ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº1 – DICIEMBRE DE 2007

“DESARROLLO DE SOFTWARE CON CALIDAD PARA UNA EMPRESA”

AUTORIA **CARLOS CABALLERO GONZÁLEZ**
TEMATICA **INFORMÁTICA**
ETAPA **ESO-BACHILLERATO-CFGM(ESI,ASI,DSI)**

Resumen

Se describe la revolución que supuso la incursión de la informática dentro de las empresas, y de un modo generalista los distintos caminos existentes para lograr obtener software de calidad para estas empresas. Se describen los distintos ciclos de vida de un desarrollo software focalizando donde se beneficia el uso de uno y sus perjuicios en casos de emplearlo. Muestra al alumno porque se debe obtener calidad a la hora de desarrollar software y como lograrlo siguiendo una metodología de trabajo.

Palabras clave

Informática, Ingeniería del software, ciclos de vida, metodología, empresas, calidad de software, ciclo formativo de grado superior, ciclo formativo de grado medio, cascada, prototipos, Boehm, software.

1.-INTRODUCCIÓN

Para una empresa la preocupación permanente consiste en mejorar la administración, finanza y producción. Esto ha conducido a la implantación de sistemas automáticos que facilitan las tareas mecánicas y rutinarias además de evitar errores y mejorar la gestión de clientes y contabilidad. Esto repercute directamente en el incremento de la calidad de una empresa.

Esta revolución empresarial ha venido apareciendo desde las últimas tres décadas en las que poco a poco se han ido integrando los sistemas informáticos a los sistemas empresariales. La experiencia de las empresas que se han adaptado a las nuevas tecnologías con adelanto a sus rivales ha dejado claro que la adaptación a las nuevas tecnologías en una empresa es un sello de calidad.

Por tanto el desarrollo de estas herramientas que son las causantes de la revolución empresarial de los últimos tiempos no pueden emplear métodos “artesanales” para la elaboración de las herramientas. Es por esto, que surge la *ingeniería del software*.

El término de ingeniería del software ha tenido diversas definiciones a lo largo de la historia. Estas definiciones son de hace más de veinte años y aún siguen teniendo un semántica correcta. No obstante es en los últimos años cuando la ingeniería del software ha tomado un cariz de importancia debido a la

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº1 – DICIEMBRE DE 2007

necesidad de calidad en las empresas y a los avances tecnológicos. Las definiciones más importantes de ingeniería del software a lo largo de la historia han sido:

1. **Ingeniería de Software** trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener de modo rentable que sea fiable y trabaje en máquinas reales (Bauer, 1972).
2. **Ingeniería de Software** es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar (funcionar) y mantenerlos (Bohem, 1976).
3. **Ingeniería de Software** es el estudio de los principios y metodologías para desarrollo y mantenimiento de sistemas de software (Zelkovitz, 1978).

2.- SOFTWARE

El software es la parte lógica por la que están compuestos los sistemas informáticos, siendo la parte física el hardware. Para poder entender perfectamente lo más adelante expuesto lo fundamental es proporcionar una buena definición:

Definición

Se puede definir software como el conjunto de instrucciones, estructuras de datos y documentación que realizan el procedimiento o control requerido.

Las instrucciones cuando se ejecutan, proporcionan la funcionalidad deseada. Las estructuras de datos son los elementos que facilitan a las instrucciones manipular adecuadamente la información y finalmente los documentos son los encargados de describir el desarrollo, uso, instalación y mantenimiento de las diferentes herramientas software.

Características

Las características fundamentales del software son tres:

1. **Se desarrolla, no se fabrica en el sentido clásico.** El software no tiene una cadena de fabricación clásica en la que primero se pasa por una etapa de fabricación de componentes y posteriormente de montaje. El software se desarrolla una sola vez por un equipo de trabajo y posteriormente estaríamos en la etapa de venta que englobaría al marketing y producción.
2. **No se estropea.** El software a diferencia del hardware no se deteriora con el paso del tiempo por deterioramiento de sus componentes. Si el software se elabora con suficiente calidad esta calidad se mantendrá constante.
3. **Se construye a medida, en vez de ensamblar componentes existentes.** Aunque existen productos genéricos que se venden a los usuarios como soluciones adaptables a sus problemas tradicionalmente el software es elaborado para solventar un problema concreto, es decir, es elaborado a medida al problema que se pretende resolver.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº1 – DICIEMBRE DE 2007

3.- CICLO DE VIDA

El ciclo de vida cuando se elabora software o proyecto informático es el conjunto de etapas y estados por los que pasa el desarrollo desde que se plantea como necesidad o problema, por parte de un cliente, hasta que se da por terminado y se considera como una solución completa, correcta y estable (que resuelve el problema inicial).

Las etapas o fases principales del ciclo de vida de una aplicación son las siguientes:

- **Análisis.**
  - Es la etapa inicial, en esta etapa se persigue extraer el conocimiento del problema que se desea resolver. Extrayendo sus características, detalles, limitaciones,… En la etapa de análisis se descompone el problema en partes para obtener un conjunto de subproblemas que sean comprendidos y de fácil resolución.
  - En resumen se establece **QUÉ** se quiere desarrollar en función de las necesidades del usuario y las distintas restricciones del desarrollo como pueden ser factores tecnológicos, financieros o de recursos humanos.
  - El análisis del sistema se realiza teniendo presente los objetivos de:
    - **Identificar las necesidades del cliente**
      - En esta etapa se hace necesario una comunicación fluida entre el usuario o cliente y el analista para poder identificar las necesidades del mismo. Esta tarea se lleva a cabo a través de entrevistas y cuestionarios.
    - **Evaluar la viabilidad del proyecto**
      - En esta etapa se determina la posibilidad de realizar con garantías el proyecto, encontramos cuatro áreas de interés:
        - **Viabilidad económica.** Estimar los costes de desarrollo y los beneficios finales.
        - **Viabilidad técnica.** Posibilidad de realizar el proyecto con las herramientas que se disponen.
        - **Viabilidad legal.** No incurrir en violaciones legales en el desarrollo de del proyecto.
- **Diseño**
  - El objetivo de esta etapa es decidir **CÓMO** resolver cada uno de los subproblemas identificados en la etapa de análisis y **CÓMO** integrar todas las soluciones diseñadas en una solución global.
- **Implementación**
  - En esta fase se realiza la codificación, se realiza la programación de la solución diseñada, en el lenguaje de programación y plataforma elegida a tal efecto teniendo en cuenta las restricciones obtenidas en la etapa de análisis.
- **Pruebas**
  - El objetivo de esta fase es garantizar el correcto funcionamiento de las aplicaciones desarrolladas en la etapa de implementación, así como su adecuación a los requisitos y necesidades expresadas por el cliente o usuario en la etapa de análisis.
  - Estos objetivos se plasman en dos aspectos que son complementarios entre sí

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº1 – DICIEMBRE DE 2007

- **La verificación.** Consiste en comprobar el correcto funcionamiento del código programado.
- **La validación.** Consiste en asegurar que la aplicación que se ha obtenido es el producto correcto.

o Se realizan dos tipos de pruebas distintas

- **Pruebas unitarias.** Para comprobar que cada módulo hace lo que tiene que hacer sujeto a las restricciones.
- **Pruebas de integración.** Para comprobar que los módulos funcionan de manera correcta entre sí.

- **Implantación**
  - o En esta fase se implanta el proyecto en el entorno que se va a emplear esta etapa consiste en
    - Instalación
    - Transición desde el sistema anterior
    - Eliminación del sistema anterior
    - Formación de los usuarios del sistema
- **Mantenimiento**
  - o Aunque terminada la fase de implantación se cierra el proyecto, es necesaria la fase de mantenimiento del software durante un tiempo. No obstante tradicionalmente se llevan a cabo las siguientes etapas
    - **Correctivo.** Corrige los errores no detectados en la fase de construcción.
    - **Adaptativo.** Adapta el programa a nuevas características y/o cambios en las normativas.
    - **Perfectivo.** Añade nuevas características al software.
    - **Preventivo.** Se realizan cambios en el software para facilitar el mantenimiento de futuras funcionalidades.

Como se puede observar existen una gran cantidad de etapas en el ciclo de vida y estas consumen un gran tiempo en el desarrollo de software de calidad. No obstante, paralelamente se realizan actividades que complementan a cada etapa y ayudan a garantizar la integridad y coherencia en el proceso de este modo se consigue ganar calidad en el desarrollo de software. Las tareas paralelas que se desarrollan se pueden clasificar en las siguientes:

- **Gestión de cambios.** Cuando en el desarrollo se produce un cambio, éste se propaga a lo largo de las distintas etapas del ciclo de vida. Para controlar estos cambios y las repercusiones que estos producen en los módulos y en el sistema se debe realizar un seguimiento y control de cambios
- **Gestión de configuración.** Al comienzo del ciclo de vida se define una configuración inicial o básica de los recursos necesarios (Hardware y software). Esta configuración va evolucionando a lo largo del desarrollo, por lo que se deben gestionar los cambios que se produzcan en ellos.
- **Gestión de la documentación.** Agrupa a las actividades dedicadas a planificar, diseñar, editar, producir, distribuir y mantener los documentos necesarios para los desarrolladores y los usuarios.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº1 – DICIEMBRE DE 2007

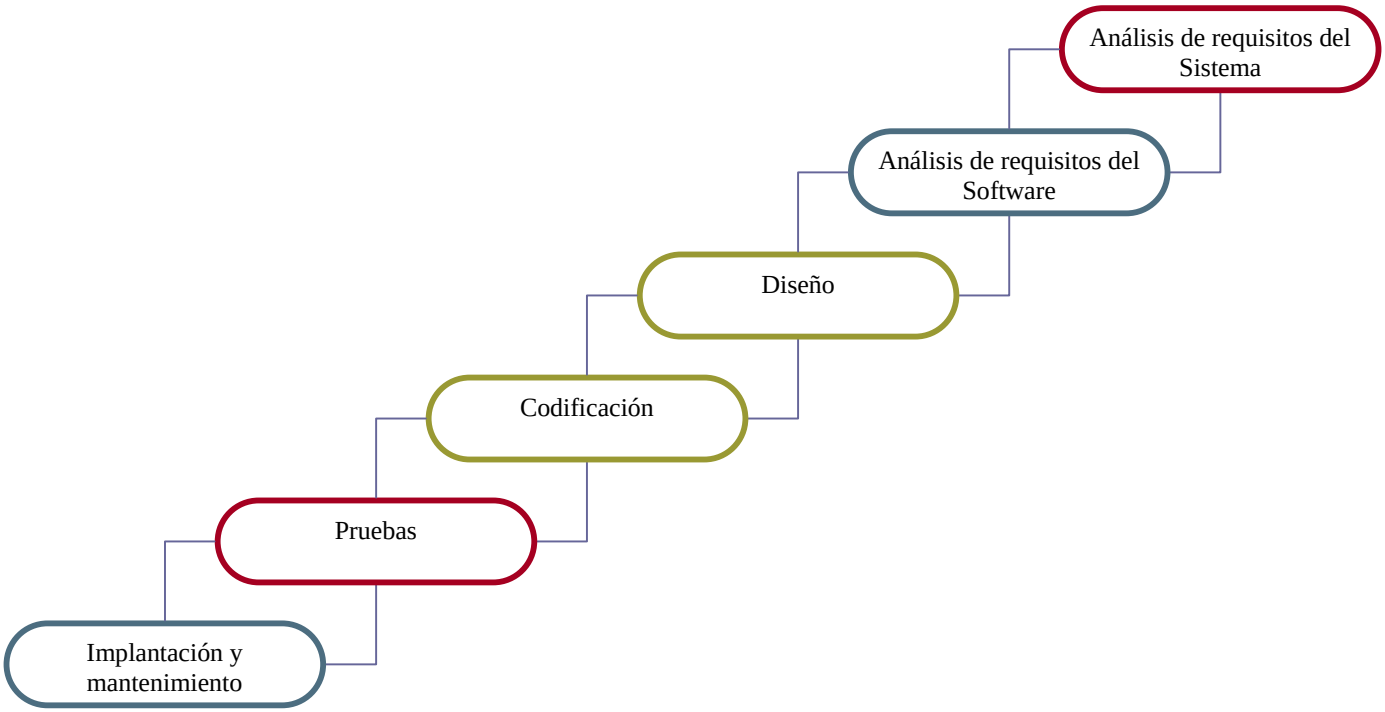
- **Gestión de la calidad.** Está formado por un conjunto de técnicas y procedimientos que garantiza que el producto que se va construyendo no se aparta de los criterios y estándares de calidad adoptados en la planificación inicial y especificación de requisitos del sistema.

4.- TIPOS DE CICLOS DE VIDA

Tal y como hemos descrito anteriormente todo proyecto de desarrollo de software debe tener un ciclo de vida tal y como hemos descrito. Sin embargo, estas etapas se pueden entender y llevar a cabo de formas diferentes, según las condiciones concretas de cada proyecto. Lo cual da lugar a que existan distintos tipos de ciclos de vida.

- **Ciclo de vida clásico o en cascada**
  - Las etapas del ciclo de vida en cascada se agrupan en tres grandes fases
    - **Definición del problema**. Que incluye el análisis de requisitos del sistema y de requisitos de software.
    - **Desarrollo**. Que abarca el diseño, la programación y las pruebas.
    - **Mantenimiento**.
  - Se denomina ciclo de vida en cascada porque se va pasando, “cayendo”, de una fase a la siguiente de manera lineal, lo que significa que cada fase empieza cuando se ha terminado la fase anterior.
  - Al final de cada fase se puede realizar una revisión del proyecto.
  - En ocasiones, por circunstancias del proyecto, se puede dejar una etapa sin terminar y pasar a la siguiente, para regresar más tarde a completarla. En otras ocasiones debido a decisiones tomadas posteriormente en el tiempo obligan a modificar etapas ya terminadas y es porque el ciclo de vida en cascada contempla la posibilidad de volver atrás desde cualquier etapa, este proceso se llama retroalimentación.
  - El principal problema del modelo en cascada es que se retrasa en gran medida la entrega del proyecto a los clientes o usuarios finales.
  - Para pasar de una fase a otra hay que conseguir TODOS los objetivos de la fase previa.

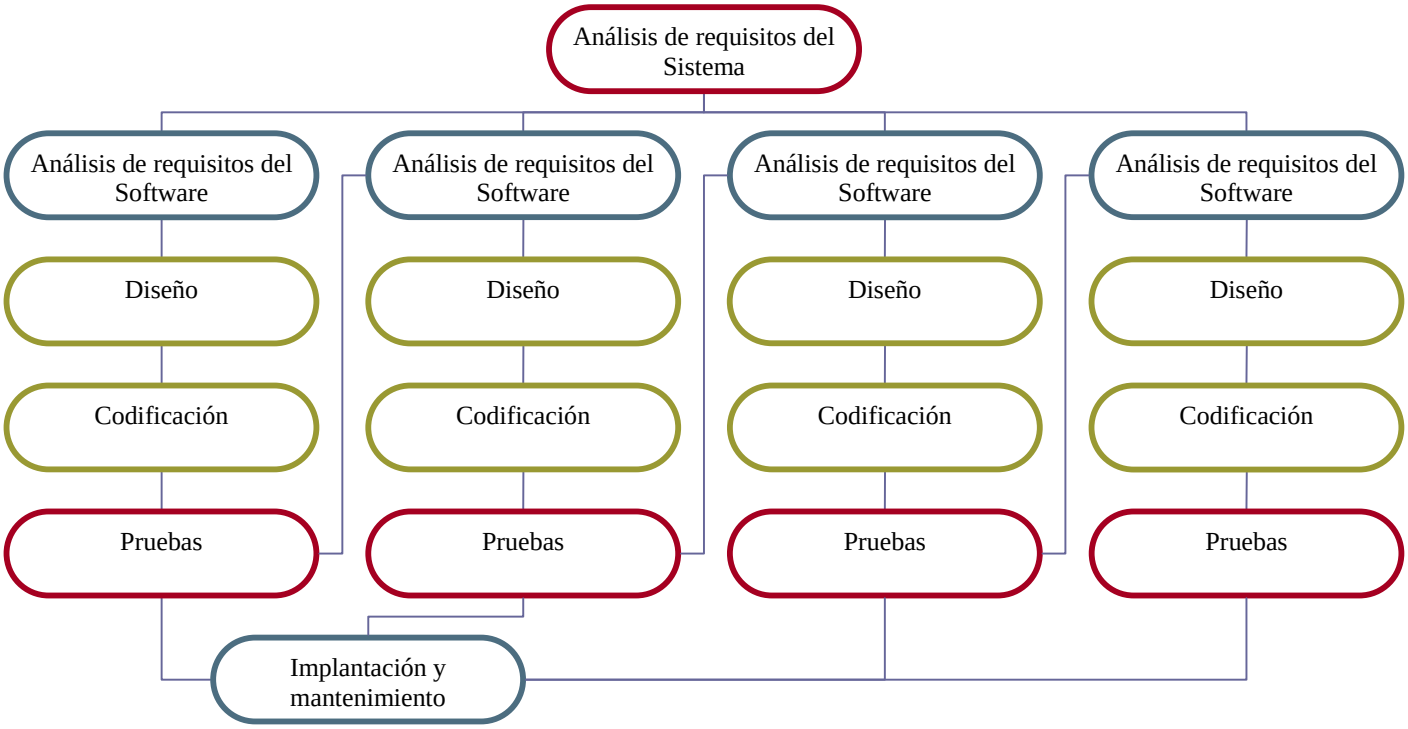
ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº1 – DICIEMBRE DE 2007



o

- **Modelo incremental**
  - El modelo incremental realiza una secuencia de pasos de desarrollo no lineal, mediante el cual se va creando el desarrollo software añadiéndole componentes funcionales (incrementos).
  - En cada paso se actualiza el sistema con nuevos requisitos, se hace uso de una versión previa a la que se le van añadiendo nuevas funcionalidades obteniendo un software final que será el resultado del último refinamiento.
  - Algunas de las ventajas son
    - Se evitan proyectos largos y se entrega “algo de valor” para el cliente con cierta frecuencia.
    - El usuario se involucra más en el desarrollo software.
    - Se encuentran errores o desacuerdos antes debido a que el cliente o usuario puede probar las funcionalidades antes.
  - Algunas de las desventajas de este tipo de ciclo de vida son
    - Difícil de evaluar el coste total a priori debido a que hay que realizar diversos refinamientos.
    - Requiere de gestores experimentados.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº1 – DICIEMBRE DE 2007



- **Modelo en Espiral o de Boehm**
  - Este modelo se distribuye en cuatro grandes etapas:
    - **Planificación.** De acuerdo con el cliente se obtienen los requisitos siguientes:
      - **Objetivos a conseguir.**
      - **Restricciones a aplicar.**
    - **Análisis de riesgo.** Se definen riesgos basándose en los requisitos anteriores. SE decido el modo de resolverlos y las posibles alternativas a cada uno de ellos. Como resultado se decide continuar o no con el desarrollo.
    - **Ingeniería.** Es la fase de análisis propiamente dicha, se crea un prototipo de acuerdo con las especificaciones obtenidas hasta ese momento, el modelo de prototipo se utiliza cuando se tienen los objetivos generales del sistema pero falta definición de los requisitos de entrada, proceso y/o salida.
    - **Evaluación por el cliente.** El usuario evalúa el producto obtenido, y según el resultado puede dar comienzo otro ciclo en el que se modifica la planificación en función de los datos y valoraciones aportados por el cliente. Esta nueva planificación obliga a un nuevo análisis de riesgos, a partir del cual se crea un prototipo de segundo nivel que es presentado al cliente para su evaluación. Se repite este proceso las veces que sea necesario hasta obtener el sistema final.
  - Recibe el nombre por la forma circular y creciente en la que se van sucediendo las distintas etapas, además en cada vuelta del ciclo de vida cada etapa es más compleja, comprende más trabajo y consume más recursos pero igualmente está más cercana a la solución deseada.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº1 – DICIEMBRE DE 2007

- Este modelo usa un enfoque evolutivo, permitiendo al desarrollador y al cliente entender y reaccionar a los riesgos en cada nivel (ciclo completo).
- Este tipo de ciclo de vidas tiene los siguientes inconvenientes
  - Se requiere una considerable habilidad para valorar el riesgo.
  - Puede ser difícil de convencer al cliente de que el enfoque evolutivo es controlable y además de su colaboración.

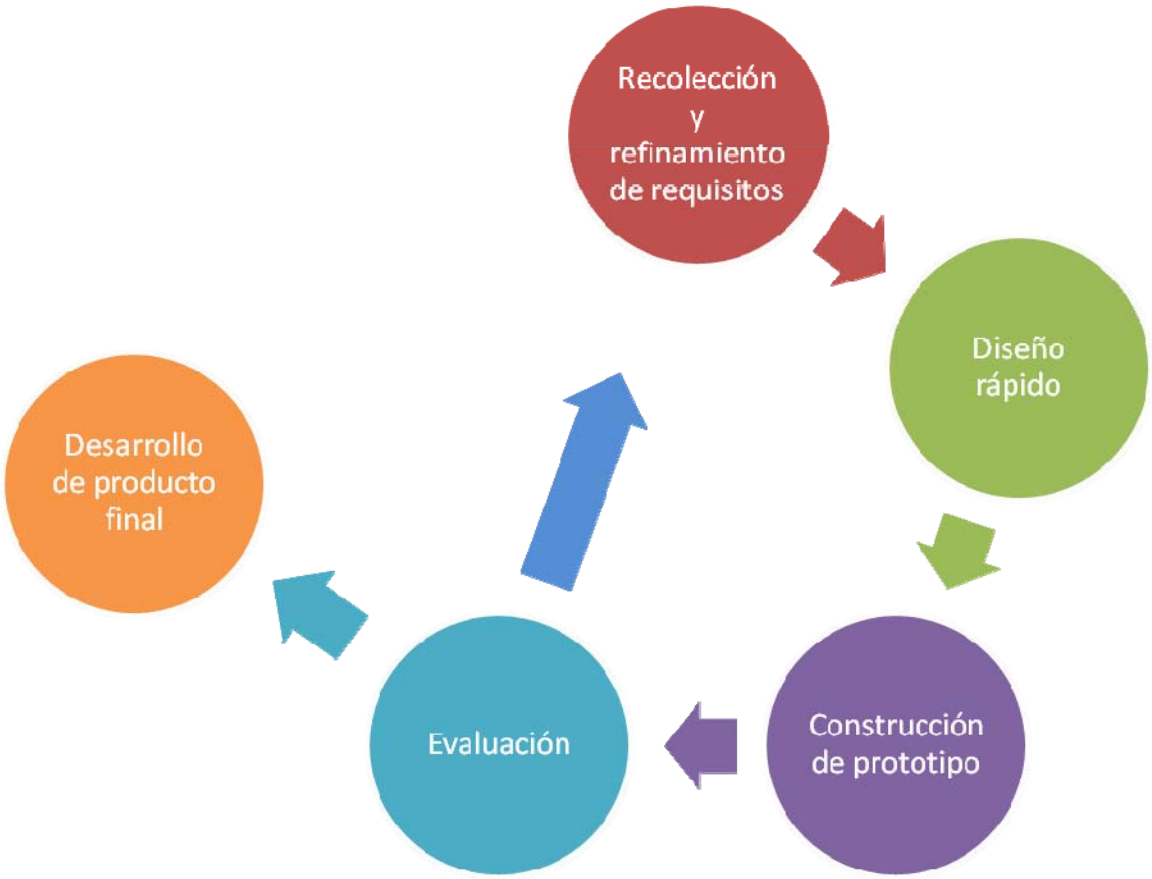
- **Ciclo de vida en prototipos**
  - Este modelo se basa en la construcción de prototipos del producto software que se irán refinando poco a poco. Hace una simulación de la realidad, se usa el prototipo como simulador del resultado final.
  - Estos prototipos se construyen con dos finalidades totalmente distintas:
    - **Prototipos desechables.** Son los prototipos que se construyen y posteriormente son desechados del desarrollo software.
    - **Prototipo cíclico.** Es el prototipo elegido que se va mejorando poco a poco hasta llegar al resultado final.
  - Las actividades que se realizan en este ciclo de vida son:

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº1 – DICIEMBRE DE 2007

- **Recolección y refinamiento de requisitos.** Se identifican los requisitos conocidos y se perfilan las áreas en donde sea necesaria una mayor definición.
- **Diseño rápido.** Se realiza un diseño rápido que se enfoca sobre la representación de los aspectos visibles al usuario.
- **Construcción de prototipos.** Se construye el prototipo, que es evaluado por el cliente/usuario y se utiliza para refinar los requisitos del software a desarrollar.
- **Desarrollo del producto final.**

o Con este modelo de ciclo de vida conseguimos valiosos beneficios que son.

- Reduce el riesgo de construir productos que no satisfagan las necesidades de los usuarios
- Reduce costos y aumenta la probabilidad de éxito.



5.- METODOLOGÍA.

Una metodología es el camino para desarrollar software de una manera sistemática y de ese modo ganar calidad a la hora de desarrollar software. Para conseguir este objetivo algunas metodologías

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº1 – DICIEMBRE DE 2007

definen una estructura jerárquica de pasos a seguir, organizados en fases, módulos, actividades y tareas.

Las tareas son las actividades elementales que se deben realizar en cada una de las fases, para cada tarea se identifica un procedimiento. Los procedimientos indican la forma de ejecutar la tarea. Las actividades concretan las acciones que se deben llevar a cabo en cada una de los módulos. Los módulos son cada una de las subetapas en las que se dividen las fases. Puede ser una o más de una. Y describen el trabajo a desarrolla en cada una de las fases. Las fases son las primeras divisiones del proyecto.

Las técnicas son los apoyos formales para realizar las tareas, indican los medios y pasos a seguir para su ejecución.

Por tanto las herramientas software realizan la automatización de las técnicas, y ayudan a la ejecución de las mismas.

6.- BIBLIOGRAFÍA

1. Piattini M.G y García F.O. (2003). *Calidad en el desarrollo y mantenimiento del software,* Editorial RA-MA.
2. Minguet J.M y Hernández J.F. (2003). *La calidad del software y su medida,* Universitaria Ramón Areces.

Autoría

· CARLOS CABALLERO GONZÁLEZ
· CENTRO: I.E.S MIGUEL ROMERO ESTEO. MÁLAGA
· E-MAIL: CARLOS.CABALLERO.GONZALEZ@GMAIL.COM