Comunicación:

Es el primer y mas importante caso para poder intercambiar ideas entre contratador y desarrollador ya que el que contrata requiere y el que desarrolla analiza e intercambia ideas para poder mejorar el software que se llevara a cabo por el equipo de desarrolladores. Esto evitara contratiempos por falta de comunicación entre contratador y desarrollador.

Recoleccion de solicitudes:

Desde este momento el equipo de desarrollo de software pone en practica la recolección de datos, de inconvenientes, se intercambian ideas para poder trabajar y tirar hacia adelante el proyecto que se solicito.

La recolección de todos los requisitos se lleva a cabo como se especifica:

* Estudiando el software y el sistema actual u obsoleto.
* Entrevistando a usuarios y a desarrolladores de Software.
* Consultando la base de datos.
* Recogiendo respuestas a través de cuestionarios.

Estudio de viabilidad

Se recolectan los requisitos, ideas acerca del producto, se analizan los pro y contras para realizar el producto y estudiar lo que es necesario y lo que esta obsoleto en el software anterior, todo esto se hace con el equipo de trabajo contratado para realizar y mejorar el funcionamiento del software. Se habla de un consto financiero favorable a su vez.

Análisis del sistema

Se traza el plan con el cual se llevara acabo el desarrollo del modelo del software, se toman los cambios que se requieren e implementación en el software para obtener un rendimiento de mejor calidad, una optimización de problemas, una forma mejor de poder solucionar los problemas que se daba en el software anterior, ya sea por tema de estar obsoleto o por falta de mantenimiento o en este caso falta de actualización o implementación de nuevas herramientas. El equipo del proyecto analiza las posibilidades del proyecto y planifica la temporalización y los recursos correspondientes.

Diseño de Software

Se diseña el software con la ayuda de todo el equipo de trabajo o de desarrollo, se toman en cuenta el intercambio de ideas que sean útiles para el buen funcionamiento del software, el resultado será dado dependiendo del diseño lógico y diseño físico que se propuso. Esto se da por Diagramas de flujo, pseudocódigos y diagrama de clases que se dieron entre el grupo de trabajo.

Codificación

Esta es la fase de programación o desarrollo, donde se implementan el diseño lógico y físico, a medida que se va desarrollando se van poniendo a prueba las ejecutables y darse cuenta de los errores que se van obteniendo y si hay alguno bucar alguna forma de dar solución.

Pruebas

Hay un estimado de todos los procesos de desarrollo de software que deberían ser evaluados, debido a que se pueden presentar errores o funcionamiento no esperado en algún estilo del código o físico del software, aca se toman en cuenta la evaluación de modulos, evaluación del programa, evaluación del producto y asi obtener la evaluación final y obtener un resultado positivo.

Integración

Aca es donde se da el trabajo que muchos quisieran omitir, ya que el software creado se debe de integrar a bases de datos o bibliotecas que quizá no se tomaron en cuenta, se enfoca en la integración del software con el mundo exterior.

Implementación

Se adapta el software a configuraciones para el consumidor final con posterioridad. Se evalua su adaptabilidad y su portabilidad y fiabilidad del software realizado, todo lo anterior se resuelve durante la implementación.

Mantenimiento y Funcionamiento

Se confirma el funcionamiento del software, se obtienen datos o criticas constructivas acerca del software, de estos datos se toma en cuenta el mantenimiento necesario adicional que requiere el software, en esta fase se encaran retos originados por virus ocultos o problemas no identificados del mundo real.

Disposición

Con el paso del tiempo, puede que el software falle en su ejecución. Puede que se vuelva totalmente obsoleto o que necesite actualizaciones. De ahí surge una necesidad urgente de eliminar una parte importante del sistema. Esta fase incluye archivar datos y componentes software requeridos, cierre del sistema, planificación de la actividad de disposición y terminación de sistema en el momento final del sistema.

Paradigma de desarrollo de Software

Ayuda al desarrollador a escoger una estrategia para desarrollar el software. Tiene su propio set de herramientas, métodos y procedimientos, los cuales son expresados de forma clara, y define el ciclo de vida del desarrollo del software. Algunos paradigmas de desarrollo de software o modelos de proceso se definen a continuación:

Modelo de cascada

El modelo de cascada es el modelo de paradigma más simple en desarrollo de software. Lo que significa que solamente cuando la primera fase se termina se puede empezar con la segunda, y así progresivamente.

Forma de implementarla en mi proyecto:

Analizando lo que el ingeniero o auxiliar pide, las restricciones que se han asignado. Tomando en cuenta las ideas de muchos de mis compañeros o tomando en cuenta las criticas realizadas hacia algunas apps ya sea en funcionamiento o en forma física de la misma aplicación, las pruebas se van realizando a medida que se va codificando ya que debo de estar seguro de que no haya algún método que haga que el código este mal, antes de todo esto realizo mi diagrama de clases para poder tener un código con menos problemas a la hora de codificar. Tomar en cuenta las restricciones es algo muy importante porque en este caso las restricciones son el que contrata ya que no desea que se realicen ciertas partes del proyecto o lo quiere con ese estilo. Respecto al modelo cascada, no tendría la capacidad para modificar 2 codigos o realizar ambos para que todo este en orden y ponerlos a prueba, seria algo muy complicado.