario para su implementación
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones del tiempo
4. El programador construye ese valor de negocio.

* **FDD** (Feature Driven Development): Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software.

Pasos:

1. Desarrollar el modelo global: Esta etapa define el alcance del proyecto en términos de los que abarca el modelo de dominio de la aplicación.
2. Construir la listade características o de funcionalidades: Aquí en se debe capturar las funcionalidades más notables y de mayor importancia para el cliente.
3. Generar un plan de releases en base a la lista: Aquí podemos ayudarnos de un diagrama de Gantt.
4. Diseñar el software en base a la lista e Implementar la lista: Los dos últimos pasos (4 y 5) son los que se iteran, es decir una vez definido el modelo y el conjunto de características, entonces se empieza a repetir los pasos de diseño e implementación. Es necesario tener en cuenta que el paso de implementación se subdivide también para la realización de pruebas.