



Universidad  
Católica del  
Uruguay

# Procesos de la Ingeniería de software

Catalina Fellegi  
Gabriela Sasco

# Contenido

- 1 INTRODUCCIÓN ..... 4
  - 1.1 ¿POR QUÉ CONTAR CON UN PROCESO DE SOFTWARE? ..... 4
- 2 PROCESO DE SOFTWARE ..... 4
  - 2.1 DEFINICIONES DE PROCESO DE SOFTWARE ..... 4
  - 2.2 EL PROCESO DE DESARROLLO DEL SOFTWARE..... 5
    - 2.2.1 Esquema del desarrollo de software ..... 6
    - 2.2.2 Proceso de Software y su relación con la Ingeniería de Software ..... 7
- 3 MITOS DEL SOFTWARE ..... 7
  - 3.1 MITOS DEL CLIENTE ..... 7
  - 3.2 MITOS DE LOS DESARROLLADORES..... 8
  - 3.3 MITOS DE GESTIÓN ..... 8
- 4 MODELO DE PROCESO DE LA IEEE 1074 ..... 9
  - 4.1 SELECCIÓN DE CICLO DE VIDA..... 10
    - 4.1.1 ¿Qué es el ciclo de vida?..... 10
    - 4.1.2 Selección del ciclo de vida ..... 10
    - 4.1.3 Diferencias entre: Proceso y Ciclo de vida ..... 10
  - 4.2 PROCESO DE GESTIÓN DE PROYECTO ..... 10
    - 4.2.1 Iniciar el proyecto ..... 10
      - 4.2.1.1 Iniciación..... 10
      - 4.2.1.2 Componentes del proceso: Iniciación del proyecto ..... 11
    - 4.2.2 Proceso de gestión de proyecto ..... 11
      - 4.2.2.1 Componentes del proceso de Seguimiento y control ..... 12
    - 4.2.3 Gestión de la calidad del software ..... 12
      - 4.2.3.1 Definición de calidad ..... 12
      - 4.2.3.2 Gestión de calidad de software..... 13
      - 4.2.3.3 Componentes del proceso: Gestión de calidad de software ..... 13
  - 4.3 PROCESOS ORIENTADOS AL DESARROLLO DE SOFTWARE ..... 13
    - 4.3.1 Procesos de pre-desarrollo ..... 13
      - 4.3.1.1 Proceso de exploración de conceptos ..... 13
      - 4.3.1.2 Componentes del proceso: Exploración de conceptos ..... 14
      - 4.3.1.3 Procesos de asignación del sistema ..... 14
      - 4.3.1.4 Componentes del proceso: Asignación del sistema ..... 15
    - 4.3.2 Procesos de desarrollo ..... 15
      - 4.3.2.1 Procesos de requisitos ..... 15
      - 4.3.2.2 Componentes del proceso: Requisitos ..... 16
      - 4.3.2.3 Proceso de diseño ..... 16
      - 4.3.2.4 Componentes del proceso: Diseño ..... 18
      - 4.3.2.5 Proceso de implementación ..... 18
      - 4.3.2.6 Componentes del proceso: Implementación ..... 18
    - 4.3.3 Proceso de post-desarrollo ..... 18
      - 4.3.3.1 Proceso de operación y soporte ..... 18
      - 4.3.3.2 Proceso de instalación ..... 19
      - 4.3.3.3 Componentes del proceso: Instalación ..... 19
      - 4.3.3.4 Proceso de operación y soporte ..... 19
      - 4.3.3.5 Componentes del proceso: Operación y soporte ..... 19
      - 4.3.3.6 Proceso de mantenimiento ..... 19
      - 4.3.3.7 Componentes del proceso: Mantenimiento ..... 20
      - 4.3.3.8 Proceso de Retiro..... 20
      - 4.3.3.9 Componentes del proceso: Retiro ..... 20
  - 4.4 PROCESOS INTEGRALES DEL PROYECTO..... 20
    - 4.4.1 Proceso de verificación y validación ..... 21
      - 4.4.1.1 Componentes del proceso: Verificación y validación ..... 21
    - 4.4.2 Proceso de gestión de la configuración ..... 21
      - 4.4.2.1 Componentes del proceso: Gestión de configuración ..... 21
    - 4.4.3 Proceso de desarrollo de documentación ..... 22
      - 4.4.3.1 Componentes del proceso: Documentación ..... 22

4.4.4 Proceso de formación .....22

4.4.4.1 Componentes del proceso: Formación.....22

# 1 Introducción

En un mundo de cambios constantes y competencia global, las organizaciones de desarrollo de software son presionadas a alcanzar mayor eficiencia con menores costos. Para poder lograr este objetivo, es necesario adoptar una forma de trabajo que permita entender, controlar, comunicar, mejorar, predecir y certificar el trabajo realizado.

## 1.1 ¿Por qué contar con un proceso de software?

Hasta hace poco tiempo, la producción de software era realizada con un enfoque artesanal, a diferencia de un enfoque industrial. En los últimos años, ante la constante presencia de proyectos fallidos, y con el objetivo de mejorar la calidad de los productos, las organizaciones introdujeron la aplicación de Ingeniería de software.

Entre otras tareas se formalizó el enfoque de ingeniería de producto para desarrollar software, incorporando marcos de trabajo que las ayuden hacer las cosas de la manera más eficiente.

Cabe destacar que seleccionar un proceso de software implica introducir varios elementos organizacionales de gestión y de organización del proyecto, los cuales deben aplicados y exigidos por quienes sean responsables del mismo.

## 2 Proceso de Software

Un proceso de desarrollo de software involucra a personas, estructuras de organización, reglas, políticas, actividades a y sus procedimientos, componentes de software, metodologías, y herramientas utilizadas. De ser efectivo, habilita a la organización a incrementar su productividad al desarrollar software. De esta forma:

- Permite **estandarizar** esfuerzos, promover reuso, repetición y consistencia entre proyectos.
- Provee la oportunidad de introducir **mejores prácticas** de la industria.
- Permite entender que las **herramientas** deben ser utilizadas para soportar un proceso.
- Establece la base para una mayor **consistencia** y mejoras futuras.

Además de esto, un proceso de software mejora los esfuerzos de mantenimiento y soporte permitiendo definir:

- Cómo manejar los cambios y liberaciones a sistemas de software existentes.
- Cómo lograr la transición del software a la operación, y cómo ejecutar los esfuerzos de operación y soporte.

Es necesario contar con un proceso de software cuya funcionalidad esté probada en la práctica, y que se encuentre personalizado para que cumpla con las necesidades específicas del proyecto.

### 2.1 Definiciones de Proceso de Software

Siempre que se deba alcanzar algún fin deseado será necesario ejecutar una serie de acciones, las cuales habrán de poseer un orden, dependencias, roles responsables, resultados, tiempos de ejecución y herramientas de apoyo. En estos casos, estaremos hablando de procesos, que pueden ser predefinidos y personalizados.

A continuación se presentarán definiciones de diferentes autores:

- *“El conjunto completo de actividades de ingeniería de software necesarias para transformar los requerimientos del usuario en software”* [Humphrey]
- *“Un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto software.”* [Fugetta]
- *“El proceso de software es un conjunto de actividades y resultados asociados que conducen a la creación de un producto de software”.* [Sommerville]

Por su parte, la expresión "Proceso de Software" se ha usado para referirse al **conjunto de procesos, con sus tareas, actividades, funciones, tareas, metodologías, herramientas y recursos que ejecutados de**

cierta manera garantizan determinados niveles de garantía de que el producto final será o estará "bien" definido y construido.

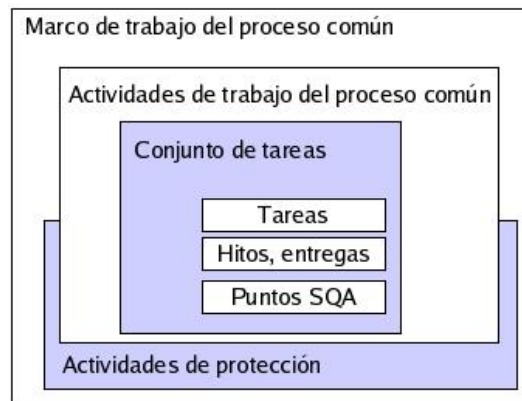


Figura 1 – Procesos de Software

Pressman [1] caracteriza un proceso de desarrollo de software como se muestra en la Figura 1. Los elementos involucrados se describen a continuación:

- **Un marco común del proceso**, definiendo un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos de software, con independencia del tamaño o complejidad.
- **Un conjunto de tareas**, cada uno es una colección de tareas de ingeniería del software, hitos de proyectos, entregas y productos de trabajo del software, y puntos de garantía de calidad (SQA), que permiten que las actividades del marco de trabajo se adapten a las características del proyecto de software y los requisitos del equipo del proyecto.
- **Las actividades de protección**, tales como garantía de calidad del software, gestión de configuración del software y medición, abarcan el modelo del proceso. Las actividades de protección son independientes de cualquier actividad del marco de trabajo y aparecen durante todo el proceso.

## 2.2 El proceso de desarrollo del software

Un proceso de desarrollo de software tiene como propósito la **producción eficaz y eficiente de un producto** software con el fin de que reúna los requisitos del cliente. Dicho proceso se muestra en la Figura 2, en términos globales:

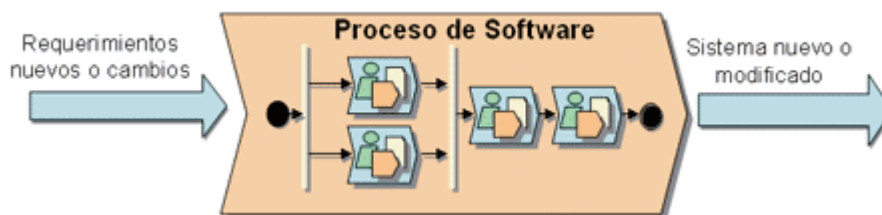


Figura 2: proceso de desarrollo de software.

A pesar de que un proyecto de desarrollo de software es equiparable en muchos aspectos a cualquier otro proyecto de ingeniería, en el desarrollo de software existe una serie de desafíos adicionales, relativos esencialmente a la naturaleza del producto obtenido. A continuación, se explican algunas particularidades asociadas al desarrollo de software y que influyen en su proceso de construcción:

- Un producto software en sí es complejo. Es prácticamente inviable conseguir un 100% de confiabilidad de un programa por pequeño que sea. Existe una inmensa combinación de factores que impiden una verificación exhaustiva de las todas posibles situaciones de ejecución que se puedan presentar (entradas, valores de variables, datos almacenados, software del sistema, otras aplicaciones que intervienen, el hardware sobre el cual se ejecuta, etc.).
- Un producto software es intangible y por lo general muy abstracto, esto dificulta la definición del producto y sus requisitos, sobre todo cuando no se tiene precedentes en productos software similar. Esto hace que los requisitos sean difíciles de consolidar tempranamente. Así, los cambios en los requisitos son inevitables, no sólo después de entregado en producto sino también durante el proceso de desarrollo.

- El proceso de desarrollo de software *no es único*. No existe un proceso de software universal que sea efectivo para todos los contextos de proyectos de desarrollo.

A pesar de la variedad de propuestas de proceso de software, existe un conjunto de actividades fundamentales que se encuentran presentes en todos ellos:

- **Especificación de software:** Se debe definir la funcionalidad y restricciones operacionales que debe cumplir el software.
- **Diseño e Implementación:** Se diseña y construye el software de acuerdo a la especificación.
- **Validación:** El software debe validarse, para asegurar que cumpla con lo que quiere el cliente.
- **Evolución:** El software debe evolucionar, para adaptarse a las necesidades del cliente.

Además de estas actividades fundamentales, Pressman menciona un conjunto de “actividades integrales”, que se aplican a lo largo de todo el proceso del software, a saber:

- Seguimiento y control de proyecto de software.
- Revisiones técnicas formales.
- Garantía de calidad del software.
- Gestión de configuración del software.
- Preparación y producción de documentos.
- Mediciones.
- Gestión de riesgos.

### 2.2.1 Esquema del desarrollo de software

Otra perspectiva utilizada para determinar los elementos del proceso de desarrollo de software es establecer las relaciones entre elementos, que permitan responder **Quién** debe hacer **Qué**, **Cuándo** y **Cómo** debe hacerlo.

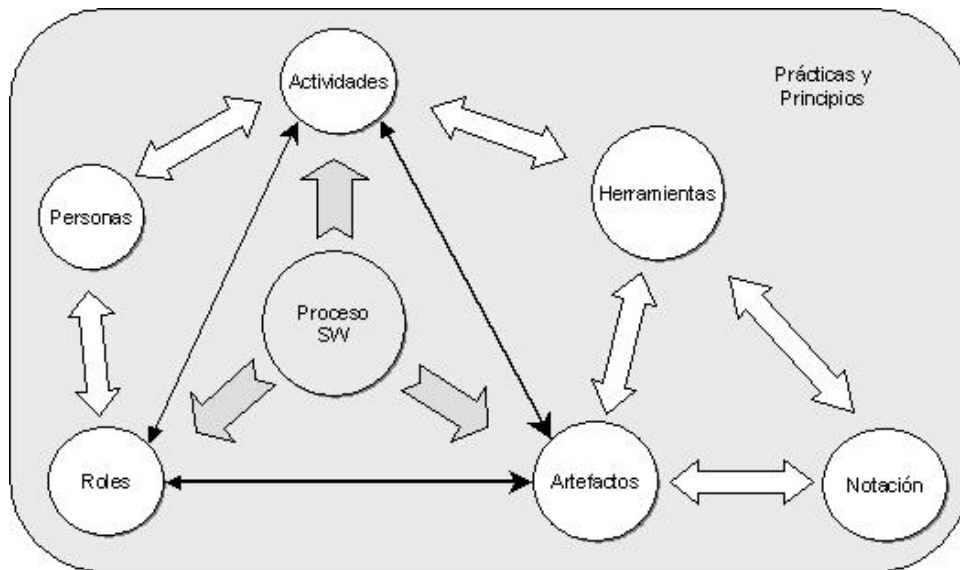


Figura 3: Relación entre elementos del proceso del software

En la Figura 3 se muestran los elementos de un proceso de desarrollo de software y sus relaciones. Así, las interrogantes se responden de la siguiente forma:

- **Quién:** Las Personas participantes en el proyecto de desarrollo desempeñando uno o más Roles específicos.
- **Qué:** Un Artefacto<sup>1</sup> es producido por un Rol en una de sus Actividades. Los Artefactos se describen soportando ciertas Notaciones.

<sup>1</sup> Un artefacto es una pieza de información que (1) es producida, modificada o usada por el proceso, (2) define un área de responsabilidad para un rol y (3) está sujeta a control de versiones. Un artefacto puede ser un modelo, un elemento de modelo o un documento.

- **Cómo y Cuándo:** Las Actividades son una serie de pasos que lleva a cabo un Rol durante el proceso de desarrollo. El avance del proyecto está controlado mediante hitos que establecen un determinado estado de finalización de ciertos Artefactos.

Las Prácticas y Principios enfatizan ciertas actividades y/o la forma como deben realizarse, por ejemplo: desarrollar iterativamente, gestionar requisitos, desarrollo basado en componentes, modelar visualmente, verificar continuamente la calidad, gestionar los cambios, etc.

### 2.2.2 Proceso de Software y su relación con la Ingeniería de Software

Todo proyecto de ingeniería posee fines ligados a la obtención de un producto, proceso o servicio que es necesario generar a través de diversas actividades. Algunas de estas actividades pueden agruparse en fases porque globalmente contribuyen a obtener un producto intermedio, necesario para continuar hacia el producto final y facilitar la gestión del proyecto. Al conjunto de las fases empleadas se le denomina **“ciclo de vida”**.

La forma de agrupar las actividades, los objetivos de cada fase, los tipos de productos intermedios que se generan, etc. pueden ser muy diferentes dependiendo del tipo de producto o proceso a generar y de las tecnologías empleadas.

La complejidad de las relaciones entre las distintas actividades crece exponencialmente con el tamaño, por lo que se haría inabordable rápidamente si no fuese por la vieja táctica de “divide y vencerás”. Mediante la misma, la división de los proyectos en fases sucesivas permite la reducción de su complejidad, tratándose de seleccionar las partes de manera que sus relaciones entre sí sean lo más simples posibles.

La definición de un ciclo de vida facilita el control del tiempo en que es necesario aplicar recursos de diferente tipo (personal, equipos, suministros, etc.) al proyecto. Si el proyecto incluye subcontratación de partes a otras organizaciones, el control del trabajo subcontratado se facilita en la medida en que esas partes encajen bien en la estructura de las fases. El control de calidad también se ve facilitado si la separación entre fases se hace corresponder con puntos en los que ésta deba verificarse (mediante comprobaciones sobre los productos parciales obtenidos).

De la misma forma, la práctica acumulada en el diseño de modelos de ciclo de vida para situaciones muy diversas permite que nos beneficiemos de la experiencia adquirida utilizando el enfoque que mejor se adapte a nuestros requerimientos.

## 3 Mitos del software

Muchas de las causas de la crisis del software se pueden encontrar en los mitos que surgen durante los primeros años del desarrollo del software. Dichos mitos poseen varios atributos: aparecieron como declaraciones razonables de hechos (algunas veces conteniendo elementos verdaderos), tuvieron un sentido intuitivo y frecuentemente fueron promulgados por expertos.

### 3.1 Mitos del Cliente

Muchas veces sucede que el cliente que solicita una aplicación de software posee creencias y percepciones erróneas sobre el software y su proceso de fabricación. Dichas creencias hacen que el cliente se cree una falsa expectativa y, finalmente, quede insatisfecho con la persona o empresa que desarrolla el software.

#### Mito:

1. Una declaración general de los objetivos es suficiente para comenzar a escribir los programas. Podemos dar los detalles más adelante.
2. Los requisitos del proyecto cambian continuamente, pero los cambios pueden acomodarse fácilmente ya que el software es flexible.

#### Realidad:

1. Una mala definición inicial es la principal causa del trabajo inútil en software. Es esencial una descripción formal y detallada del ámbito de la información. Es necesaria una exhaustiva comunicación entre el cliente y el analista.

2. Es verdad que los requisitos del software cambian, pero el impacto del cambio varía según en el momento que se presenten.

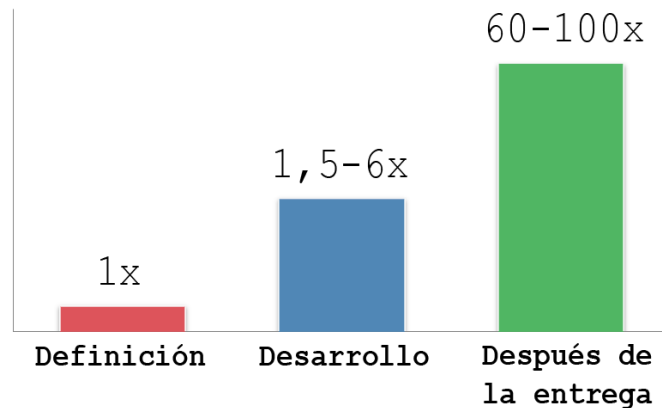


Figura 4 – Impacto de modificación en el Software

- **Definición:** Impacto Bajo
- **Desarrollo:** Impacto Alto
- **Después de la entrega:** Impacto Crítico

### 3.2 Mitos de los desarrolladores

#### Mito:

- 1) Una vez que escribimos el programa y hacemos que funcione, nuestro trabajo ha terminado.
- 2) Hasta que no tengo el programa «ejecutándose», realmente no tengo forma de comprobar su calidad.
- 3) Lo único que se entrega al terminar el proyecto es el programa funcionando.

#### Realidad:

- 1) Alguien dijo una vez: «cuanto más pronto se comience a escribir código, más se tardará en terminarlo».
- 2) Cuando más pronto se comience a escribir código, más se tardará en terminarlo. Entre el 50% y el 70% de todo el esfuerzo dedicado a un programa se realizará después de que se la haya entregado al cliente por primera vez.
- 3) Un programa que funciona es sólo una parte de una configuración del software que incluye programas, documentos y datos. La documentación es la base de un desarrollo y, lo que es más importante, proporciona guías para la tarea de mantenimiento del software.

### 3.3 Mitos de gestión

Los gestores con responsabilidad sobre el software, como en la mayoría de las disciplinas, están normalmente bajo la presión de cumplir los presupuestos, hacer que no se retrase el proyecto y mejorar la calidad.

#### Mito:

- 1) Tenemos ya un libro que está lleno de estándares y procedimientos para construir software, ¿no le proporciona ya a mi gente todo lo que necesita saber?
- 2) Mi gente dispone de las herramientas de desarrollo de software más avanzadas, después de todo, les compramos las computadoras más modernas.
- 3) Si fallamos en la planificación, podemos añadir más programadores y adelantar el tiempo perdido (el llamado algunas veces «concepto de la horda Mongoliana»).

#### Realidad:



- 1) El libro existe pero: ¿se utiliza? ¿Conocen los funcionarios su existencia? ¿Refleja las prácticas modernas de desarrollo de software? ¿Es completo?
- 2) Se necesita mucho más que el último modelo de computadora para hacer un software de calidad. Las herramientas CASE son importantes, más que el hardware.
- 3) El desarrollo de software no es un proceso mecánico en el cual se agregan personas a un proyecto retrasado.

## 4 Modelo de proceso de la IEEE 1074

A continuación se describe en detalle las fases o subprocesos que conforman el proceso base de construcción de software correspondiente al estándar IEEE 1074-1989. Cada subproceso detalla el nivel de propósito, actividades involucradas y documentación principal propuesta por el estándar.

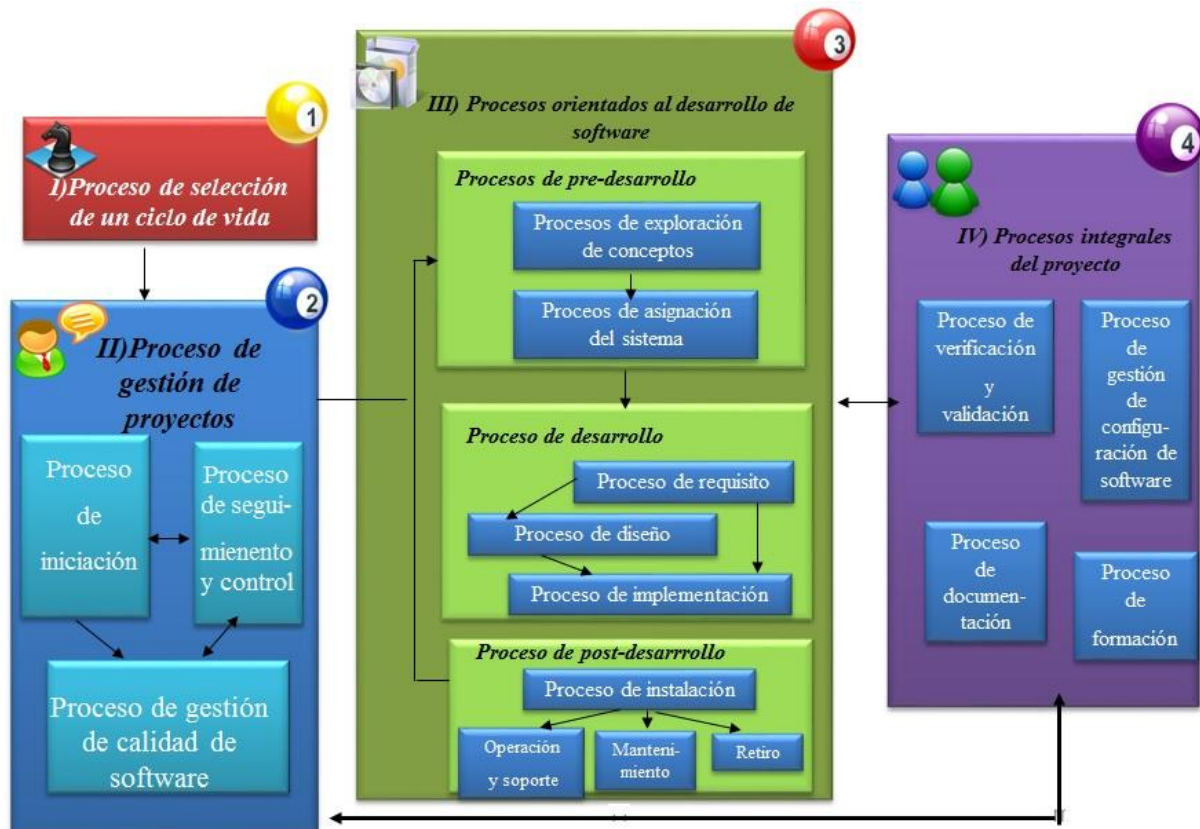


Figura 5 - Estándar IEEE 1074-1989

El estándar está dirigido a responsables de la gestión y desarrollo de software. Se compone de cuatro procesos principales, cada uno de los cuales agrupa una serie de actividades:

- 1) **Proceso de selección de un modelo de ciclo de vida** del producto que identifica y selecciona un ciclo de vida orientado al software que se va a construir.
- 2) **Proceso de gestión del proyecto**, donde se crean la estructura del proyecto y aseguran el nivel apropiado de la gestión del mismo durante todo el ciclo de vida del software.
- 3) **Procesos orientados al desarrollo del software**, que producen, instalan, operan y mantienen el software y lo retiran de su uso. Se clasifican en procesos de pre-desarrollo, desarrollo y post-desarrollo.
- 4) **Procesos integrales del proyecto**, son los necesarios para completar con éxito las actividades del proyecto software. Aseguran la terminación y calidad de las funciones del mismo. Los mismos se ejecutan en simultáneos a los procesos orientados al desarrollo de software e incluyen actividades que no son de desarrollo.

## 4.1 Selección de ciclo de vida

### 4.1.1 ¿Qué es el ciclo de vida?

“Un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso” **ISO 12207-1**

Cuando el proceso implica la construcción de algún producto, solemos referirnos al proceso como **Ciclo de Vida**. El ciclo de vida describe el desarrollo de software, desde su fase inicial hasta su fase final, en el cual se establecen fases intermedias que permitan garantizar que el software en su totalidad cumpla con los objetivos planteados y además permita verificar que se estén utilizando los métodos más apropiados para su desarrollo.

Dentro del ciclo de vida, se definen las actividades necesarias para el desarrollo del software, así como también se introducen puntos de control para realizar revisiones y controles de calidad. Esto le da la posibilidad a los desarrolladores de concentrarse en la calidad del software permitiendo detectar errores lo antes posible dentro del desarrollo del proyecto. Asimismo, se establece un marco de trabajo que permite realizar estimaciones sobre los recursos, los costos y la planificación. Dichas estimaciones se realizan al comienzo del proyecto, dentro de un tiempo limitado, las cuales se recomienda actualizar a medida que se progresa.

Todo este conjunto de actividades establecidas en el ciclo de vida no sólo permite a los desarrolladores desarrollar e implantar un proyecto de software, sino también mantener una coherencia entre todos los proyectos de una organización.

### 4.1.2 Selección del ciclo de vida

- **Responsable:** Gerente del proyecto.
- Personalizar las actividades de la IEEE 1074 a los requerimientos del proyecto y de la empresa.
- Seleccionar el ciclo de vida:
  - En cascada
  - Prototipo
  - RDA
  - Incremental
  - Basado en componentes
  - En espiral

### 4.1.3 Diferencias entre: Proceso y Ciclo de vida

Proceso	Ciclo de Vida
Serie de pasos con actividades	Cuando el proceso involucra la construcción de un producto, a veces se menciona como Ciclo de Vida (del producto).

## 4.2 Proceso de gestión de proyecto

Este proceso se compone de tres subprocesos: Iniciar el proyecto, Supervisar y controlar el proyecto y gestión de la calidad del proyecto.

### 4.2.1 Iniciar el proyecto

#### 4.2.1.1 Iniciación

Abarca aquellas actividades de creación de la estructura del proyecto. Es aquí donde:

- Se define el ciclo de vida del software para este proyecto y se establecen los planes para su gestión.
- Se determinan los costos y recursos necesarios a fin de ejecutar las distintas tareas que demanda el proyecto.
- Se identifican y seleccionan estándares, metodologías y herramientas para la gestión, herramientas para la ejecución del mismo
- Se prepara y establece un plan para su implementación adecuada y oportuna. El plan de gestión de proyectos software que conducirá al desarrollo se produce como culminación de este proceso.

Se pueden definir claramente tres etapas:

- **Estimación**
  - En las mismas se debe predecir el esfuerzo del personal, los costos para terminar las actividades y los productos que obtendremos en cada una de las etapas.
- **Planificación**
  - Predicción de la duración de las tareas y actividades, los recursos requeridos y la relación, orden y solapamiento de los mismos.
- **Selección de estándares, metodologías, herramientas etc.**
  - Elaboración de planes de gestión.

#### 4.2.1.2 Componentes del proceso: Iniciación del proyecto

- **Responsable:**
  - Gerente del proyecto
- **Entregable:**
  - Plan de gestión del proyecto de software
- **Propósitos:**
  - Establecer la correspondencia entre las actividades y el modelo del ciclo de vida del software
  - Asignar los recursos al proyecto
  - Establecer el ambiente del proceso
  - Realizar la planificación de la gestión del proyecto
- **Técnicas a aplicar**
  - Análisis de camino crítico
  - Análisis PERT
  - Diagrama de GANTT
  - Técnicas estadísticas
  - Técnicas de simulación (método de Montecarlo)
  - Puntos de función
  - Modelos empíricos de estimación
  - Técnicas de descomposición funcional

### 4.2.2 Proceso de gestión de proyecto

El propósito del Seguimiento y Control de Proyectos es el de proveer una visión objetiva del estado actual del proyecto y determinar las posibles desviaciones. Le llamamos *Seguimiento* a la evaluación rutinaria del estado y *Control* a la toma de medidas correctivas. Esta práctica guarda íntima relación con la Planificación de Proyectos ya que es donde se encuentra correctamente documentado el proyecto, que es el insumo para verificar su cumplimiento.

Una vez detectadas las desviaciones, es necesario que el equipo determine oportunamente la corrección requerida y la lleve a cabo durante la siguiente iteración o en el momento en que se considere oportuno.

Finalmente, es necesario que la corrección planteada sea a su vez objeto de seguimiento. Esto implica que la planificación debe ser actualizada para que refleje las acciones que se han determinado necesarias para corregir la desviación.

A la hora de realizar las reuniones de seguimiento del proyecto es útil contar con momentos para discutir y revisar el Plan de Riesgos, a fin de verificar la ocurrencia de alguno de estos eventos negativos. Además, si bien es una práctica avanzada, debe considerarse la posibilidad de calcular alguna forma de medida o métrica que aplicada al proyecto de desarrollo sirva de indicador sobre su estado, a fin de obtener una evaluación lo más completa y objetiva posible del estado del proyecto.

Otros factores que deben ser objeto de seguimiento incluyen el presupuesto en tiempo y dinero del proyecto y el cumplimiento de los hitos señalados como objetivos del ciclo de vida.

Se pueden definir claramente tres etapas:

**1) Gestión de proyecto**

Implica la recolección y registro de los datos sobre los recursos, costos e hitos asociados al proyecto

**2) Análisis de riesgo**

Implica identificar, analizar, estimar (proyectar) y gestionar los riesgos que pueden ser: técnicos, de proyecto, de negocio etc.). Involucra estrategias para minimizar, supervisar y gestionar riesgos proporcionando planes de contingencia.

**3) Gestión y coordinación del proyecto.**

En este proceso se debe supervisar el progreso del proyecto, comparando el resultado de lo que se planificó y lo que realmente se obtuvo. En el mismo se debe coordinar recursos, personal, actividades y realizar revisiones.

#### **4.2.2.1 Componentes del proceso de Seguimiento y control**

- **Responsable:**
  - Gerente del proyecto
- **Entregable:**
  - Plan de gestión del proyecto de software, actualizado
- **Propósito:**
  - Analizar riesgos
  - Realizar planeación de contingencias
  - Gestionar el proyecto
  - Conservar los registros históricos
  - Implementar el modelo de reporte de problemas
- **Técnicas a utilizar:**
  - Análisis de riesgo técnico
  - Análisis de riesgo económico
  - Análisis de riesgo operativo y de soporte
  - Análisis de riesgo de programa (Camino crítico)

### **4.2.3 Gestión de la calidad del software**

#### **4.2.3.1 Definición de calidad**

De acuerdo a la definición del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, Std. 610-1990) *“la calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario.”* Su objetivo es la planificación y administración de las acciones necesarias para proveer una confianza adecuada de la calidad de los productos software. Se abarcan actividades en todo el ciclo de vida y se documenta en un plan de garantía de la calidad del software.

Para abordar este proceso de protección del software globalmente se consideran los siguientes aspectos:

● **Métricas para el control del proyecto**

- Las métricas del software se definen como la aplicación continua de técnicas basadas en las medidas de los procesos de desarrollo de software y de sus productos para producir una información de gestión significativa y oportuna, a la vez que se mejoran los procesos y sus productos.

- **Verificación y validación**
  - o La verificación y la validación del software involucran actividades imprescindibles para el control de la calidad del software.
  - o Se entiende por verificación al conjunto de actividades para la comprobación de que un producto software está técnicamente bien construido, es decir, que el producto funciona.
  - o La validación trata la comprobación de que el producto software construido es el que se deseaba construir, es decir que funciona como el usuario quiere y hace las funciones que fueron concertadas con él.
- **Gestión de la configuración del producto**

Factores que determinan la Calidad el Software:

- Corrección. ¿Hace lo que quiero?
- Fiabilidad. ¿Lo hace de forma fiable todo el tiempo?
- Eficiencia. ¿Se ejecutará en mi hardware lo mejor que pueda?
- Seguridad (Integridad). ¿Es seguro?
- Facilidad de uso. ¿Está diseñado para ser usado?

#### 4.2.3.2 *Gestión de calidad de software*

Consiste en determinar las acciones necesarias para proveer una confianza en la calidad de los productos software, asegurar la satisfacción de requisitos técnicos y de usuarios.

Las acciones son:

- Planificar la gestión de Calidad de Software. Identificar acciones de SQA, requisitos de calidad, satisfacción del cliente.
- Gestionar la calidad del software.
- Definición y control de métricas. Técnicas para medir el proceso y el producto software
- Identificar necesidades de mejorar la calidad

#### 4.2.3.3 *Componentes del proceso: Gestión de calidad de software*

- **Responsable:**
  - o Jefe de Gestión de la calidad
- **Entregable:**
  - o Plan de gestión de la calidad
- **Propósito:**
  - o Planear administración de la calidad
  - o Definir medidas.
  - o Gestionar la calidad.
  - o Identificar necesidades de mejora de calidad.
- **Técnicas:**
  - o Técnicas de planificación y estimación
  - o Métricas de calidad de software

## 4.3 Procesos orientados al desarrollo de software

### 4.3.1 Procesos de pre-desarrollo

Son los procesos que se deben realizar antes de que comience el desarrollo del software propiamente dicho.

El desarrollo se inicia con la identificación de una necesidad, la cual para ser satisfecha necesita de una nueva aplicación, o cambio de todo o parte de la aplicación existente.

#### 4.3.1.1 *Proceso de exploración de conceptos*

Este proceso incluye la identificación de una necesidad, la formulación de soluciones potenciales, el estudio de viabilidad, y refinamiento a nivel de sistema. Una vez establecidos sus límites, se genera el

informe de la necesidad del sistema a desarrolla, dando inicio al proceso de asignación del sistema y/o el proceso de requisitos y alimentando los procesos de gestión del proyecto.

El informe de la necesidad es un documento que constituye la base de todo el trabajo de ingeniería posterior.

#### *4.3.1.2 Componentes del proceso: Exploración de conceptos*

- **Responsable:**
  - Gestor y Cliente
- **Entregable:**
  - Enunciado del problema
- **Propósito:**
  - Producir el enunciado del problema (Statement of needs)
  - Identificar ideas o necesidades
  - Formular enfoques potenciales
  - Realizar estudios de factibilidad
  - Planear la transición del sistema
  - Refinar y finalizar la idea o necesidad
- **Técnicas:**
  - Técnicas de adquisición de conocimiento
  - Análisis económico, técnico
  - Análisis alternativo
  - Técnicas de modelización (diagramas)
  - Prototipos

#### *4.3.1.3 Procesos de asignación del sistema*

Este proceso se realiza cuando el sistema requiere tanto del desarrollo de hardware como el de software, o cuando no se está seguro que solo se necesita desarrollo de software. En el informe de necesidad se identifican:

- Las entradas y el procesamiento que se les va a aplicar.
- Las salidas requeridas.
- Las funciones del sistema total, que permiten desarrollar la arquitectura del sistema e identificar las funciones del hardware, del software y de las interfaces.

Este proceso culmina con la especificación de requisitos del software, la especificación de requisitos del hardware y la especificación de la interfaz del sistema.

Se comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando un subconjunto de estos requisitos del software, para su análisis y refinamiento en el proceso de requisitos. Este planteamiento del sistema es esencial cuando el software debe interrelacionarse con otros elementos, tales como hardware, personas y bases de datos.

El análisis de sistema requiere una comunicación intensa entre el cliente y el analista. El cliente debe comprender los objetivos del sistema y ser capaz de exponerlos claramente. El analista debe ser qué preguntas hacer, que consejos dar y qué investigación realizar, pues si la comunicación se rompe, el éxito del proyecto entero estará en peligro. En el análisis del sistema se definen:

- Sus objetivos.
- La información que se va a obtener.
- La información que se va a suministrar.
- Las funciones, el comportamiento y el rendimiento requerido.

Una vez que la función, el rendimiento y las interfaces están delimitados, se procede a realizar la tarea denominada **asignación**, donde las funciones son asociadas a uno o más elementos genéricos del sistema,

es decir, software, hardware, personal, bases de datos, documentación, procedimientos. Esencialmente, lo que se hace es asignar a cada elemento del sistema un ámbito de funcionamiento y de rendimiento.

Asignadas las funciones, se puede crear un modelo que represente las interrelaciones entre los distintos elementos del sistema y establezca una base para los posteriores pasos de análisis de requisitos y de diseño. Se representa el sistema definido mediante modelos de la arquitectura del sistema (salida, entrada, proceso y control, interfaz de usuario, mantenimiento y autocomprobación).

En primer lugar, se realiza un diagrama de contexto de la arquitectura, que establece los límites de información entre los que se está implementando el sistema y el entorno en el que va a funcionar. Luego, se refina el diagrama de contexto de la arquitectura considerando con más detalle la función del sistema. Se identifican los subsistemas principales que permiten el funcionamiento del sistema considerado en el contexto definido por el diagrama. Se definen los subsistemas principales en un diagrama de flujo de arquitectura. El diagrama de flujo de arquitectura muestra los subsistemas principales y las líneas importantes de flujo de información (control y datos).

El diagrama inicial de flujo de la arquitectura se constituye en el nodo raíz de la jerarquía de diagramas de flujo. Se puede ampliar cada subsistema del diagrama de flujo inicial en otro diagrama de arquitectura dedicado exclusivamente a él. Este proceso de descomposición de arriba hacia abajo permite disponer de una jerarquía de diagramas de flujo del sistema, donde cada uno de ellos se puede utilizar como punto de partida para los posteriores pasos de ingeniería del subsistema que describe.

#### **4.3.1.4 Componentes del proceso: Asignación del sistema**

- **Responsable:**
  - Gestor, Cliente y Arquitecto de software
- **Entregable:**
  - Enunciado del problema y documento de arquitectura
- **Propósito:**
  - Establecer la arquitectura inicial
  - Base de comunicación entre los integrantes del proyecto
  - Identificar el hardware, software, requerimientos funcionales
- **Técnicas:**
  - Técnicas de adquisición de conocimiento
  - Técnicas de modelización del sistema (diagramas)

### **4.3.2 Procesos de desarrollo**

Son los procesos que se deben realizar para la construcción del producto software. Definirán qué información obtener y como estructurar los datos, qué algoritmos usar para procesar los datos y cómo implementarlos, y qué interfaces desarrollar para operar con el software y cómo hacerlo.

A partir del informe de la necesidad, con el soporte de las actividades de los procesos integrales y bajo el plan de gestión del proyecto, los procesos de desarrollo producen el software (código y documentación).

#### **4.3.2.1 Procesos de requisitos**

Incluye actividades iterativas dirigidas al desarrollo de la especificación de requisitos del software. Para la determinación completa y consistente de todos los requerimientos del software el análisis se enfatiza sobre la salida resultante, la descomposición de los datos, el procesamiento de los datos, las bases de datos y las interfaces de usuario, del software y del hardware.

La especificación de requerimientos del software es el establecimiento conciso y preciso de un conjunto de requisitos que deben ser satisfechos por un producto software, indicando el procedimiento mediante el cual se puede determinar si se satisfacen los requerimientos dados. Describe los requisitos funcionales, de rendimiento y de interfaz de software y definen los entornos de operación y soporte. Este documento es la salida con la cual culmina este proceso.

Existen tres tipos de requerimientos:



- **Requerimiento funcional:** especifica la función que un sistema o componente de un sistema debe ser capaz de realizar.
- **Requerimiento de rendimiento:** especifica una característica numérica tanto estática (número de terminales del sistema, número de usuarios simultáneos que soportará el sistema, número de archivos o registros del mismo, etc.) como dinámica (tiempo de procesamiento en el sistema) que debe tener un sistema o componente de un sistema.
- **Requisitos de interfaz:** determinan características que el software debe soportar para cada interfaz humano del producto software (interfaz de usuario), las características lógicas de cada interfaz entre el producto software y los componentes de hardware del sistema (interfaz del hardware), el uso de otros productos software (un sistema de gestión de base de datos, un sistema operativo, un paquete estadístico) e interfaces con otros sistemas de aplicación (interfaz de software).

Existen además otros atributos del software (seguridad, consistencia, facilidad de traza, etc.) que pueden dar lugar a requisitos específicos del mismo, por ejemplo la restricción a determinados datos.

#### 4.3.2.2 Componentes del proceso: Requisitos

- **Responsable:**
  - Equipo de desarrollo y cliente
- **Entregable:**
  - Documento de requerimiento
- **Propósito:**
  - Definir y desarrollar los requerimientos de software
  - Definir los requerimientos de interfaz
  - Establecer la prioridad e integrar los requerimientos de software
- **Técnicas:**
  - Técnicas orientadas a los procesos, a los datos, a los objetos
  - Técnicas formales de especificación.

#### 4.3.2.3 Proceso de diseño

Su objetivo es realizar una representación coherente y organizada del sistema software que satisfaga la especificación de requisitos del software. Traduce el “qué hacer” de las especificaciones de los requerimientos en el “cómo hacerlo” de las especificaciones de diseño.

Inicialmente, la representación describe una visión sistémica y holística del software. Posteriores refinamientos de diseño conducen a una representación que se acerca al código fuente.

El diseño de software puede verse desde dos perspectivas:

- **Perspectiva técnica**
  - El diseño comprende cuatro actividades: diseño de los datos, diseño arquitectónico, diseño procedimental y diseño de interfaces.
- **Perspectiva de gestión del proyecto**
  - El diseño va del diseño arquitectónico (diseño preliminar o de alto nivel) al diseño detallado (diseño de bajo nivel).
  - El nivel de diseño arquitectónico se focaliza en las funciones y estructuras de las componentes que conforman el sistema software. El nivel de diseño detallado se ocupa del refinamiento de la representación arquitectónica que lleva a una estructura de datos detallada y a las representaciones algorítmicas que se usan para cada componente modular del software.

El proceso de diseño de software comienza con la actividad de realizar el diseño arquitectónico. Esta actividad genera la descripción del diseño arquitectónico del software en donde:

- Se describe cada una de las componentes software.



- Se especifican los datos, las relaciones y restricciones.
- Se definen todas las interfaces externas (usuario, software y hardware) y los internos (entre componentes).

La última actividad del proceso de diseño es realizar el diseño detallado, donde se genera la descripción del diseño del software, que especifica la estructura de los datos, los algoritmos y la información de control de cada componente software, y los detalles de las interfaces (usuario, hardware y software).

El diseño detallado se deriva del diseño preliminar; en consecuencia, sus correspondientes actividades se realizan en consecuencia mientras que el resto de las actividades de este proceso (analizar el flujo de información, diseñar la base de datos, diseñar las interfaces, desarrollar los algoritmos) se ejecutan en paralelo.

Una actividad relevante es el diseño de la base de datos que comprende el diseño conceptual, lógico y físico, de la base de datos. Los requisitos se modernizan dentro de un esquema externo que describe las entidades de datos, atributos, relaciones y restricciones. Los distintos esquemas externos se integran en un esquema conceptual único. El esquema conceptual se aplica entonces en un esquema lógico dependiente de la implementación. Finalmente, se definen las estructuras físicas de datos y los caminos de acceso.

El diseño es el proceso en el que se asienta la calidad del desarrollo de software ya que produce las representaciones del software de las que puede evaluarse su calidad. Es la única forma de traducir con precisión los requerimientos del cliente en un producto o sistema acabado y sirve como base de todas las posteriores etapas del desarrollo y del proceso de mantenimiento.

Como ya se mencionó, desde el punto de vista técnico y en el contexto de los diseños preliminar y detallado, se llevan a cabo varias actividades de diseño diferentes:

- **Diseño de datos**
  - o Transforma el modelo del campo de información, creado durante el análisis, en las estructuras de datos que se van a requerir para implementar el software.
- **Diseño arquitectónico**
  - o Define las relaciones entre los principales elementos estructurales del programa. El objetivo principal de este diseño es desarrollar una estructura de programa modular y representar las relaciones de control entre los módulos. Mezcla la estructura de programas y la estructura de datos y define las interfaces que facilitan el flujo de los datos a lo largo del programa.
- **Diseño procedimental**
  - o El diseño procedimental, que transforma los elementos estructurales en una descripción procedimental del software; se realiza después de que se ha establecido la estructura del programa y de los datos.
- **Diseño de la interfaz**
  - o El diseño de la interfaz, que establece principalmente la disposición y los mecanismos para la interacción hombre - máquina.

Para evaluar la calidad de una representación del diseño se deben tener en cuenta las siguientes consideraciones:

- 1) **Jerarquización:** se debe exhibir de una organización jerárquica que haga un uso inteligente del control entre los componentes del software.
- 2) **Modularidad:** el software debe estar dividido de forma lógica en elementos que realicen funciones y sub-funciones específicas.
- 3) **Separabilidad:** se debe contener representaciones distintas y separadas de los datos y de los procedimientos.
- 4) **Independencia funcional:** se debe conseguir módulos que exhiban características funcionales independientes.
- 5) **Conectividad:** producir interfaces que reduzcan la complejidad de las conexiones entre los módulos y el entorno exterior.

- 6) **Reproductibilidad:** debe obtenerse mediante un método que sea reproducible y que esté conducido por la información obtenida durante el análisis de los requerimientos del software.

#### 4.3.2.4 Componentes del proceso: Diseño

- **Responsable:**
  - Arquitecto
- **Entregable:**
  - Documento con el diseño del software, descripción de la arquitectura, descripción del flujo de información.
  - Estructura de la base de datos
  - Descripción de interfaces y de algoritmos.
- **Propósito:**
  - Su objetivo es realizar una representación coherente y organizada del sistema software que satisfaga la especificación de requisitos del software.
- **Técnicas:**
  - Técnicas orientadas a los procesos, a los datos, a los objetos
  - Técnicas formales de especificación.

#### 4.3.2.5 Proceso de implementación

Este proceso transforma la representación del diseño detallado de un producto software a una realización en un lenguaje de programación apropiado. Produce el código fuente, el código de la base de datos y la documentación, que constituyen la manifestación física del diseño de acuerdo a los estándares y metodologías del proyecto. Además, en este proceso se debe integrar el código y la base de datos. En el caso de que el sistema conste de componentes hardware y software, se debe planificar y realizar la integración del sistema.

La salida de este proceso está sujeta a las pruebas de verificación y validación adecuadas. El código y la base de datos junto con la documentación producida durante los procesos previos son la primera representación completa del producto software.

#### 4.3.2.6 Componentes del proceso: Implementación

- **Responsable:**
  - Jefe de equipo de desarrollo
- **Entregable:**
  - Código, documentación y resultados de test
- **Propósito:**
  - Construir una solución
  - Crear datos de pruebas
  - Crear código fuente
  - Crear la documentación operativa
  - Planear la integración
  - Realizar la integración
- **Técnicas:**
  - Técnicas orientadas a los procesos, a los datos, a los objetos
  - Técnicas de diseño a bajo nivel. Programación
  - Técnicas de prototipado

### 4.3.3 Proceso de post-desarrollo

#### 4.3.3.1 Proceso de operación y soporte

Son los procesos que se deben realizar para instalar, operar, soportar, mantener y retirar un producto software. Una vez terminada la prueba del software, éste está casi preparado para ser entregado a los usuarios finales, pero antes deben llevarse a cabo una serie de actividades de garantía de calidad para asegurar que se hayan generado y catalogado los registros, y documentos internos adecuados, que se ha

desarrollado una documentación de alta calidad para el usuario, y que se han establecido los mecanismos apropiados de control de configuraciones.

Tan pronto como se entregue el software a los usuarios finales, el trabajo del ingeniero del software cambia, en este momento el enfoque pasa de la construcción al mantenimiento; corrección de errores, adaptación al entorno y mejora de la funcionalidad.

En todos los casos, la modificación del software no solo afecta al código, sino también a la configuración entera, es decir, a todos los documentos, datos y programas desarrollados en la fase de planificación y desarrollo.

#### **4.3.3.2 Proceso de instalación**

Implica la instalación de un sistema software desde el entorno de desarrollo al entorno de destino. Incluye la carga de la base de datos, las modificaciones necesarias del software, las comprobaciones en el entorno de destino y la aceptación del cliente. Si durante la instalación surge un problema se identifica e informa acerca de él.

El proceso de instalación verifica que se implemente la configuración adecuada del software y termina con la aceptación formal del mismo por parte del cliente conforme a lo especificado en el plan de gestión del proceso software y la realización con éxito de la prueba de aceptación del usuario.

#### **4.3.3.3 Componentes del proceso: Instalación**

- **Responsable:**
  - Gerente y cliente
- **Entregable:**
  - Sistema
- **Propósito:**
  - Planear la instalación
  - Instalar el software
  - Aceptar el software en el ambiente operacional

#### **4.3.3.4 Proceso de operación y soporte**

Involucra la operación del sistema por parte del usuario y el soporte continuo al usuario que incluye asistencia técnica, consultas con el usuario y registro de las peticiones de soporte en el histórico de peticiones de soporte. Así, este proceso puede desencadenar la actividad del proceso de mantenimiento que provee información de realimentación al ciclo de vida del software.

#### **4.3.3.5 Componentes del proceso: Operación y soporte**

- **Responsable:**
  - Gerente y cliente
- **Entregable:**
  - Sistema
- **Propósito:**
  - Operar el sistema
  - Proporcionar asistencia técnica y consultoría
  - Mantener la bitácora de pedidos de cliente

#### **4.3.3.6 Proceso de mantenimiento**

Se interesa por los errores, defectos, fallas, mejoras y cambios del software. Un requisito de mantenimiento del software inicia los cambios del ciclo de vida del software; éste se reasigna y se ejecuta.

El mantenimiento se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas por la evolución del entorno del software y a las modificaciones debidas a los cambios de los requisitos del cliente dirigidos a reforzar o ampliar el sistema. El proceso de mantenimiento vuelve a aplicar los pasos del ciclo de vida, pero en el contexto del software ya existente.

Durante el mantenimiento se encuentran cuatro tipos de cambios:

**1) Correctivo:**

Incluso llevando a cabo las mejores actividades de garantía de calidad, es muy probable que el cliente descubra defectos en el software. El mantenimiento correctivo cambia el software para corregir los defectos.

**2) Preventivo:**

Se realizan cambios para la prevención de posibles problemas que puedan llegar a surgir con el sistema.

**3) Adaptativo:**

Con el paso del tiempo es probable que cambie el entorno original para el que se desarrolló el software. El mantenimiento adaptativo consiste en modificar el software para acomodarlo a los cambios de su entorno externo.

**4) Perfectivo:**

Es el proceso por el cual se mejora y optimiza el software que se ha instalado.

La salida de este proceso son las recomendaciones del mantenimiento que entran al ciclo de vida del software en el proceso de exploración de conceptos para mejorar la calidad del sistema software.

#### **4.3.3.7 Componentes del proceso: Mantenimiento**

- **Responsable:**

- Gerente y equipo de desarrollo

- **Entregable:**

- Sistema actualizado

- **Propósito:**

- Mantener el sistema adecuado a los cambios en los requerimientos, nuevas necesidades o aplicación de nuevas tecnologías.

#### **4.3.3.8 Proceso de Retiro**

Se puede decir que es “la jubilación” de un sistema existente, de su soporte activo o de su uso. Se realiza mediante el cese de su operación o soporte o su reemplazo por un nuevo sistema o por su actualización.

Si el sistema en uso se reemplaza por uno nuevo, se requiere un tiempo de operación en paralelo. En este periodo se utiliza el sistema en retiro para los resultados oficiales, mientras se prepara el nuevo sistema para su operación formal. Es un período de formación para el usuario sobre el nuevo sistema y de validación del mismo.

#### **4.3.3.9 Componentes del proceso: Retiro**

- **Responsable:**

- Gerente y cliente

- **Entregable:**

- Planificación del retiro del sistema

- **Propósito:**

- Notificar a los usuarios
  - Realizar operaciones paralelas (en caso de sustituir el sistema por otro)
  - Retirar el sistema

## **4.4 Procesos integrales del proyecto**

Son procesos simultáneos y complementarios a los procesos orientados hacia el desarrollo. Incluyen actividades imprescindibles para que el sistema construido sea fiable (procesos de verificación y validación, gestión de la configuración) y sea utilizado al máximo de sus capacidades (procesos de formación, documentación). Los procesos integrales comprenden dos tipos de actividades: Aquellas que se realizan discretamente y se aplican dentro de un ciclo de vida del software y las que se realizan para completar otra actividad, estas solo se invocan y no se aplican dentro del ciclo de vida para cada instancia.

#### 4.4.1 Proceso de verificación y validación

Abarca la planificación y la realización de todas las tareas de verificación, incluyendo pruebas de verificación, revisiones y auditorías, y de todas las tareas de validación, que se ejecutan durante el ciclo de vida del software para asegurar que se satisfacen todos los requisitos del software.

Una actividad útil para la verificación y la validación del software es la prueba del software. Constituye el proceso de ejecución del software con determinados datos de entrada para observar los resultados que produce y compararlos con los que se ha establecido que debería producir, con el fin de detectar posibles fallas. Estas pruebas sólo podrán realizarse cuando en el proceso de desarrollo ya exista código ejecutable.

La depuración es un proceso frecuentemente asociado a las pruebas que consiste en tratar de deducir dónde están los defectos en el software que provocan que éste no funcione adecuadamente. Estudia los resultados de las pruebas y otras actividades de control para intentar buscar qué está mal en el software.

##### 4.4.1.1 Componentes del proceso: Verificación y validación

- **Responsable:**
  - Gerente y equipo de desarrollo
- **Entregable:**
  - Plan de verificación y validación
  - Informes de evaluación
  - Planes de pruebas
  - Especificación de las pruebas
  - Informe de resumen de pruebas
  - Software aprobado
- **Propósito:**
  - Planear la verificación y validación: inspecciones, revisiones, auditorías, pruebas
  - Ejecutar las tareas de verificación y validación
  - Recopilar y analizar datos de medidas
  - Planear pruebas
  - Desarrollar los requerimientos de las pruebas
  - Ejecutar las pruebas

#### 4.4.2 Proceso de gestión de la configuración

Este proceso involucra un conjunto de actividades desarrolladas para gestionar los cambios durante todo el ciclo de vida del software. Su objetivo es el control de los cambios en el sistema, mantener su coherencia y su rastreabilidad o trazabilidad, y poder realizar auditorías de control sobre la evolución de las configuraciones. Identifica la estructura de un sistema en un momento dado, es decir, qué rutinas, módulos, datos, archivos lo componen. A esta estructura se le denomina configuración del sistema.

La gestión de la configuración realiza las siguientes funciones:

- Identificación de la configuración de un sistema o descripción documentada de sus características reales en un determinado momento.
- Control de configuración.
- Establecimiento y control de cambios de la configuración inicial o básica.
- Informes del estado de la configuración
- Auditorías de la configuración
- Revisiones independientes de la configuración para comprobar que los elementos de la configuración cumplen los requisitos de configuración establecidos.

##### 4.4.2.1 Componentes del proceso: Gestión de configuración

- **Responsable:**
  - Jefe de proyecto
- **Entregable:**
  - Planificación de la gestión de configuración (orden de cambio, cambio de estado, informe de estado)

- Realizar la identificación de la configuración
- Realizar el control de configuración
- Realizar la información del estado de la configuración
- **Propósito:**
  - Planear la gestión de la configuración
  - Desarrollar la identificación de la configuración
  - Realizar el control de cambios
  - Asegurar que el contenido del producto es conocido y se encuentra disponible siempre
  - Ayudar a la coordinación entre el equipo
  - Tener un repositorio (depósito) único para los entregables.
  - Tener una base histórica con los cambios del producto durante el tiempo.

#### 4.4.3 Proceso de desarrollo de documentación

El proceso de desarrollo de documentación para el desarrollo y uso del software es el conjunto de actividades que planifican, diseñan, implementan, editan, producen, distribuyen y mantienen los documentos necesarios para los desarrolladores y los usuarios.

##### 4.4.3.1 Componentes del proceso: Documentación

- **Responsable:**
  - Jefe de proyecto
- **Entregable:**
  - Planificar la documentación
- **Propósito:**
  - Planear la documentación
  - Implementar la documentación
  - Producir y distribuir la documentación

#### 4.4.4 Proceso de formación

Incluye la planificación, desarrollo, validación e implementación de los programas de formación de desarrolladores, personal de soporte técnico y clientes o usuarios y la elaboración de los materiales de formación adecuados.

Para conseguir una utilización efectiva del sistema software se debe proporcionar a los usuarios del sistema instrucciones, guía y ayuda para el entendimiento de las capacidades del sistema y de sus limitaciones. Por ello es imprescindible la formación de los usuarios en el nuevo sistema.

El desarrollo de productos software de calidad depende en gran medida de los conocimientos de las personas y del personal especializado involucrados en el proyecto. Por ello, es esencial la formación de los desarrolladores y personal de soporte técnico.

##### 4.4.4.1 Componentes del proceso: Formación

- **Responsable:**
  - Jefe de proyecto
- **Entregable:**
  - Plan de formación
- **Propósito:**
  - Desarrollar los materiales de formación
  - Validar el programa de formación
  - Implementar el programa de formación